



Panduan Pengguna

Sistem File Elastis Amazon



Sistem File Elastis Amazon: Panduan Pengguna

Copyright © 2026 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 pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Sistem File Elastis Amazon?	1
Apakah Anda pengguna pertama kali Amazon EFS?	2
Cara kerjanya	3
Bagaimana Amazon EFS bekerja dengan Amazon EC2	4
Cara Amazon EFS bekerja dengan AWS Direct Connect dan AWS Managed VPN	7
Bagaimana Amazon EFS bekerja dengan AWS Backup	8
Fitur	9
Kontrol autentikasi dan akses	9
Konsistensi data di Amazon EFS	9
Ketersediaan dan daya tahan	10
Replikasi	17
Memulai	18
Prasyarat	18
Buat sistem file dan luncurkan instans EC2	18
Transfer file ke sistem file Anda	19
Prasyarat	20
Pembersihan sumber daya	20
Menginstal klien EFS	22
Dependensi untuk alat EFS	23
Distribusi yang didukung	23
Menginstal klien EFS secara manual	25
Menginstal klien Amazon EFS di instans Amazon EC2 Linux	25
Menginstal klien Amazon EFS di distribusi Linux lainnya	26
Menginstal klien EFS pada instans EC2 Mac	27
Menginstal klien EFS secara otomatis	27
Apa yang dilakukan klien Amazon EFS selama instalasi	28
Sistem operasi yang didukung Systems Manager	28
Mengkonfigurasi AWS Systems Manager untuk menginstal klien EFS	30
Instalasi dan upgrade botocore	31
Upgrade stunnel	31
Menyelesaikan masalah dengan menginstal stunnel	33
Mengaktifkan mode FIPS	35
Membuat dan mengelola sumber daya	37
Ringkasan implementasi	38

Sumber Daya IDs	39
Token pembuatan dan idempotensi	40
Membuat sistem file	40
Izin IAM yang diperlukan untuk membuat sistem file	41
Opsi konfigurasi untuk sistem file	41
Menghapus sistem file	58
Membuat kebijakan sistem file	59
Membuat titik akses	61
Menghapus titik akses	64
Memberikan tag ke sumber daya	64
Dasar-dasar tanda	65
Pembatasan tag	65
Menggunakan tag untuk kontrol akses	66
Memberikan tag pada sumber daya Anda	66
Tutorial: Membuat subdirektori per pengguna yang dapat ditulis	67
Memasang sistem file	70
Pertimbangan pemasangan untuk Linux	70
Menggunakan EFS mount helper	72
Pengaturan pemasangan yang digunakan oleh EFS mount helper	74
Mendapatkan log dukungan	75
Prasyarat	76
Pemasangan di EC2 Linux	77
Pemasangan di EC2 Mac	79
Pemasangan dari Wilayah yang berbeda	81
Memasang sistem file One Zone	82
Pemasangan dengan otorisasi IAM	85
Pemasangan dengan titik akses EFS	86
Memasang beberapa instans EC2	87
Pemasangan dari akun lain atau VPC	88
Menggunakan NFS	91
Prasyarat	92
Dukungan NFS	92
Menginstal klien NFS	93
Pengaturan pemasangan NFS yang disarankan	95
Pemasangan pada EC2 dengan DNS	97
Pemasangan dengan alamat IP	100

Secara otomatis memasang sistem file	102
Instans Linux EC2 baru	102
Instans Linux EC2 yang ada	104
Instans Linux dan Mac menggunakan NFS	108
Melepaskan sistem file	110
Tutorial: Membuat dan memasang sistem file menggunakan AWS CLI	111
Prasyarat	112
Menyiapkan AWS CLI	113
Langkah 1: Buat sumber daya EC2	114
Langkah 2: Buat sumber daya EFS	119
Langkah 3: Pasang dan uji sistem file	122
Langkah 4: Membersihkan	126
Tutorial: Pemasangan dengan klien lokal	128
Prasyarat	129
Langkah 1: Buat sumber daya EFS Anda	131
Langkah 2: Instal klien NFS	132
Langkah 3: Pasang sistem file EFS di klien lokal	133
Langkah 4: Bersihkan sumber daya dan lindungi AWS akun Anda	135
Opsional: Mengenkripsi data dalam perjalanan	135
Tutorial: Pasang sistem file dari VPC yang berbeda	139
Prasyarat	139
Langkah 1: Tentukan ID Availability Zone target mount	140
Langkah 2: Tentukan alamat IP target mount	141
Langkah 3: Tambahkan entri host untuk target pemasangan	142
Langkah 4: Pasang sistem file Anda menggunakan EFS mount helper	142
Langkah 5: Bersihkan sumber daya dan lindungi Anda Akun AWS	144
Memecahkan masalah pemasangan	145
Pemasangan sistem file pada instance Windows gagal	145
Akses ditolak oleh server	145
Pemasangan otomatis gagal dan instans tidak responsif	146
Pemasangan beberapa sistem file Amazon EFS di /etc/fstab gagal	146
Perintah mount gagal dengan pesan kesalahan “jenis fs yang salah”	147
Perintah mount gagal dengan pesan kesalahan “opsi pemasangan yang salah”	148
Pemasangan dengan titik akses gagal	148
Pemasangan sistem file gagal segera setelah pembuatan sistem file	148
Pemasangan sistem file hang dan kemudian gagal dengan kesalahan timeout	149

Pemasangan sistem file dengan NFS menggunakan nama DNS gagal	149
Pemasangan sistem file gagal dengan “nfs tidak merespons”	150
Status siklus hidup target mount macet	151
Status siklus hidup target pemasangan menunjukkan kesalahan	151
Mount tidak merespons	151
Klien yang dipasang akan terputus	152
Operasi pada sistem file yang baru dipasang mengembalikan Kesalahan “pegangan file buruk”	152
Melepas sistem file gagal	153
Mentransfer data	154
Menggunakan AWS DataSync	154
Menggunakan AWS Transfer Family	155
Prasyarat untuk digunakan dengan Amazon EFS AWS Transfer Family	156
Mengkonfigurasi sistem file EFS Anda untuk AWS Transfer Family	156
Mengelola sistem file	162
Memahami status sistem berkas	162
Mengelola target mount	163
Pasang target dan Availability Zone	164
Pembuatan target pemasangan	167
Menghapus target mount	173
Mengubah VPC target mount	174
Mengubah grup keamanan target mount	175
Mengelola throughput	176
Mengelola siklus hidup penyimpanan	178
Operasi sistem file untuk manajemen siklus hidup	179
Mengkonfigurasi kebijakan siklus hidup	180
Memantau	183
Alat-alat pemantauan	184
Alat otomatis	184
Alat-alat pemantauan manual	185
Pengukuran sistem file	185
Mengukur objek	185
Ukuran sistem file terukur	186
Throughput pengukuran	188
Melihat ukuran kelas penyimpanan	189
Memantau metrik dengan CloudWatch	191

CloudWatch metrik	191
Mengakses metrik CloudWatch	197
Menggunakan CloudWatch metrik	199
Menggunakan matematika metrik dengan CloudWatch metrik	201
Memantau keberhasilan dan kegagalan upaya pemasangan	207
Membuat alarm	208
Logging panggilan API dengan CloudTrail	210
Informasi Amazon EFS di CloudTrail	211
Memahami entri file log Amazon EFS	212
Entri file log Amazon EFS untuk sistem encrypted-at-rest file	219
Menggunakan laporan penagihan dan penggunaan	220
Menggunakan tag alokasi biaya	220
Tanda alokasi biaya yang ditentukan pengguna	221
Laporan penagihan	222
Laporan penggunaan	223
Memahami laporan penagihan dan penggunaan	224
Melacak operasi dalam laporan penggunaan Anda	228
Spesifikasi kinerja	229
Ringkasan kinerja	229
Kelas penyimpanan	231
Mode kinerja	231
Mode throughput	232
Memilih mode throughput	233
Throughput elastis	233
Throughput yang disediakan	234
Pembatasan pada pengalihan throughput dan perubahan jumlah yang disediakan	237
Tips performa	237
I/O Ukuran rata-rata	237
Mengoptimalkan beban kerja yang menuntut throughput tinggi dan IOPS	237
Koneksi simultan	238
Model permintaan	238
Pengaturan pemasangan klien NFS	238
Mengoptimalkan kinerja file kecil	239
Mengoptimalkan kinerja direktori	240
Mengoptimalkan ukuran read_ahead_kb NFS	240
Memecahkan masalah kinerja	241

Tidak dapat membuat sistem berkas EFS	242
Akses ditolak ke file yang diizinkan pada sistem file NFS	242
Kesalahan saat mengakses konsol Amazon EFS	243
Instans Amazon EC2 hang	243
Aplikasi menulis sejumlah besar data hang	243
Kinerja buruk saat membuka banyak file secara paralel	244
Pengaturan NFS khusus yang menyebabkan penundaan penulisan	245
Membuat backup dengan Oracle Recovery Manager lambat	245
Memecahkan masalah AMI dan kernel	246
Tidak dapat chown	246
Sistem file terus melakukan operasi berulang kali karena bug klien	247
Klien menemui jalan buntu	247
Daftar file dalam direktori besar membutuhkan waktu lama	247
Mencadangkan dan mereplikasi data	248
Mencadangkan sistem file	248
Cara AWS Backup bekerja dengan Amazon EFS	249
Izin IAM yang diperlukan	252
Kinerja Backup	252
Mengelola backup otomatis	252
Mereplikasi sistem file	254
Biaya	255
Performa replikasi	255
Izin IAM yang diperlukan	256
Mengkonfigurasi replikasi ke sistem file baru	257
Mengkonfigurasi replikasi ke sistem file yang ada	261
Mereplikasi di seluruh akun AWS	267
Melihat detail replikasi	273
Menghapus konfigurasi replikasi	277
Menggunakan replika	279
Mengamankan data Anda	281
Perlindungan data	282
Enkripsi data	283
Privasi antar jaringan	293
Manajemen identitas dan akses	294
Audiens	295
Mengautentikasi dengan identitas	295

Mengelola akses menggunakan kebijakan	297
Bagaimana Amazon Elastic File System bekerja dengan IAM	298
Contoh kebijakan berbasis identitas	304
Contoh kebijakan berbasis sumber daya	309
AWS kebijakan terkelola	311
Menggunakan tanda	317
Menggunakan Peran Terkait Layanan	321
Pemecahan masalah	325
Mengontrol akses data sistem file	327
Kebijakan Sistem File Default	328
Tindakan EFS untuk klien	328
Kunci kondisi EFS untuk klien	329
Contoh kebijakan sistem file	329
Validasi kepatuhan	330
Ketahanan	330
Mengontrol akses jaringan	330
Menggunakan grup keamanan VPC	331
Bekerja dengan VPC endpoint	334
Pengguna, grup, dan izin tingkat NFS	335
Izin file dan direktori	336
Contoh kasus penggunaan dan izin sistem file EFS	337
Izin ID pengguna dan grup untuk file dan direktori dalam sistem file	338
Tidak ada akar terjepit	339
Caching izin	342
Mengubah kepemilikan objek sistem file	342
Titik akses EFS	342
Bekerja dengan titik akses	343
Titik akses bekerja dengan target pemasangan	343
Menegakkan identitas pengguna	344
Menegakkan direktori root	345
Menggunakan titik akses dalam kebijakan IAM	347
Memblokir akses publik ke sistem file	348
Memblokir akses publik dengan AWS Transfer Family	349
Arti “publik”	349
Isolasi jaringan	352
Kuota	353

Kuota Amazon EFS yang dapat Anda tingkatkan	353
Meminta peningkatan kuota	356
Kuota sumber daya Amazon EFS yang tidak dapat Anda ubah	356
Kuota untuk klien NFS	358
Kuota untuk sistem file Amazon EFS	359
Fitur NFSv4 0 dan 4.1 yang tidak didukung	359
Pertimbangan tambahan	361
Memecahkan masalah kesalahan operasi file yang terkait dengan kuota	361
Memeriksa file terbuka dan kunci file	362
Perintah gagal dengan kesalahan “Kuota disk terlampaui”	362
Perintah gagal dengan “kesalahan I/O”	363
Perintah gagal dengan kesalahan “Nama file terlalu panjang”	363
Perintah gagal dengan kesalahan “File tidak ditemukan”	364
Perintah gagal dengan kesalahan “Terlalu banyak tautan”	364
Perintah gagal dengan kesalahan “File terlalu besar”	364
Amazon EFS API	365
Titik akhir API	365
IPv4 titik akhir	366
Titik akhir tumpukan ganda (IPv4 dan IPv6)	366
Menentukan titik akhir	367
Versi API	367
Topik terkait	367
Bekerja dengan tingkat permintaan API kueri untuk Amazon EFS	367
Pemungutan Suara	368
Pemrosesan coba ulang atau batch	368
Menghitung interval tidur	368
Tindakan	368
CreateAccessPoint	370
CreateFileSystem	380
CreateMountTarget	398
CreateReplicationConfiguration	417
CreateTags	427
DeleteAccessPoint	431
DeleteFileSystem	434
DeleteFileSystemPolicy	439
DeleteMountTarget	443

DeleteReplicationConfiguration	448
DeleteTags	452
DescribeAccessPoints	456
DescribeAccountPreferences	462
DescribeBackupPolicy	466
DescribeFileSystemPolicy	471
DescribeFileSystems	477
DescribeLifecycleConfiguration	484
DescribeMountTargets	489
DescribeMountTargetSecurityGroups	496
DescribeReplicationConfigurations	501
DescribeTags	507
ListTagsForResource	513
ModifyMountTargetSecurityGroups	518
PutAccountPreferences	524
PutBackupPolicy	528
PutFileSystemPolicy	533
PutLifecycleConfiguration	540
TagResource	550
UntagResource	555
UpdateFileSystem	559
UpdateFileSystemProtection	569
Tipe Data	575
AccessPointDescription	577
BackupPolicy	580
CreationInfo	581
Destination	583
DestinationToCreate	586
FileSystemDescription	590
FileSystemProtectionDescription	595
FileSystemSize	596
LifecyclePolicy	598
MountTargetDescription	600
PosixUser	603
ReplicationConfigurationDescription	605
ResourceIdPreference	607

RootDirectory	608
Tag	610
Riwayat dokumen	611
.....	dcxxxviii

Apa itu Sistem File Elastis Amazon?

Amazon Elastic File System (Amazon EFS) menyediakan penyimpanan file nirserver dan sepenuhnya elastis sehingga Anda dapat berbagi data file tanpa perlu menyediakan atau mengelola kapasitas dan performa penyimpanan. Amazon EFS dibangun untuk menskalakan sesuai permintaan ke petabyte tanpa mengganggu aplikasi, tumbuh dan menyusut secara otomatis saat Anda menambahkan dan menghapus file. Karena Amazon EFS memiliki antarmuka layanan web yang sederhana, Anda dapat membuat dan mengonfigurasi sistem file dengan cepat dan mudah. Layanan ini mengelola semua infrastruktur penyimpanan file untuk Anda, artinya Anda dapat menghindari kompleksitas penerapan, penambalan, dan pemeliharaan konfigurasi sistem file yang kompleks.

Amazon EFS mendukung protokol Network File System versi 4 (NFSv4.1 dan NFSv4 .0), sehingga aplikasi dan alat yang Anda gunakan saat ini bekerja dengan mulus dengan Amazon EFS. Amazon EFS dapat diakses di sebagian besar jenis instans komputasi Amazon Web Services, termasuk Amazon, EC2 Amazon ECS, Amazon EKS, AWS Lambda dan. AWS Fargate

Layanan ini dirancang agar sangat skalabel, sangat tersedia, dan sangat tahan lama. Amazon EFS menawarkan jenis sistem file berikut untuk memenuhi kebutuhan ketersediaan dan daya tahan Anda:

- **Regional (Direkomendasikan)** — Sistem file regional (disarankan) menyimpan data secara berlebihan di beberapa Availability Zone yang terpisah secara geografis dalam zona yang sama. AWS Region Menyimpan data di beberapa Availability Zone menyediakan ketersediaan berkelanjutan ke data, bahkan ketika satu atau beberapa Availability Zone dalam sebuah tidak AWS Region tersedia.
- **Satu Zona** — Sistem file One Zone menyimpan data dalam satu Availability Zone. Menyimpan data dalam Availability Zone tunggal memberikan ketersediaan berkelanjutan ke data. Namun, dalam kasus kehilangan atau kerusakan pada semua atau sebagian dari Availability Zone, data yang disimpan dalam jenis sistem file ini mungkin hilang.

Untuk informasi selengkapnya tentang jenis sistem file, lihat [Jenis sistem file EFS](#).

Amazon EFS menyediakan throughput, IOPS, dan latensi rendah yang diperlukan untuk berbagai beban kerja. Sistem file EFS dapat berkembang menjadi skala petabyte, mendorong throughput tingkat tinggi, dan memungkinkan akses paralel besar-besaran dari instans komputasi ke data Anda. Untuk sebagian besar beban kerja, sebaiknya gunakan mode default, yang merupakan mode kinerja Tujuan Umum dan mode throughput Elastis.

- Tujuan Umum — Mode kinerja Tujuan Umum sangat ideal untuk aplikasi yang sensitif terhadap latensi, seperti lingkungan penyajian web, sistem manajemen konten, direktori rumah, dan penyajian file umum.
- Elastis — Mode throughput elastis dirancang untuk secara otomatis menskalakan kinerja throughput ke atas atau ke bawah untuk memenuhi kebutuhan aktivitas beban kerja Anda.

Untuk informasi selengkapnya tentang mode performa dan throughput EFS, lihat [Spesifikasi kinerja Amazon EFS](#).

Amazon EFS menyediakan file-system-access semantik, seperti konsistensi data yang kuat dan penguncian file. Untuk informasi selengkapnya, lihat [Konsistensi data di Amazon EFS](#). Amazon EFS juga mendukung pengendalian akses ke sistem file Anda melalui izin Portable Operating System Interface (POSIX). Untuk informasi selengkapnya, lihat [Mengamankan data Anda di Amazon EFS](#).

Amazon EFS mendukung kemampuan autentikasi, otorisasi, dan enkripsi untuk membantu Anda memenuhi persyaratan keamanan dan kepatuhan Anda. Amazon EFS mendukung dua bentuk enkripsi untuk sistem file: enkripsi dalam perjalanan dan enkripsi saat istirahat. Anda dapat mengaktifkan enkripsi saat istirahat saat membuat sistem file EFS. Jika Anda melakukannya, semua data dan metadata Anda dienkripsi. Anda dapat mengaktifkan enkripsi saat transit saat Anda memasang sistem file. Akses klien NFS ke Amazon EFS dikendalikan oleh kebijakan AWS Identity and Access Management (IAM) dan kebijakan keamanan jaringan, seperti grup keamanan. Lihat informasi selengkapnya di [Enkripsi data di Amazon EFS](#), [Manajemen identitas dan akses untuk Amazon EFS](#), dan [Mengontrol akses jaringan ke sistem file EFS untuk klien NFS](#).

Note

Menggunakan Amazon EFS dengan EC2 instans Amazon berbasis Microsoft Windows tidak didukung.

Apakah Anda pengguna pertama kali Amazon EFS?

Jika Anda adalah pengguna pertama kali Amazon EFS, kami sarankan Anda membaca bagian berikut secara berurutan:

1. Untuk ikhtisar harga dan produk Amazon EFS, lihat [Amazon EFS](#).
2. Untuk ikhtisar teknis Amazon EFS, lihat [Cara kerja Amazon EFS](#).

3. Coba [Memulai](#) latihannya.

Jika Anda ingin mempelajari lebih lanjut tentang Amazon EFS, topik berikut membahas layanan secara lebih rinci:

- [Membuat dan mengelola sumber daya EFS](#)
- [Mengelola sistem file EFS](#)
- [Amazon EFS API](#)

Cara kerja Amazon EFS

Amazon Elastic File System (EFS) menyediakan sistem file set-and-forget elastis yang sederhana, tanpa server. Dengan Amazon EFS, Anda dapat membuat sistem file, memasang sistem file pada instans Amazon EC2, lalu membaca dan menulis data ke dan dari sistem file Anda. Anda dapat memasang sistem file EFS di virtual private cloud (VPC) Anda, melalui protokol Network File System versi 4.0 dan 4.1 (NFSv4). Sebaiknya gunakan klien Linux NFSv4 .1 generasi saat ini, seperti yang ditemukan di Amazon Linux terbaru, Amazon Linux 2, Red Hat, Ubuntu, dan macOS Big AMIs Sur, bersama dengan EFS mount helper. Untuk petunjuk, lihat [Menginstal klien Amazon EFS](#).

Untuk daftar Amazon EC2 Linux dan MacOS Amazon Machine Images (AMIs) yang mendukung protokol ini, lihat [Dukungan NFS](#). Untuk beberapa AMI, Anda harus menginstal klien NFS untuk memasang sistem file Anda di instans Amazon EC2 Anda. Untuk petunjuk, lihat [Menginstal klien NFS](#).

Anda dapat mengakses sistem file EFS Anda secara bersamaan dari beberapa klien NFS, sehingga aplikasi yang berskala di luar satu koneksi dapat mengakses sistem file. Amazon EC2 dan instans AWS komputasi lainnya yang berjalan di beberapa Availability Zone dalam hal yang sama AWS Region dapat mengakses sistem file, sehingga banyak pengguna dapat mengakses dan berbagi sumber data umum.

Untuk daftar Wilayah AWS di mana Anda dapat membuat sistem file EFS, lihat [Referensi Umum Amazon Web Services](#).

Untuk mengakses sistem file EFS Anda di VPC, Anda membuat satu atau beberapa target mount di VPC. Target mount menyediakan alamat IP untuk titik akhir NFSv4 tempat Anda dapat memasang sistem file EFS. Anda memasang sistem file Anda menggunakan nama Domain Name Service

(DNS), yang menyelesaikan ke alamat IP target pemasangan EFS di Availability Zone yang sama dengan instans EC2 Anda. Anda dapat membuat satu target mount di setiap Availability Zone di file AWS Region. Jika ada beberapa subnet di Availability Zone di VPC Anda, Anda membuat target mount di salah satu subnet. Kemudian semua instans EC2 di Availability Zone tersebut berbagi target mount tersebut.

 Note

Sistem file EFS dapat memiliki target mount hanya dalam satu VPC pada satu waktu.

Target mount sendiri dirancang agar sangat tersedia. Saat Anda merancang ketersediaan tinggi dan failover ke Availability Zone lainnya, ingatlah bahwa meskipun alamat IP dan DNS untuk target mount Anda di setiap Availability Zone bersifat statis, mereka adalah komponen redundan yang didukung oleh beberapa sumber daya. Untuk informasi lebih lanjut tentang target pemasangan, lihat [Mengelola target mount](#).

Setelah memasang sistem file dengan menggunakan nama DNS-nya, Anda menggunakannya seperti sistem file lainnya. Untuk informasi tentang izin tingkat NFS dan pertimbangan terkait, lihat [Network File System \(NFS\) tingkat pengguna, grup, dan izin](#)

Anda dapat memasang sistem file EFS di server pusat data lokal saat tersambung ke VPC AWS Direct Connect Amazon dengan atau. Site-to-Site VPN Anda dapat memasang sistem file EFS di server lokal untuk memigrasikan kumpulan data ke EFS, mengaktifkan skenario ledakan cloud, atau mencadangkan data lokal ke Amazon EFS.

Berikut ini, Anda dapat menemukan deskripsi tentang cara Amazon EFS bekerja dengan layanan lain.

Topik

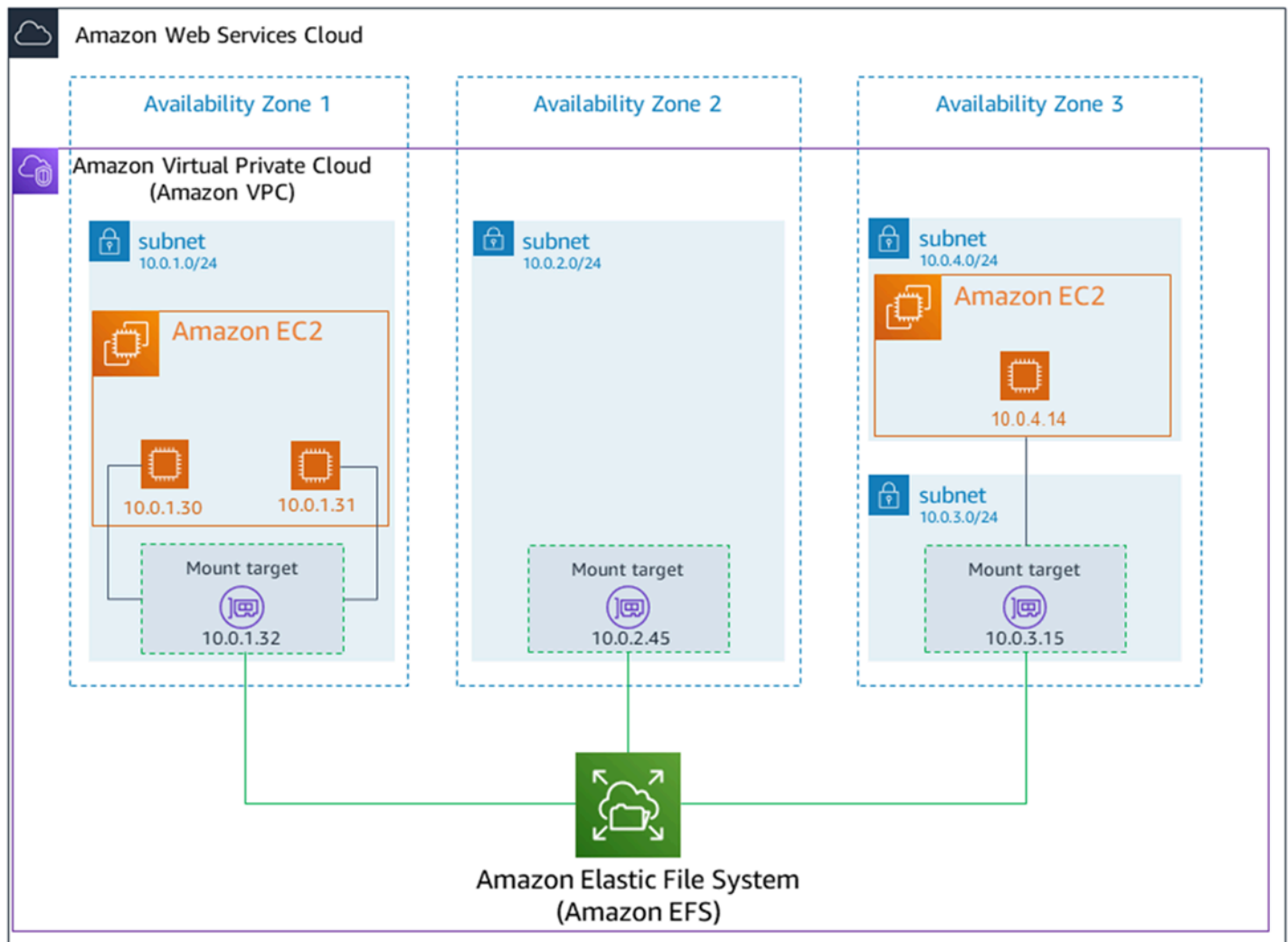
- [Bagaimana Amazon EFS bekerja dengan Amazon EC2](#)
- [Cara Amazon EFS bekerja dengan AWS Direct Connect dan AWS Managed VPN](#)
- [Bagaimana Amazon EFS bekerja dengan AWS Backup](#)

Bagaimana Amazon EFS bekerja dengan Amazon EC2

Bagian ini menjelaskan bagaimana sistem file Amazon EFS Regional dan One Zone dipasang ke instans EC2 di Amazon VPC.

Sistem file EFS regional

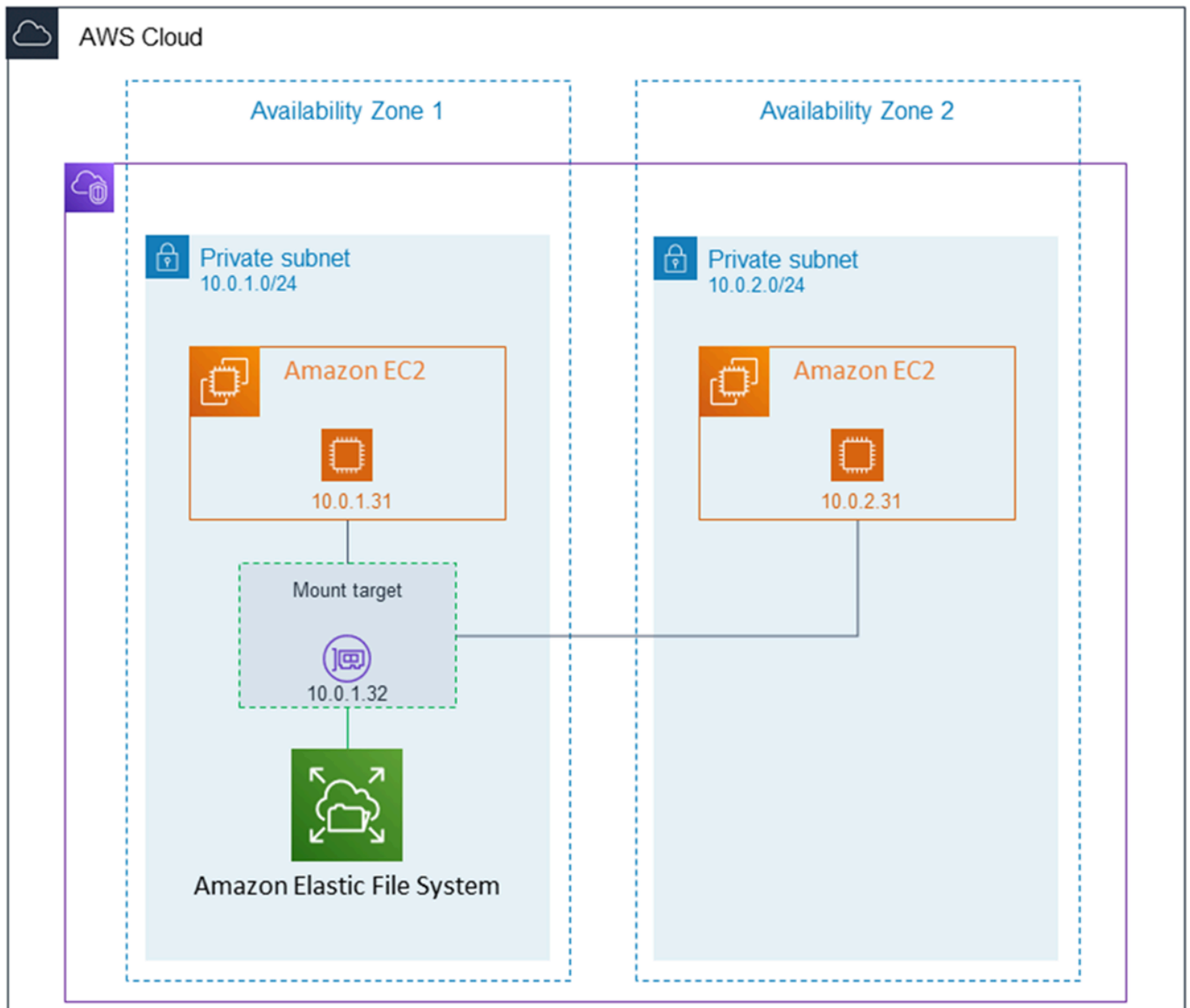
Ilustrasi berikut menunjukkan beberapa instans EC2 yang mengakses sistem file Amazon EFS yang dikonfigurasi untuk beberapa Availability Zone dalam file. AWS Region



Dalam ilustrasi ini, virtual private cloud (VPC) memiliki tiga Availability Zone. Karena sistem file adalah Regional, target mount dibuat di setiap Availability Zone. Kami menyarankan Anda mengakses sistem file dari target pemasangan dalam Availability Zone yang sama untuk alasan kinerja dan biaya. Salah satu Availability Zones memiliki dua subnet. Namun, target mount dibuat hanya di salah satu subnet. Untuk informasi selengkapnya, lihat [Memasang sistem file EFS menggunakan EFS mount helper](#).

Sistem file One Zone EFS

Ilustrasi berikut menunjukkan beberapa instans EC2 mengakses sistem file One Zone dari Availability Zone yang berbeda dalam satu. AWS Region



Dalam ilustrasi ini, VPC memiliki dua Availability Zone, masing-masing dengan satu subnet. Karena jenis sistem file adalah One Zone, ia hanya dapat memiliki satu target mount. Untuk kinerja dan biaya yang lebih baik, sebaiknya Anda mengakses sistem file dari target pemasangan di Availability Zone yang sama dengan instans EC2 tempat Anda memasangnya.

Dalam contoh ini, instans EC2 di Availability Zone us-west-2c akan membayar biaya akses data EC2 untuk mengakses target mount di Availability Zone yang berbeda. Untuk informasi selengkapnya, lihat [Memasang sistem file One Zone](#).

Cara Amazon EFS bekerja dengan AWS Direct Connect dan AWS Managed VPN

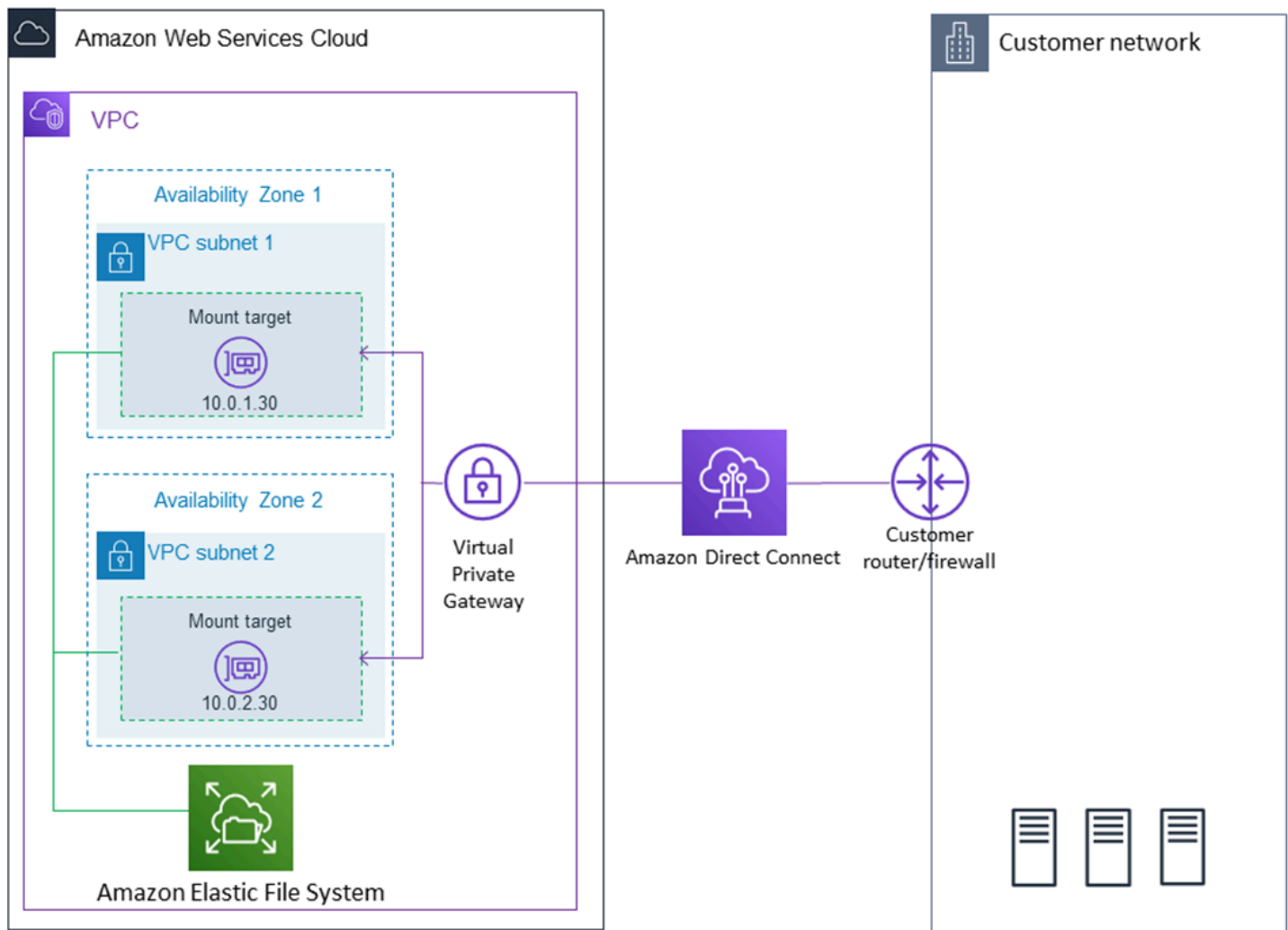
Dengan menggunakan sistem file Amazon EFS yang dipasang di server lokal, Anda dapat memigrasikan data lokal ke AWS Cloud host di sistem file Amazon EFS. Anda juga bisa memanfaatkan meledak. Dengan kata lain, Anda dapat memindahkan data dari server lokal ke Amazon EFS dan menganalisisnya pada armada instans Amazon EC2 di Amazon VPC Anda. Anda kemudian dapat menyimpan hasilnya secara permanen di sistem file Anda atau memindahkan hasilnya kembali ke server lokal Anda.

Ingatlah pertimbangan berikut saat menggunakan Amazon EFS dengan server lokal:

- Server lokal Anda harus memiliki sistem operasi berbasis Linux. Kami merekomendasikan kernel Linux versi 4.0 atau yang lebih baru.
- Demi kesederhanaan, sebaiknya pasang sistem file Amazon EFS di server lokal menggunakan alamat IP target mount, bukan nama DNS.

Tidak ada biaya tambahan untuk akses lokal ke sistem file Amazon EFS Anda. Anda dikenakan biaya untuk Direct Connect koneksi ke VPC Amazon Anda. Untuk informasi selengkapnya, lihat [harga Direct Connect](#).

Ilustrasi berikut menunjukkan contoh cara mengakses sistem file Amazon EFS dari lokal (server lokal memiliki sistem file yang terpasang).



Anda dapat menggunakan target pemasangan apa pun di VPC jika Anda dapat mencapai subnet target mount tersebut dengan menggunakan Direct Connect koneksi antara server lokal dan VPC. Untuk mengakses Amazon EFS dari server lokal, tambahkan aturan ke grup keamanan target pemasangan Anda untuk mengizinkan lalu lintas masuk ke port NFS (2049) dari server lokal Anda. Untuk informasi lebih lanjut, termasuk prosedur terperinci, lihat [Prasyarat](#).

Bagaimana Amazon EFS bekerja dengan AWS Backup

Untuk implementasi pencadangan komprehensif untuk sistem file Anda, Anda dapat menggunakan Amazon EFS dengan file AWS Backup. AWS Backup adalah layanan pencadangan yang dikelola sepenuhnya yang memudahkan untuk memusatkan dan mengotomatiskan pencadangan data di seluruh AWS layanan di cloud dan lokal. Dengan menggunakan AWS Backup, Anda dapat mengonfigurasi kebijakan pencadangan secara terpusat dan memantau aktivitas pencadangan untuk AWS sumber daya Anda. Amazon EFS selalu memprioritaskan operasi sistem file daripada operasi

pencadangan. Untuk mempelajari lebih lanjut tentang mencadangkan sistem file EFS menggunakan AWS Backup, lihat [Mencadangkan sistem file EFS](#).

Fitur Amazon EFS

Berikut ini adalah fitur Amazon EFS.

Topik

- [Kontrol autentikasi dan akses](#)
- [Konsistensi data di Amazon EFS](#)
- [Ketersediaan dan daya tahan sistem file EFS](#)
- [Replikasi](#)

Kontrol autentikasi dan akses

Anda harus memiliki kredensial yang valid untuk menggunakan konsol manajemen Amazon EFS dan membuat permintaan Amazon EFS API, seperti membuat sistem file. Selain itu, Anda juga harus memiliki izin untuk membuat atau mengakses EFS dan AWS sumber daya lainnya.

Pengguna dan peran yang Anda buat AWS Identity and Access Management (IAM) harus diberikan izin untuk membuat atau mengakses sumber daya. Untuk informasi selengkapnya tentang izin, lihat [Manajemen identitas dan akses untuk Amazon EFS](#).

Otorisasi IAM untuk klien NFS adalah opsi keamanan tambahan untuk Amazon EFS yang menggunakan IAM untuk menyederhanakan manajemen akses untuk klien Network File System (NFS) dalam skala besar. Dengan otorisasi IAM untuk klien NFS, Anda dapat menggunakan IAM untuk mengelola akses ke sistem file EFS dengan cara yang dapat diskalakan secara inheren. Otorisasi IAM untuk klien NFS juga dioptimalkan untuk lingkungan cloud. Untuk informasi selengkapnya tentang penggunaan otorisasi IAM untuk klien NFS, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#)

Konsistensi data di Amazon EFS

Amazon EFS menyediakan semantik close-to-open konsistensi yang diharapkan aplikasi dari NFS.

Di Amazon EFS, operasi tulis untuk sistem file Regional disimpan dengan tahan lama di seluruh Availability Zone dalam situasi berikut:

- Aplikasi melakukan operasi tulis sinkron (misalnya, menggunakan perintah `open` Linux dengan `O_DIRECT` bendera, atau perintah `fsync` Linux).
- Aplikasi menutup file.

Bergantung pada pola aksesnya, Amazon EFS dapat memberikan jaminan konsistensi yang lebih kuat daripada close-to-open semantik. Aplikasi yang melakukan akses data sinkron dan melakukan penulisan non-appending memiliki read-after-write konsistensi untuk akses data.

Penguncian file

Aplikasi klien NFS dapat menggunakan penguncian file NFS versi 4 (termasuk penguncian rentang byte) untuk operasi baca dan tulis pada file Amazon EFS.

Ingat hal berikut tentang bagaimana Amazon EFS mengunci file:

- Amazon EFS hanya mendukung penguncian penasihat dan read/write operasi tidak memeriksa kunci yang bertentangan sebelum dijalankan. Misalnya, untuk menghindari masalah sinkronisasi file dengan operasi atom, aplikasi Anda harus mengetahui semantik NFS (seperti konsistensi). close-to-open
- Setiap file tertentu dapat memiliki hingga 512 kunci di semua instance yang terhubung dan pengguna mengakses file tersebut.

Ketersediaan dan daya tahan sistem file EFS

Bagian ini menjelaskan jenis sistem file dan opsi kelas penyimpanan untuk sistem file Amazon Elastic File System (Amazon EFS).

Jenis sistem file EFS

Amazon EFS menawarkan tipe sistem file Regional dan One Zone.

- Regional — Sistem file regional (disarankan) menyimpan data secara berlebihan di beberapa Availability Zone yang terpisah secara geografis dalam zona yang sama. AWS Region Menyimpan data di beberapa Availability Zone menyediakan ketersediaan berkelanjutan ke data, bahkan ketika satu atau beberapa Availability Zone dalam sebuah tidak AWS Region tersedia.
- Satu Zona — Sistem file One Zone menyimpan data dalam satu Availability Zone. Menyimpan data dalam Availability Zone tunggal memberikan ketersediaan berkelanjutan ke data. Namun, dalam

kasus kehilangan atau kerusakan pada semua atau sebagian dari Availability Zone, data yang disimpan dalam jenis sistem file ini mungkin hilang.

Dalam kasus yang tidak mungkin terjadi kehilangan atau kerusakan pada semua atau sebagian dari AWS Availability Zone, data dalam kelas penyimpanan One Zone mungkin hilang. Misalnya, peristiwa seperti kebakaran dan kerusakan air dapat mengakibatkan kehilangan data. Terlepas dari jenis peristiwa ini, kelas penyimpanan One Zone kami menggunakan desain teknik yang serupa dengan kelas penyimpanan Regional kami untuk melindungi objek dari kegagalan disk, host, dan tingkat rak independen, dan masing-masing dirancang untuk memberikan daya tahan data 99,999999999%.

Untuk perlindungan data tambahan, Amazon EFS secara otomatis mencadangkan sistem file One Zone dengan file AWS Backup. Anda dapat mengembalikan cadangan sistem file ke Availability Zone operasional apa pun di dalam AWS Region, atau Anda dapat mengembalikannya ke yang lain. AWS Region Pencadangan sistem file EFS yang dibuat dan dikelola menggunakan AWS Backup direplikasi ke tiga Availability Zone dan dirancang untuk daya tahan. Untuk informasi lebih lanjut, lihat [Ketahanan](#) di AWS Backup

Note

Sistem file One Zone hanya tersedia untuk Availability Zone tertentu. Untuk tabel yang mencantumkan Availability Zones di mana Anda dapat menggunakan sistem file One Zone, lihat [Zona Ketersediaan yang Didukung untuk sistem file One Zone](#).

Tabel berikut membandingkan jenis sistem file, termasuk ketersediaan, daya tahan, dan pertimbangan lainnya.

Tipe Sistem File	Dirancang untuk	Ketahanan (dirancang untuk)	Ketersediaan	Zona Ketersediaan	Pertimbangan lainnya
Regional	Data yang membutuhkan daya tahan dan ketersediaan tertinggi.	99,999999 999% (11 9 detik)	99,99%	>=3	Tidak ada

Tipe Sistem File	Dirancang untuk	Ketahanan (dirancang untuk)	Ketersediaan	Zona Ketersediaan	Pertimbangan lainnya
Satu Zona	Data yang tidak memerlukan daya tahan dan ketersediaan tertinggi.	99,999999 999% (11 9 detik)	99,99%	1	Tidak tahan terhadap hilangnya Availability Zone

Zona Ketersediaan yang Didukung untuk sistem file One Zone

Sistem file One Zone hanya tersedia untuk Availability Zone tertentu. Tabel berikut mencantumkan AWS Region dan AZ IDs untuk setiap Availability Zone di mana Anda dapat menggunakan sistem file One Zone. Untuk melihat pemetaan AZ IDs ke Availability Zones di akun Anda, lihat [Availability Zone IDs untuk AWS Resource Anda](#) di Panduan Pengguna AWS Resource Access Manager.

Availability Zones yang mendukung sistem file One Zone

AWS Region Nama	AWS Region Kode	AZ yang didukung IDs
AS Timur (Ohio)	us-east-2	gunakan2-az1, gunakan2-az2, gunakan2-az3
US East (Northern Virginia)	us-east-1	gunakan1-az1, gunakan1-az2, gunakan1-az4, gunakan1-az5, gunakan1-az6
AS Barat (California Utara)	us-west-1	usw1-az1, usw1-az3
AS Barat (Oregon)	us-west-2	usw2-az1, usw2-az2, usw2-az3, usw2-az4
Africa (Cape Town)	af-south-1	afs1-az1, afs1-az2, afs1-az3
Asia Pasifik (Hong Kong)	ap-east-1	ape1-az1, ape1-az2, ape1-az3
Asia Pasifik (Mumbai)	ap-south-1	aps1-az1, aps1-az2, aps1-az3

AWS Region Nama	AWS Region Kode	AZ yang didukung IDs
Asia Pacific (Osaka)	ap-northeast-3	apne3-az1, apne3-az2, apne3-az3
Asia Pasifik (Seoul)	ap-northeast-2	apne2-az1, apne2-az2, apne2-az3
Asia Pasifik (Singapura)	ap-southeast-1	apse1-az1, apse1-az2
Asia Pasifik (Sydney)	ap-southeast-2	apse2-az1, apse2-az2, apse2-az3
Asia Pasifik (Tokyo)	ap-northeast-1	apne1-az1, apne1-az4
Kanada (Pusat)	ca-central-1	cac1-az1, cac1-az2
Tiongkok (Beijing)	cn-north-1	cnn1-az1, cnn1-az2
Tiongkok (Ningxia)	cn-northwest-1	cnnw1-az1, cnnw1-az2, cnnw1-az3
Eropa (Frankfurt)	eu-central-1	euc1-az1, euc1-az2, euc1-az3
Eropa (Irlandia)	eu-west-1	euw1-az1, euw1-az2, euw1-az3
Eropa (London)	eu-west-2	euw2-az1, euw2-az2
Europe (Milan)	eu-south-1	eus1-az1, eus1-az2, eus1-az3
Eropa (Paris)	eu-west-3	euw3-az1, euw3-az3
Eropa (Stockholm)	eu-north-1	eun1-az1, eun1-az2, eun1-az3
Timur Tengah (Bahrain)	me-south-1	mes1-az1, mes1-az2, mes1-az3
Amerika Selatan (Sao Paulo)	sa-east-1	sae1-az1, sae1-az2, sae1-az3

AWS Region Nama	AWS Region Kode	AZ yang didukung IDs
AWS GovCloud (AS-Timur)	us-gov-east-1	usge1-az1, usge1-az2, usge1-az3
AWS GovCloud (AS-Barat)	us-gov-west-1	usgw1-az1, usgw1-az2, usgw1-az3

Kelas penyimpanan EFS

Amazon EFS menawarkan kelas penyimpanan berbeda yang dirancang untuk penyimpanan paling efektif tergantung pada kasus penggunaan.

- **EFS Standard** — Kelas penyimpanan EFS Standard menggunakan penyimpanan solid state drive (SSD) untuk memberikan tingkat latensi terendah untuk file yang sering diakses. Data sistem file baru pertama kali ditulis ke kelas penyimpanan EFS Standard dan kemudian dapat berjenjang ke kelas penyimpanan EFS Infrequent Access dan EFS Archive dengan menggunakan manajemen siklus hidup.
- **EFS Infrequent Access (IA)** — Kelas penyimpanan yang dioptimalkan biaya untuk data yang diakses hanya beberapa kali setiap kuartal.
- **EFS Archive** — Kelas penyimpanan yang dioptimalkan biaya untuk data yang diakses beberapa kali setiap tahun atau kurang.

Kelas penyimpanan EFS Archive didukung pada sistem file EFS dengan throughput Elastic. Anda tidak dapat memperbarui throughput sistem file Anda ke Bursting atau Provisioned setelah sistem file memiliki data di kelas penyimpanan Arsip.

Membandingkan kelas penyimpanan

Tabel berikut menyediakan perbandingan kelas penyimpanan. Untuk detail selengkapnya tentang kinerja setiap kelas penyimpanan, lihat [Spesifikasi kinerja Amazon EFS](#).

Kelas penyimpanan	Dirancang untuk	Latensi baca byte pertama	Daya tahan (dirancang untuk) ¹	Ketersediaan SLA	Zona Ketersediaan	Biaya penagihan minimum per file ²	Durasi penyimpanan minimum
EFS Standar	Data aktif yang membutuhkan kinerja latensi sub-milidetik yang cepat	Sub-milidetik	99,999999999% (11 9)	99,99% (Regional)	=>3 (Regional)	Tidak berlaku	Tidak berlaku
EFS Akses Jarang	Data tidak aktif yang diakses hanya beberapa kali setiap kuartal.	Puluhan milidetik		99,9% (Satu Zona)		128 KiB	Tidak berlaku
Arsip EFS	Data tidak aktif yang diakses beberapa kali setiap tahun atau kurang	Puluhan milidetik		99,9% (Regional)	=>3 (Regional)	128 KiB	90 hari

¹ Karena sistem file One Zone menyimpan data dalam AWS Availability Zone tunggal, data yang disimpan dalam jenis sistem file ini mungkin hilang jika terjadi bencana atau kesalahan lain yang memengaruhi semua salinan data dalam Availability Zone, atau jika terjadi kerusakan Availability Zone.

² Kebijakan Siklus Hidup yang diperbarui pada atau setelah pukul 12 PM PT, 26 November 2023 akan menilai file yang lebih kecil dari 128 KiB ke kelas IA. Untuk informasi selengkapnya tentang cara Amazon EFS mengukur dan menagih untuk masing-masing file dan metadata, lihat. [Bagaimana Amazon EFS melaporkan sistem file dan ukuran objek](#)

Penagihan kelas penyimpanan

Anda ditagih untuk jumlah data di setiap kelas penyimpanan. Anda juga ditagih biaya akses data saat file di IA atau penyimpanan Arsip dibaca, atau untuk data yang bertransisi antar kelas penyimpanan

menggunakan manajemen siklus hidup. AWS Tagihan menampilkan kapasitas untuk setiap kelas penyimpanan dan akses terukur terhadap kelas penyimpanan sistem file. Untuk mempelajari lebih lanjut, lihat [Harga Amazon EFS](#).

Selain itu, kelas penyimpanan Infrequent Access (IA) dan Arsip memiliki biaya penagihan minimum per file 128 KiB. Support untuk file yang lebih kecil dari 128 KiB hanya tersedia untuk kebijakan siklus hidup yang diperbarui pada atau setelah 12:00 PM PT, 26 November 2023. Untuk informasi selengkapnya tentang cara Amazon EFS mengukur dan menagih untuk masing-masing file dan metadata, lihat [Bagaimana Amazon EFS melaporkan sistem file dan ukuran objek](#)

Pertimbangan penagihan tambahan berlaku tergantung pada mode throughput.

- Untuk sistem file yang menggunakan throughput Elastic, Anda ditagih untuk jumlah total metadata bulanan dan data yang ditransfer untuk sistem file Anda, terlepas dari biaya penyimpanan.
- Untuk sistem file yang menggunakan throughput Provisioned, Anda ditagih untuk throughput yang disediakan di atas apa yang Anda berikan berdasarkan jumlah data yang ada di kelas penyimpanan EFS Standard.
- Untuk sistem file yang menggunakan throughput Bursting, throughput yang diizinkan ditentukan berdasarkan jumlah data yang disimpan di kelas penyimpanan EFS Standard saja.

Untuk informasi selengkapnya tentang mode throughput EFS, lihat [Mode throughput](#).

Note

Anda tidak dikenakan biaya akses data saat menggunakan AWS Backup untuk mencadangkan sistem file EFS yang mendukung manajemen siklus hidup. Untuk mempelajari lebih lanjut tentang AWS Backup Amazon EFS, lihat [Mencadangkan sistem file EFS](#).

Manajemen siklus hidup

Untuk mengelola sistem file Anda sehingga disimpan secara efektif sepanjang siklus hidupnya, gunakan manajemen siklus hidup. Manajemen siklus hidup secara otomatis mentransisikan data antar kelas penyimpanan sesuai dengan konfigurasi siklus hidup yang ditentukan untuk sistem file. Konfigurasi siklus hidup adalah sekumpulan kebijakan siklus hidup yang menentukan kapan harus mentransisikan data sistem file ke kelas penyimpanan lain. Untuk informasi selengkapnya, lihat [Mengelola siklus hidup penyimpanan](#).

Replikasi

Anda dapat membuat replika sistem file Amazon EFS Anda sesuai preferensi Anda menggunakan replikasi. AWS Region Replikasi secara otomatis dan transparan mereplikasi data dan metadata pada sistem file EFS Anda ke sistem file EFS tujuan baru yang dibuat sesuai pilihan Anda. AWS Region EFS secara otomatis menjaga sistem file sumber dan tujuan tetap disinkronkan. Replikasi berkelanjutan dan dirancang untuk memberikan tujuan titik pemulihan (RPO) dan tujuan waktu pemulihan (RTO) menit. Fitur-fitur ini membantu Anda dalam memenuhi kepatuhan dan tujuan kelangsungan bisnis Anda. Lihat informasi yang lebih lengkap di [Mereplikasi sistem file EFS](#).

Memulai dengan Amazon EFS

Jika Anda menggunakan Amazon Elastic File System (Amazon EFS) untuk pertama kalinya, selesaikan langkah-langkah berikut untuk memulai sistem file EFS pertama Anda.

1. [Tinjau prasyarat untuk memulai](#)
2. [Buat sistem file EFS Anda dan luncurkan instans EC2 Anda](#)
3. [Transfer file ke sistem file EFS Anda menggunakan AWS DataSync](#)
4. [Bersihkan sumber daya dan lindungi AWS akun Anda](#)

Prasyarat

Sebelum menyelesaikan langkah-langkah memulai, pastikan Anda memiliki persyaratan berikut:

- Anda sudah siap dengan Amazon EC2 dan terbiasa meluncurkan instans EC2. Anda memerlukan Akun AWS, pengguna dengan akses administratif, key pair, dan grup keamanan. Untuk informasi selengkapnya, lihat [Memulai Amazon EC2](#) di Panduan Pengguna Amazon EC2.
- Sumber daya Amazon Virtual Private Cloud (Amazon VPC), EC2, dan EFS semuanya sama AWS Region dan Anda memiliki VPC default di Wilayah. Jika Anda tidak memiliki VPC default, atau jika Anda ingin me-mount sistem file Anda dari VPC baru dengan grup keamanan baru atau yang sudah ada, lihat. [Menggunakan grup keamanan VPC](#)
- Anda belum mengubah aturan akses masuk default untuk grup keamanan default.

Anda juga dapat melakukan latihan memulai serupa menggunakan perintah AWS Command Line Interface (AWS CLI) untuk melakukan panggilan EFS API. Untuk informasi selengkapnya, lihat [Tutorial: Buat sistem file EFS dan pasang pada instans EC2 menggunakan AWS CLI](#).

Buat sistem file EFS Anda dan luncurkan instans EC2 Anda

Setelah memastikan bahwa Anda memiliki prasyarat untuk memulai latihan ini, Anda dapat membuat sistem file EFS dan meluncurkan instans EC2 Anda. Cara tercepat untuk menyelesaikan semua langkah yang diperlukan untuk memulai sistem file EFS pertama Anda adalah dengan menggunakan wizard peluncuran baru EC2 selama peluncuran instans.

 Note

Anda tidak dapat menggunakan Amazon EFS dengan instans EC2 berbasis Microsoft Windows.

Untuk membuat sistem file EFS Anda dan meluncurkan instans EC2 Anda menggunakan wizard peluncuran EC2

Untuk petunjuk cara membuat dan memasang sistem file EFS saat membuat peluncuran instans EC2, lihat [Menggunakan Amazon EFS dengan Amazon](#) EC2 di Panduan Pengguna Amazon EC2.

Berikut ini adalah langkah-langkah yang akan Anda lakukan saat membuat sistem file EFS selama peluncuran instance.

1. Buat instans EC2 yang berjalan pada sistem operasi Linux menggunakan key pair dan pengaturan jaringan yang Anda pilih.
2. Buat sistem file EFS bersama yang memiliki pengaturan yang disarankan dan secara otomatis dipasang ke instans EC2.
3. Luncurkan instans EC2 sehingga sistem file EFS tersedia untuk transfer file.

Atau, di konsol Amazon EFS, Anda dapat membuat sistem file dengan pengaturan yang disarankan atau pengaturan khusus. Anda juga dapat menggunakan AWS CLI dan API untuk membuat sistem file. Untuk informasi selengkapnya tentang semua opsi Anda untuk membuat sistem file, lihat [Membuat sistem file EFS](#).

Transfer file ke sistem file EFS Anda menggunakan AWS DataSync

Setelah membuat sistem file EFS, Anda dapat mentransfer file ke sana dari sistem file yang ada dengan menggunakan AWS DataSync. DataSync adalah layanan transfer data yang menyederhanakan, mengotomatisasi, dan mempercepat pemindahan dan replikasi data antara sistem penyimpanan lokal dan layanan AWS penyimpanan melalui internet atau Direct Connect. DataSync dapat mentransfer data file Anda, dan juga metadata sistem file seperti kepemilikan, cap waktu, dan izin akses.

Untuk informasi selengkapnya tentang DataSync, lihat [AWS DataSync](#).

Prasyarat

Sebelum mentransfer file ke sistem file EFS, pastikan Anda memiliki yang berikut:

- Sistem file NFS sumber tempat Anda dapat mentransfer file dari. Sistem sumber ini harus dapat diakses melalui NFS versi 3, versi 4, atau 4.1. Contoh sistem file termasuk yang terletak di pusat data lokal, sistem file in-cloud yang dikelola sendiri, dan sistem file EFS.
- Anda diatur untuk digunakan DataSync. Untuk mempelajari lebih lanjut, lihat [Memulai AWS DataSync](#) di Panduan AWS DataSync Pengguna.

Untuk mentransfer file ke sistem file EFS Anda menggunakan AWS DataSync

Untuk petunjuk penggunaan DataSync untuk mentransfer file ke sistem file EFS, lihat [Mentransfer data Anda AWS DataSync](#) di Panduan AWS DataSync Pengguna.

Berikut ini adalah langkah-langkah yang akan Anda lakukan saat mentransfer file ke sistem file EFS menggunakan DataSync.

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Unduh, terapkan, dan aktifkan agen di lingkungan Anda.
3. Buat dan konfigurasi sumber dan lokasi tujuan.
4. Buat dan konfigurasi tugas.
5. Jalankan tugas untuk mentransfer file dari sumber ke tujuan.

Bersihkan sumber daya dan lindungi AWS akun Anda

Ketika Anda selesai dengan latihan memulai ini, lakukan langkah-langkah berikut untuk membersihkan sumber daya Anda dan melindungi Anda Akun AWS.

Untuk membersihkan sumber daya dan melindungi akun Anda

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Lepaskan sistem file EFS dengan perintah berikut.

```
$ sudo umount efs
```


3. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
4. Hapus sistem file EFS yang Anda buat pada langkah pertama latihan memulai.
 - a. Pilih sistem file EFS yang ingin Anda hapus dari daftar sistem file.
 - b. Untuk Tindakan, pilih Hapus sistem file.
 - c. Di kotak dialog Hapus sistem file secara permanen, ketik ID sistem file untuk sistem file EFS yang ingin Anda hapus, lalu pilih Hapus Sistem File.
5. Hentikan instans EC2 yang Anda luncurkan untuk latihan memulai ini. Untuk petunjuk, lihat [Menghentikan instans Amazon EC2](#) di AWS IAM Identity Center Panduan Pengguna.
6. Jika Anda membuat grup keamanan untuk memulai latihan ini, hapus. Untuk petunjuk, lihat [Menghapus grup keamanan](#) di Panduan AWS IAM Identity Center Pengguna.

**Warning**

Jangan hapus grup keamanan default untuk VPC Anda.

Menginstal klien Amazon EFS

Kami menyarankan Anda menginstal klien Amazon EFS (`amazon-efs-utils`), kumpulan alat sumber terbuka untuk Amazon EFS. Klien Amazon EFS menyertakan mount helper, yang merupakan program yang membantu menyederhanakan pemasangan sistem file EFS. Klien juga memungkinkan kemampuan untuk menggunakan Amazon CloudWatch untuk memantau status pemasangan sistem file EFS, dan itu termasuk perkakas yang membuatnya lebih mudah untuk melakukan enkripsi data dalam perjalanan untuk sistem file Amazon EFS.

[Anda dapat menginstal klien Amazon EFS secara manual di instans Amazon EC2 \(EC2\) yang menjalankan distribusi yang didukung.](#) Untuk sistem operasi tertentu yang didukung, Anda dapat mengonfigurasi AWS Systems Manager untuk menginstal atau memperbarui paket secara otomatis. Untuk daftar distribusi yang dapat Anda gunakan AWS Systems Manager, lihat [Sistem operasi yang didukung Systems Manager](#).

Important

Kami menyarankan Anda selalu menggunakan versi terbaru `amazon-efs-utils` untuk memastikan akses ke kemampuan penuhnya. Misalnya, pemasangan dengan IPv6 alamat didukung di versi 2.3 atau yang lebih baru, tetapi tidak di versi sebelumnya.

Topik

- [Dependensi untuk alat EFS](#)
- [Distribusi yang didukung](#)
- [Menginstal klien Amazon EFS secara manual](#)
- [Secara otomatis menginstal atau memperbarui klien Amazon EFS menggunakan AWS Systems Manager](#)
- [Instalasi dan upgrade botocore](#)
- [Upgrade stunnel](#)
- [Mengaktifkan mode FIPS](#)

Dependensi untuk alat EFS

Dependensi berikut ada untuk `amazon-efs-utils` dan diinstal saat Anda menginstal paket: `amazon-efs-utils`

- Klien NFS
 - `nfs-utils` untuk distribusi RHEL, CentOS, Amazon Linux, dan Fedora
 - `nfs-common` untuk distribusi Debian dan Ubuntu
- Relai jaringan (paket `stunnel`, versi 4.56 atau yang lebih baru)
- Python (versi 3.4 atau yang lebih baru)
- OpenSSL 1.0.2 atau yang lebih baru

Note

Secara default, saat menggunakan EFS mount helper dengan Transport Layer Security (TLS), mount helper memberlakukan pemeriksaan nama host sertifikat. EFS mount helper menggunakan `stunnel` program untuk fungsionalitas TLS-nya. Beberapa versi Linux tidak menyertakan versi `stunnel` yang mendukung fitur TLS ini secara default. Saat menggunakan salah satu versi Linux tersebut, pemasangan sistem file EFS menggunakan TLS gagal.

Setelah Anda menginstal `amazon-efs-utils` paket, tingkatkan `stunnel`. Lihat [Upgrade stunnel](#).

Anda dapat menggunakannya AWS Systems Manager untuk mengelola klien Amazon EFS dan mengotomatiskan tugas yang diperlukan untuk menginstal atau memperbarui `amazon-efs-utils` paket pada instans EC2 Anda. Untuk informasi selengkapnya, lihat [Secara otomatis menginstal atau memperbarui klien Amazon EFS menggunakan AWS Systems Manager](#).

Untuk masalah dengan enkripsi, lihat [Enkripsi pemecahan masalah](#).

Distribusi yang didukung

Klien Amazon EFS telah diverifikasi terhadap distribusi Linux dan Mac berikut:

Distribusi	Jenis Package	initsistem
Amazon Linux 2023 () AL2023	rpm	systemd
Amazon Linux 2 (AL2)	rpm	systemd
CentOS 8	rpm	systemd
Amazon Linux 1 (AL1) 2017.09	rpm	upstart
 Note AL1 AMI mencapai end-of-life pada 31 Desember 2023 dan tidak didukung untuk amazon-efs-utils paket yang dirilis pada April 2024 atau lebih baru (versi 2.0 dan yang lebih baru).		
Debian 11	deb	systemd
Fedora 29 - 32	rpm	systemd
macOS Big Sur		launchd
macOS Monterey		launchd
macOS Ventura		launchd
macOS Sonoma		launchd
Lompatan openSUSE, Tumbleweed	rpm	systemd
Oracle8	rpm	systemd
Perusahaan Topi Merah Linux (RHEL) 8, 9	rpm	systemd

Distribusi	Jenis Package	initsistem
SUSE Linux Server Perusahaan (SLES) 12, 15	rpm	systemd
Ubuntu 16.04 LTS, 18.04 LTS, 20.04 LTS, 22.04 LTS	deb	systemd

Untuk daftar lengkap distribusi yang didukung yang paket telah diverifikasi, lihat `amazon-efs-utils` [README](#) di Github.

Menginstal klien Amazon EFS secara manual

Anda dapat menginstal klien Amazon EFS secara manual di instans Linux Amazon EC2 (EC2) dan pada instans EC2 Mac yang menjalankan macOS Big Sur, macOS Monterey, dan macOS Ventura. Untuk daftar distribusi yang mendukung klien Amazon EFS, lihat [Distribusi yang didukung](#)

Prosedur instalasi untuk sistem operasi yang didukung dijelaskan di bagian berikut.

Topik

- [Menginstal klien Amazon EFS di instans Amazon EC2 Linux](#)
- [Menginstal klien Amazon EFS di distribusi Linux lainnya](#)
- [Menginstal klien Amazon EFS di instans EC2 Mac yang menjalankan macOS Big Sur, macOS Monterey, atau macOS Ventura](#)

Menginstal klien Amazon EFS di instans Amazon EC2 Linux

`amazon-efs-utils` Paket untuk menginstal di Amazon EC2 Linux instans dari lokasi berikut:

- Repositori paket Amazon Machine Images (AMI) untuk Amazon Linux. Petunjuk berikut adalah untuk menginstal `amazon-efs-utils` paket dari repositori paket AMI.
- AWS [efs-utils](#) GitHub Repositori. Untuk informasi lebih lanjut tentang menginstal `amazon-efs-utils` paket dari GitHub, lihat [Menginstal klien Amazon EFS di distribusi Linux lainnya](#).

 Note

- Jika Anda menggunakan Direct Connect, Anda dapat menemukan petunjuk instalasi di [Prasyarat](#).
- Amazon Linux 1 (AL1) AMI mencapai puncaknya end-of-life pada 31 Desember 2023 dan tidak didukung untuk `amazon-efs-utils` paket yang dirilis pada April 2024 dan yang lebih baru (versi 2.0 dan yang lebih baru). Kami menyarankan Anda meningkatkan aplikasi ke Amazon Linux 2023 (AL2023), yang mencakup dukungan jangka panjang hingga 2028.

Untuk menginstal **amazon-efs-utils** paket dari repositori paket AMI pada instance EC2 Linux

1. Pastikan Anda telah membuat instans EC2 AL2023 atau Amazon Linux 2 (AL2). Untuk informasi tentang cara melakukannya, lihat [Langkah 1: Meluncurkan instance](#).
2. Akses terminal untuk instans Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
3. Jalankan perintah berikut untuk menginstal paket `amazon-efs-utils`.

```
sudo yum install -y amazon-efs-utils
```

Menginstal klien Amazon EFS di distribusi Linux lainnya

Jika Anda tidak ingin mendapatkan `amazon-efs-utils` paket dari repositori paket AMI Amazon Linux, itu juga tersedia di [GitHub](#)

Setelah Anda mengkloning paket, Anda dapat membangun dan menginstal `amazon-efs-utils` menggunakan salah satu metode berikut, tergantung pada jenis paket yang didukung oleh distribusi Linux Anda:

- RPM — Jenis paket ini didukung oleh AL2023, Amazon Linux 2, Red Hat Linux, CentOS, dan sejenisnya.
- DEB — Jenis paket ini didukung oleh Ubuntu, Debian, dan sejenisnya.

Untuk petunjuk tentang cara menginstal `amazon-efs-utils` paket untuk distribusi Linux lainnya, lihat [Pada distribusi Linux lainnya di](#) `amazon-efs-utils` README di Github.

Menginstal klien Amazon EFS di instans EC2 Mac yang menjalankan macOS Big Sur, macOS Monterey, atau macOS Ventura

`amazon-efs-utils` Paket ini tersedia untuk instalasi pada instans EC2 Mac yang menjalankan macOS Big Sur, macOS Monterey, atau macOS Ventura.

Untuk petunjuk cara menginstal `amazon-efs-utils` paket pada instance Mac, lihat [Di macOS Big Sur, macOS Monterey, macOS Sonoma, dan distribusi macOS Ventura](#) di README di Github.

`amazon-efs-utils`

Langkah selanjutnya

Setelah menginstal `amazon-efs-utils` pada instans EC2 Anda, lanjutkan ke langkah berikutnya untuk memasang sistem file Anda:

- [Instal botocore](#) sehingga Anda dapat menggunakan Amazon CloudWatch untuk memantau status pemasangan sistem file Anda.
- [Tingkatkan ke versi terbaru stunnel](#) untuk mengaktifkan enkripsi data dalam perjalanan.
- [Pasang sistem file Anda](#) menggunakan EFS mount helper.

Secara otomatis menginstal atau memperbarui klien Amazon EFS menggunakan AWS Systems Manager

Anda dapat menggunakan AWS Systems Manager untuk menyederhanakan pengelolaan klien Amazon EFS (`amazon-efs-utils`). AWS Systems Manager adalah AWS layanan yang dapat Anda gunakan untuk melihat dan mengontrol infrastruktur Anda AWS. Dengan AWS Systems Manager Anda dapat mengotomatiskan tugas yang diperlukan untuk menginstal atau memperbarui `amazon-efs-utils` paket pada instans Amazon EC2 (EC2) Anda. Kemampuan Systems Manager seperti Distributor dan State Manager memungkinkan Anda mengotomatiskan proses berikut:

- Mempertahankan kontrol versi atas klien Amazon EFS.
- Menyimpan secara terpusat dan mendistribusikan klien Amazon EFS secara terpusat ke instans Amazon EC2 Anda.
- Otomatiskan proses menjaga instans EC2 Anda dalam keadaan yang ditentukan.

Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS Systems Manager](#).

Apa yang dilakukan klien Amazon EFS selama instalasi

Anda menggunakan klien Amazon EFS untuk mengotomatiskan pemantauan CloudWatch log Amazon untuk status pemasangan sistem file dan meningkatkan `stunnel` ke versi terbaru untuk distribusi Linux yang dipilih. Saat Anda menginstal klien Amazon EFS di instans Amazon EC2 menggunakan Systems Manager, diperlukan tindakan berikut:

- Menginstal `botocore` paket menggunakan langkah yang sama yang dijelaskan dalam [Instalasi dan upgrade botocore](#). Klien Amazon EFS digunakan `botocore` untuk memantau status pemasangan sistem file EFS.
- Memungkinkan pemantauan status pemasangan sistem file EFS di CloudWatch log dengan memperbarui `efs-utils.conf`. Untuk informasi selengkapnya, lihat [Memantau keberhasilan dan kegagalan upaya pemasangan](#).
- Untuk instans EC2 yang berjalan RHEL7 atau CentOS7, klien Amazon EFS secara otomatis memutakhirkan `stunnel` seperti yang dijelaskan dalam [Upgrade stunnel](#). Pemutakhiran `stunnel` diperlukan agar berhasil memasang sistem file EFS menggunakan TLS, dan `stunnel` versi yang dikirimkan bersama RHEL7 dan CentOS7 tidak mendukung klien Amazon EFS ().
`amazon-efs-utils`

Sistem operasi yang didukung Systems Manager

Instans EC2 Anda harus menjalankan salah satu sistem operasi berikut agar dapat digunakan untuk memperbarui atau menginstal klien Amazon EFS secara otomatis. AWS Systems Manager

Platform	Versi Platform	Arsitektur
Amazon Linux 2023 () AL2023	AL2023	x86_64, arm64 (prosesor Graviton2 atau yang lebih baru)
Amazon Linux 2 (AL2)	2.0	x86_64, arm64 (Amazon Linux 2, jenis instans A1)
Amazon Linux 1 (AL1)	2017.09, 2018.03	x86_64

Platform	Versi Platform	Arsitektur
 Note Amazon Linux 1 (AL1) AMI mencapai puncaknya end-of-life pada 31 Desember 2023 dan tidak didukung untuk amazon-efs-utils paket yang dirilis pada April 2024 dan yang lebih baru (versi 2.0 dan yang lebih baru). Kami menyarankan Anda meningkatkan aplikasi ke Amazon Linux 2023 (AL2023), yang mencakup dukungan jangka panjang hingga 2028.		
CentOS	7, 8	x86_64
Red Hat Enterprise Linux (RHEL)	8, 9	x86_64, arm64
Server Perusahaan SUSE Linux (SLES)	12, 15	x86_64
Ubuntu Server	16.04, 18.04, 20.04	x86_64, arm64 (Ubuntu Server 16 dan yang lebih baru, jenis instans A1)

Mengkonfigurasi AWS Systems Manager untuk menginstal klien EFS

Ada dua konfigurasi satu kali yang diperlukan untuk mengatur Systems Manager untuk menginstal atau memperbarui paket secara otomatis. `amazon-efs-utils`

1. Konfigurasi profil instans AWS Identity and Access Management (IAM) dengan izin yang diperlukan.
2. Konfigurasi Asosiasi (termasuk jadwal) yang digunakan untuk instalasi atau pembaruan oleh Manajer Negara.

Langkah 1: Konfigurasi profil instans IAM dengan izin yang diperlukan

Secara default, AWS Systems Manager tidak memiliki izin untuk mengelola klien Amazon EFS Anda dan menginstal atau memperbarui `amazon-efs-utils` paket. Anda harus memberikan akses ke Systems Manager dengan menggunakan profil instans AWS Identity and Access Management (IAM). Profil instans adalah wadah yang meneruskan informasi peran IAM ke instans Amazon EC2 (EC2) saat diluncurkan.

Gunakan kebijakan izin `AmazonElasticFileSystemsUtils` AWS terkelola untuk menetapkan izin yang sesuai untuk peran. Anda dapat membuat peran baru untuk profil instans atau menambahkan kebijakan `AmazonElasticFileSystemsUtils` izin ke peran yang ada. Anda kemudian harus menggunakan profil instans ini untuk meluncurkan instans EC2 Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi izin instans yang diperlukan untuk Systems Manager](#).

Langkah 2: Konfigurasi asosiasi yang digunakan oleh State Manager

`amazon-efs-utils` Paket ini disertakan dengan Distributor dan siap untuk Anda gunakan ke instans EC2 terkelola. Untuk melihat versi terbaru `amazon-efs-utils` yang tersedia untuk instalasi, Anda dapat menggunakan AWS Systems Manager konsol atau alat baris perintah AWS pilihan Anda. Untuk mengakses Distributor, buka <https://console.aws.amazon.com/systems-manager/> dan pilih Distributor di panel navigasi kiri. Temukan Amazon EFSUtils di bagian Dimiliki oleh Amazon. Pilih Amazon EFSUtils untuk melihat detail paket. Untuk informasi selengkapnya, lihat [Melihat paket](#).

Menggunakan State Manager, Anda dapat menginstal atau memperbarui `amazon-efs-utils` paket pada instans EC2 terkelola Anda segera atau sesuai jadwal. Selain itu, Anda dapat memastikan bahwa `amazon-efs-utils` secara otomatis diinstal pada instans EC2 baru. Untuk informasi lebih lanjut tentang pemasangan atau pembaruan paket menggunakan Distributor dan Manajer Negara, lihat [Bekerja dengan Distributor](#).

Untuk menginstal atau memperbarui amazon-efs-utils paket secara otomatis pada instance menggunakan konsol Systems Manager, lihat [Menjadwalkan instalasi atau pembaruan paket \(konsol\)](#). Ini akan meminta Anda untuk membuat asosiasi untuk Manajer Negara, yang mendefinisikan status yang ingin Anda terapkan ke serangkaian instance. Gunakan masukan berikut saat membuat asosiasi:

- Untuk Parameter pilih Action > Install and Installation Type > In-place update.
- Untuk Target, pengaturan yang disarankan adalah Pilih semua instans untuk mendaftarkan semua instans EC2 baru dan yang sudah ada sebagai target untuk menginstal atau memperbarui Amazon secara otomatis. EFSUtils Atau, Anda dapat menentukan tag instance, memilih instance secara manual, atau memilih grup sumber daya untuk menerapkan asosiasi ke subset instance. Jika Anda menentukan tag instans, Anda harus meluncurkan instans EC2 dengan tag untuk memungkinkan AWS Systems Manager menginstal atau memperbarui klien Amazon EFS secara otomatis.
- Untuk Tentukan jadwal pengaturan yang disarankan untuk Amazon EFSUtils adalah setiap 30 hari. Anda dapat menggunakan kontrol untuk membuat cron atau jadwal tarif untuk asosiasi.

Untuk digunakan AWS Systems Manager untuk me-mount sistem file EFS ke beberapa instans EC2, lihat. [Memasang EFS ke beberapa instans EC2](#)

Instalasi dan upgrade **botocore**

Klien Amazon EFS menggunakan botocore untuk berinteraksi dengan AWS layanan lain. Ini diperlukan jika Anda ingin memantau keberhasilan atau kegagalan upaya pemasangan untuk sistem file EFS Anda di CloudWatch Log. Untuk informasi selengkapnya, lihat [Memantau keberhasilan dan kegagalan upaya pemasangan](#).

Untuk petunjuk tentang cara menginstal dan meningkatkan botocore, lihat [Menginstal botocore](#) di amazon-efs-utils README di Github.

Upgrade **stunnel**

Enkripsi data dalam perjalanan dengan EFS mount helper memerlukan OpenSSL versi 1.0.2 atau yang lebih baru, dan versi stunnel yang mendukung pemeriksaan Online Certificate Status Protocol (OCSP) dan sertifikat hostname. EFS mount helper menggunakan stunnel program untuk fungsionalitas TLS-nya. Perhatikan bahwa beberapa versi Linux tidak menyertakan versi stunnel yang mendukung fitur TLS ini secara default. Saat menggunakan salah satu distribusi Linux tersebut, pemasangan sistem file EFS menggunakan TLS gagal.

Setelah menginstal EFS mount helper, Anda dapat meningkatkan versi stunnel sistem Anda dengan instruksi berikut.

Untuk meningkatkan **stunnel** di Amazon Linux, Amazon Linux 2, dan distribusi Linux lain yang didukung (kecuali untuk [SLES 12](#))

1. Di browser web, buka halaman stunnel unduhan <https://stunnel.org/downloads.html>.
2. Temukan stunnel versi terbaru yang tersedia dalam tar.gz format. Perhatikan nama file karena Anda akan membutuhkannya dalam langkah-langkah berikut.
3. Buka terminal pada klien Linux Anda, dan jalankan perintah berikut dalam urutan yang disajikan.

a. Untuk RPM:

```
sudo yum install -y gcc openssl-devel tcp_wrappers-devel
```

Untuk DEB:

```
sudo apt-get install build-essential libwrap0-dev libssl-dev
```

- b. Ganti *latest-stunnel-version* dengan nama file yang Anda catat sebelumnya di Langkah 2.

```
sudo curl -o latest-stunnel-version.tar.gz https://www.stunnel.org/downloads/latest-stunnel-version.tar.gz
```

- c.

```
sudo tar xvfz latest-stunnel-version.tar.gz
```

- d.

```
cd latest-stunnel-version/
```

- e.

```
sudo ./configure
```

- f.

```
sudo make
```

- g. stunnelPaket saat ini diinstal di bin/stunnel. Agar versi baru dapat diinstal, hapus direktori itu dengan perintah berikut.

```
sudo rm /bin/stunnel
```

- h. Instal versi terbaru:

```
sudo make install
```

i. Buat symlink:

```
sudo ln -s /usr/local/bin/stunnel /bin/stunnel
```

Untuk meningkatkan stunnel di macOS

- Buka terminal pada instans EC2 Mac Anda, dan jalankan perintah berikut untuk meningkatkan ke versi terbaru stunnel.

```
brew upgrade stunnel
```

Upgrade stunnel untuk SLES 12

- Jalankan perintah berikut dan ikuti instruksi pengelola paket zypper untuk meningkatkan stunnel pada instance komputasi yang sedang berjalan. SLES12

```
sudo zypper addrepo https://download.opensuse.org/repositories/security:Stunnel/  
SLE_12_SP5/security:Stunnel.repo  
sudo zypper refresh  
sudo zypper install -y stunnel
```

Setelah menginstal versi stunnel dengan fitur yang diperlukan, Anda dapat memasang sistem file menggunakan TLS dengan pengaturan yang direkomendasikan Amazon EFS.

Menyelesaikan masalah dengan menginstal stunnel

Jika Anda tidak dapat menginstal stunnel, coba nonaktifkan pemeriksaan nama host sertifikat. Selain itu, berikan keamanan sekuat mungkin dengan mengaktifkan Online Certificate Status Protocol (OCSP).

Topik

- [Menonaktifkan Pemeriksaan Nama Host Sertifikat](#)
- [Mengaktifkan Protokol Status Sertifikat Online](#)

Menonaktifkan Pemeriksaan Nama Host Sertifikat

Jika Anda tidak dapat menginstal dependensi yang diperlukan, Anda dapat menonaktifkan pemeriksaan nama host sertifikat di dalam konfigurasi mount helper Amazon EFS. Kami tidak menyarankan Anda menonaktifkan fitur ini di lingkungan produksi. Untuk menonaktifkan pemeriksaan nama host sertifikat, lakukan hal berikut:

1. Akses terminal untuk instans EC2 Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Menggunakan editor teks pilihan Anda, buka `/etc/amazon/efs/efs-utils.conf` file.
3. Tetapkan `stunnel_check_cert_hostname` nilainya ke `false`.
4. Simpan perubahan ke file dan tutup.

Untuk informasi selengkapnya tentang penggunaan enkripsi data dalam perjalanan, lihat [Memasang sistem file EFS](#).

Mengaktifkan Protokol Status Sertifikat Online

Untuk memaksimalkan ketersediaan sistem file jika CA tidak dapat dijangkau dari VPC Anda, Protokol Status Sertifikat Online (OCSP) tidak diaktifkan secara default saat Anda memilih untuk mengenkripsi data dalam perjalanan. Amazon EFS menggunakan [otoritas sertifikat Amazon](#) (CA) untuk menerbitkan dan menandatangani sertifikat TLS, dan CA menginstruksikan klien untuk menggunakan OCSP untuk memeriksa sertifikat yang dicabut. Titik akhir OCSP harus dapat diakses melalui Internet dari Virtual Private Cloud Anda untuk memeriksa status sertifikat. Dalam layanan, Amazon EFS terus memantau status sertifikat, dan mengeluarkan sertifikat baru untuk menggantikan sertifikat yang dicabut yang dideteksinya.

Untuk memberikan keamanan sekuat mungkin, Anda dapat mengaktifkan OCSP sehingga klien Linux Anda dapat memeriksa sertifikat yang dicabut. OCSP melindungi dari penggunaan berbahaya sertifikat yang dicabut, yang tidak mungkin terjadi dalam VPC Anda. Jika sertifikat EFS TLS dicabut, Amazon menerbitkan buletin keamanan dan merilis versi baru EFS mount helper yang menolak sertifikat yang dicabut.

Untuk mengaktifkan OCSP pada klien Linux Anda untuk semua koneksi TLS future ke EFS

1. Buka terminal di klien Linux Anda.
2. Menggunakan editor teks pilihan Anda, buka `/etc/amazon/efs/efs-utils.conf` file.

3. Tetapkan `stunnel_check_cert_validity` nilainya ke `true`.
4. Simpan perubahan ke file dan tutup.

Untuk mengaktifkan OCSP sebagai bagian dari perintah **mount**

- Gunakan perintah mount berikut untuk mengaktifkan OCSP saat memasang sistem file.

```
$ sudo mount -t efs -o tls,ocsp fs-12345678:/ /mnt/efs
```

Mengaktifkan mode FIPS

Jika sistem operasi Anda menggunakan titik akhir Federal Information Processing Standards (FIPS) saat memasang sistem file Anda, maka Anda harus mengaktifkan mode FIPS di klien Amazon EFS. Mengaktifkan mode FIPS melibatkan memodifikasi `efs-utils.conf` file pada sistem operasi.

Note

Mode FIPS mengharuskan versi OpenSSL yang diinstal dikompilasi dengan FIPS. Untuk informasi lebih lanjut tentang cara mengkonfigurasi OpenSSL dengan FIPS lihat [OpenSSL FIPS README](#).

Untuk mengaktifkan mode FIPS di klien Amazon EFS

1. Akses terminal untuk instans Amazon EC2 Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.
2. Menggunakan editor teks pilihan Anda, buka `/etc/amazon/efs/efs-utils.conf` file.
3. Temukan baris yang berisi teks berikut:

```
"fips_mode_enabled = false"
```

4. Ubah teks menjadi yang berikut:

```
"fips_mode_enabled = true"
```

5. Simpan perubahan Anda.

Membuat dan mengelola sumber daya EFS

Amazon EFS menyediakan penyimpanan file bersama yang elastis. Sistem file yang Anda buat mendukung akses baca dan tulis bersamaan dari beberapa instans Amazon EC2. Sistem file juga dapat diakses dari semua Availability Zones di AWS Region mana ia dibuat.

Anda dapat memasang sistem file Amazon EFS pada instans EC2 di virtual private cloud (VPC) berdasarkan Amazon VPC dengan menggunakan protokol Network File System versi 4.0 dan 4.1 (). NFSv4 Untuk informasi selengkapnya, lihat [Cara kerja Amazon EFS](#).

Sebagai contoh, misalkan Anda memiliki satu atau lebih instans EC2 yang diluncurkan di VPC Anda. Sekarang Anda ingin membuat dan menggunakan sistem file pada contoh ini. Berikut ini adalah langkah-langkah umum yang harus Anda lakukan untuk menggunakan sistem file Amazon EFS di VPC:

- Buat sistem file Amazon EFS — Saat membuat sistem file, sebaiknya gunakan tag Nama. Nilai tag Nama muncul di konsol dan membuatnya lebih mudah untuk mengidentifikasi sistem file. Anda juga dapat menambahkan tag opsional lainnya ke sistem file.
- Buat target mount untuk sistem file — Untuk mengakses sistem file di VPC Anda dan memasang sistem file ke instans Amazon EC2 Anda, Anda harus membuat target mount di subnet VPC.
- Buat grup keamanan — Instans Amazon EC2 dan target mount harus memiliki grup keamanan terkait. Kelompok keamanan ini bertindak sebagai firewall virtual yang mengontrol lalu lintas di antara mereka. Anda dapat menggunakan grup keamanan yang Anda kaitkan dengan target pemasangan untuk mengontrol lalu lintas masuk ke sistem file Anda. Untuk melakukan ini, tambahkan aturan masuk ke grup keamanan target mount yang memungkinkan akses dari instans EC2 tertentu. Kemudian, Anda dapat me-mount sistem file hanya pada instans EC2 itu.

Topik

- [Ringkasan implementasi](#)
- [Sumber Daya IDs](#)
- [Token pembuatan dan idempotensi](#)
- [Membuat sistem file EFS](#)
- [Menghapus sistem file EFS](#)
- [Membuat kebijakan sistem file](#)
- [Membuat titik akses](#)

- [Menghapus titik akses](#)
- [Menandai sumber daya EFS](#)
- [Tutorial: Membuat subdirektori per pengguna yang dapat ditulis](#)

Ringkasan implementasi

Di Amazon EFS, sistem file adalah sumber daya utama. Setiap sistem file memiliki properti seperti ID, token pembuatan, waktu pembuatan, ukuran sistem file dalam byte, jumlah target pemasangan yang dibuat untuk sistem file, dan kebijakan siklus hidup sistem file.

Amazon EFS juga mendukung sumber daya lain untuk mengonfigurasi sumber daya utama. Ini termasuk target mount dan titik akses:

- Pasang target - Untuk mengakses sistem file Anda, Anda harus membuat target mount di VPC Anda. Setiap target mount memiliki properti berikut: ID target mount, ID subnet tempat ia dibuat, ID sistem file yang membuatnya, alamat IP tempat sistem file dapat dipasang, grup keamanan VPC, dan status target mount. Anda dapat menggunakan alamat IP atau nama DNS dalam mount perintah Anda.

Setiap sistem file memiliki nama DNS dari formulir berikut.

```
file-system-id.efs.aws-region.amazonaws.com
```

Anda dapat menentukan nama DNS ini dalam mount perintah Anda untuk me-mount sistem file Amazon EFS. Misalkan Anda membuat `efs-mount-point` subdirektori dari direktori home Anda di instans EC2 atau server lokal. Kemudian, Anda dapat menggunakan perintah mount untuk me-mount sistem file. Misalnya, pada Amazon Linux AMI, Anda dapat menggunakan mount perintah berikut.

```
$ sudo mount -t nfs -o  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport file-  
system-DNS-name:/ ~/efs-mount-point
```

Untuk informasi selengkapnya, lihat [Mengelola target mount](#).

- Access Points — Titik akses menerapkan jalur pengguna, grup, dan sistem file sistem operasi ke permintaan sistem file apa pun yang dibuat menggunakan titik akses. Pengguna dan grup sistem operasi titik akses mengesampingkan informasi identitas apa pun yang disediakan oleh klien NFS.

Jalur sistem file diekspos ke klien sebagai direktori root titik akses. Ini memastikan bahwa setiap aplikasi selalu menggunakan identitas sistem operasi yang benar dan direktori yang benar saat mengakses kumpulan data berbasis file bersama. Aplikasi yang menggunakan titik akses hanya bisa mengakses data di direktori sendiri dan di bawah ini. Untuk informasi selengkapnya, lihat [Bekerja dengan titik akses](#).

Target dan tag mount adalah subsumber daya yang terkait dengan sistem file. Anda hanya dapat membuatnya dalam konteks sistem file yang ada.

Amazon EFS menyediakan operasi API bagi Anda untuk membuat dan mengelola sumber daya ini. Selain operasi buat dan hapus untuk setiap sumber daya, Amazon EFS mendukung operasi deskripsi yang memungkinkan Anda mengambil informasi sumber daya. Anda memiliki opsi berikut untuk membuat dan mengelola sumber daya ini:

- Gunakan konsol Amazon EFS — Sebagai contoh, lihat [Memulai](#).
- Gunakan antarmuka baris perintah Amazon EFS (CLI) — Sebagai contoh, lihat [Tutorial: Buat sistem file EFS dan pasang pada instans EC2 menggunakan AWS CLI](#)
- Anda juga dapat mengelola sumber daya ini secara terprogram sebagai berikut:
 - Gunakan AWS SDKs — AWS SDKs Sederhanakan tugas pemrograman Anda dengan membungkus API Amazon EFS yang mendasarinya. Klien SDK juga mengautentikasi permintaan Anda dengan menggunakan kunci akses yang Anda berikan. Untuk informasi selengkapnya, lihat [Contoh Kode dan Pustaka](#).
 - Panggil Amazon EFS API langsung dari aplikasi Anda — Jika Anda tidak dapat menggunakannya karena SDKs alasan tertentu, Anda dapat melakukan panggilan Amazon EFS API langsung dari aplikasi Anda. Namun, Anda perlu menulis kode yang diperlukan untuk mengautentikasi permintaan Anda jika Anda menggunakan opsi ini. Untuk informasi selengkapnya tentang Amazon EFS API, lihat [Amazon EFS API](#).

Sumber Daya IDs

Amazon EFS menetapkan pengidentifikasi sumber daya unik (IDs) ke semua sumber daya EFS saat dibuat. Semua sumber daya EFS IDs terdiri dari pengenalan sumber daya dan kombinasi angka 0—9 dan huruf kecil a—f.

Sebelum Oktober 2021, yang IDs ditugaskan ke sistem file yang baru dibuat dan sumber daya target mount menggunakan 8 karakter setelah tanda hubung (misalnya, fs-12345678). Dari Mei 2021

hingga Oktober 2021, kami mengubah jenis sumber daya ini untuk menggunakan 17 karakter setelah tanda hubung (misalnya, `fs-1234567890abcdef0`). IDs Bergantung pada kapan akun Anda dibuat, Anda mungkin memiliki sistem file dan memasang sumber daya target dengan singkat IDs, meskipun sumber daya baru dari jenis ini menerima lebih lama IDs. ID Sumber Daya tidak pernah berubah.

Token pembuatan dan idempotensi

Idempotency memastikan bahwa permintaan API hanya selesai sekali. Dengan permintaan idempoten, jika permintaan asli berhasil diselesaikan, permintaan berikutnya tidak memiliki efek tambahan. Ini berguna untuk mencegah duplikat pekerjaan dibuat saat Anda berinteraksi dengan Amazon EFS API.

Amazon EFS API mendukung idempotensi dengan token permintaan klien. Token permintaan klien adalah string unik yang Anda tentukan saat Anda membuat permintaan pekerjaan.

Token permintaan klien dapat berupa string apa pun yang mencakup hingga 64 karakter ASCII. Jika Anda menggunakan kembali token permintaan klien dalam waktu satu menit setelah permintaan berhasil, API akan mengembalikan detail pekerjaan dari permintaan asli.

Jika Anda menggunakan konsol, itu menghasilkan token untuk Anda. Jika Anda menggunakan alur Custom Create di konsol, token pembuatan yang dibuat untuk Anda memiliki format berikut:

```
"CreationToken": "console-d215fa78-1f83-4651-b026-facafd8a7da7"
```

Jika Anda menggunakan Quick Create untuk membuat sistem file dengan pengaturan yang direkomendasikan layanan, token pembuatan memiliki format berikut:

```
"CreationToken": "quickCreated-d7f56c5f-e433-41ca-8307-9d9c0f8a77a2"
```

Membuat sistem file EFS

Anda dapat membuat sistem file saat membuat instance peluncuran EC2 baru, seperti yang dijelaskan dalam [latihan Memulai](#). Namun, Anda juga dapat membuat sistem file dengan menggunakan konsol Amazon EFS, AWS Command Line Interface (AWS CLI), atau Amazon EFS API.

Saat membuat sistem file menggunakan konsol Amazon EFS, Anda memiliki opsi berbeda, tergantung pada apakah Anda ingin sistem file menggunakan pengaturan yang disarankan atau jika Anda ingin menyesuaikan pengaturan.

- Anda dapat menggunakan Quick create untuk membuat sistem file dengan cepat dengan pengaturan yang disarankan berikut.
 - Ketersediaan wilayah
 - Kebijakan siklus hidup untuk mentransisikan sistem file ke penyimpanan EFS Infrequent Access (IA) setelah 30 hari, ke penyimpanan EFS Archive setelah 90 hari, dan tidak beralih ke penyimpanan EFS Standard
 - Enkripsi data saat istirahat diaktifkan
 - Mode throughput elastis
 - Mode kinerja Tujuan Umum
 - Pasang target yang dikonfigurasi di setiap Availability Zone AWS Region di mana sistem file dibuat menggunakan yang tersedia IPv4 di subnet yang ditentukan
- Anda dapat menggunakan Sesuaikan untuk membuat sistem file dengan pengaturan yang Anda pilih.

Untuk tabel yang mencantumkan pengaturan sistem file dan nilai yang disarankan, lihat [Opsi konfigurasi untuk sistem file](#).

Izin IAM yang diperlukan untuk membuat sistem file

Untuk membuat sumber daya EFS, seperti sistem file dan titik akses, Anda harus memiliki izin AWS Identity and Access Management (IAM) untuk operasi dan sumber daya API yang sesuai.

Buat pengguna IAM dan beri mereka izin untuk tindakan Amazon EFS dengan kebijakan pengguna. Anda juga dapat menggunakan peran untuk memberikan izin lintas akun. Amazon Elastic File System juga menggunakan peran terkait layanan IAM yang menyertakan izin yang diperlukan untuk memanggil orang lain Layanan AWS atas nama Anda. Untuk informasi selengkapnya tentang mengelola izin untuk operasi API, lihat [Manajemen identitas dan akses untuk Amazon EFS](#).

Opsi konfigurasi untuk sistem file

Sistem file EFS dikonfigurasi dengan pengaturan yang tercantum dalam tabel berikut.

- Jika Anda menggunakan Quick create untuk membuat sistem file, sistem file dibuat dengan nilai yang disarankan yang ditunjukkan untuk pengaturan.
- Jika menggunakan Sesuaikan untuk membuat sistem file kustom, Anda dapat mengubah nilai yang disarankan untuk pengaturan.

Pengaturan sistem file EFS

Pengaturan	Deskripsi
Jenis sistem file	Direkomendasikan: Regional Jenis sistem file menentukan ketersediaan dan daya tahan dengan mana sistem file EFS menyimpan data dalam file file AWS Region. • Sistem file regional menyimpan data dan metadata secara berlebihan di semua Availability Zone dalam file. AWS Region Anda juga dapat membuat target mount di setiap Availability Zone di AWS Region. Regional menawarkan tingkat ketersediaan dan daya tahan tertinggi. • Sistem file One Zone menyimpan data dan metadata secara berlebihan dalam satu Availability Zone. Sistem file One Zone hanya dapat memiliki satu target mount. Target mount ini harus ditempatkan di Availability Zone yang sama di mana sistem file dibuat.  Note Sistem file One Zone hanya tersedia untuk Availability Zone tertentu.

Pengaturan	Deskripsi
	Untuk informasi selengkapnya tentang jenis sistem file, lihat Ketersediaan dan daya tahan sistem file EFS .

Pengaturan	Deskripsi
Manajemen siklus hidup	Direkomendasikan: Transisi ke penyimpanan IA setelah 30 hari dan ke penyimpanan Arsip setelah 90 hari. Jangan beralih ke penyimpanan Standar. Manajemen siklus hidup menggunakan kebijakan siklus hidup untuk secara otomatis memindahkan file masuk dan keluar dari kelas penyimpanan Infrequent Access (IA) berbiaya rendah berdasarkan pola akses. Saat Anda membuat sistem file menggunakan Konsol Manajemen AWS, kebijakan siklus hidup sistem file dikonfigurasi dengan setelan default berikut: • Transisi ke IA diatur ke 30 hari sejak akses terakhir. • Transisi ke Arsip diatur ke 90 hari sejak akses terakhir. • Transisi ke Standar diatur ke Tidak Ada. Saat Anda membuat sistem file menggunakan Amazon EFS API, atau AWS SDKs, Anda tidak dapat menetapkan kebijakan siklus hidup secara bersamaan. AWS CLI Anda harus menunggu hingga sistem file dibuat, lalu gunakan operasi PutLifecycleConfiguration API untuk memperbarui kebijakan siklus hidup. Untuk informasi lebih lanjut tentang manajemen siklus hidup, lihat Mengelola siklus hidup penyimpanan.

Pengaturan	Deskripsi
Enkripsi saat istirahat	Direkomendasikan: Diaktifkan Amazon EFS menggunakan AWS Key Management Service (AWS KMS) EFS service key (aws/elasticfilesystem) untuk mengenkripsi data saat istirahat secara default. Dengan enkripsi, semua data dan metadata yang tersimpan di dalamnya dienkripsi. Setelah Anda membuat sistem file EFS, Anda tidak dapat mengubah pengaturan enkripsi. Untuk mengonversi sistem file yang tidak terenkripsi menjadi terenkripsi, Anda harus membuat sistem file baru. Untuk informasi selengkapnya tentang enkripsi di Amazon EFS, lihat Enkripsi data di Amazon EFS.

Pengaturan	Deskripsi
Mode throughput	Direkomendasikan: Elastis Anda dapat memilih dari mode throughput berikut: • Elastis — Menyediakan throughput yang menskalakan naik dan turun secara otomatis secara real time, untuk memenuhi kebutuhan kinerja beban kerja Anda. • Provisioned — Tentukan jumlah throughput yang Anda inginkan, terlepas dari ukuran sistem file. • Bursting — Menyediakan throughput yang menskalakan dengan jumlah data dalam penyimpanan Standar.  Note Biaya tambahan terkait dengan penggunaan throughput Elastic dan Provisioned. Untuk informasi selengkapnya, lihat harga Amazon EFS. Untuk informasi selengkapnya tentang mode throughput, lihat Mode throughput.

Pengaturan	Deskripsi
Mode kinerja	Direkomendasikan: Tujuan Umum Mode kinerja Tujuan Umum memiliki latensi per operasi terendah dan direkomendasikan untuk semua sistem file. Max I/O adalah tipe kinerja generasi sebelumnya yang dirancang untuk beban kerja yang sangat paralel yang dapat mentolerir latensi yang lebih tinggi daripada mode Tujuan Umum. I/O Mode maks tidak didukung untuk sistem file One Zone atau sistem file yang menggunakan throughput Elastic.  Important Karena latensi per operasi yang lebih tinggi dengan Max I/O, sebaiknya gunakan mode kinerja Tujuan Umum untuk semua sistem file. Untuk informasi selengkapnya, lihat Mode kinerja.

Pengaturan	Deskripsi
Akses jaringan	Direkomendasikan: Pasang target yang dibuat di setiap Availability Zone di mana sistem file tersedia Pengaturan target pemasangan adalah sebagai berikut: • VPC — Anda dapat membuat target mount untuk sistem file dalam satu VPC pada satu waktu. Secara default, subnet default instans EC2 dipilih. Ubah VPC jika Anda perlu mengakses sistem file dari VPC yang berbeda. - Jika Anda ingin mengubah VPC setelah membuat sistem file setelah, Anda harus terlebih dahulu menghapus target mount dari VPC saat ini. Untuk informasi selengkapnya, lihat Mengubah VPC target mount. • Availability Zone dan Subnet ID — Anda dapat mengubah Availability Zone atau subnet untuk target mount. • Jenis alamat IP — Secara default, Amazon EFS menggunakan IPv4 alamat untuk membuat target pemasangan. Anda dapat memilih IPv6 hanya untuk mendukung IPv6 alamat saja, atau Dual-stack untuk mendukung keduanya IPv4 dan IPv6 alamat.  Note Jenis alamat IP harus sesuai dengan jenis IP subnet. Selain itu, jenis alamat IP mengesampingkan atribut pengalamatan IP subnet Anda.

Pengaturan	Deskripsi
	Misalnya, jika jenis alamat IP IPv4-only dan atribut IPv6 pengalamatan diaktifkan untuk subnet Anda, antarmuka jaringan yang dibuat di subnet menerima IPv4 alamat dari kisaran subnet. Untuk informasi selengkapnya, lihat Memodifikasi atribut pengalamatan IP subnet Anda. • Alamat IP — Secara default, Amazon EFS membuat target pemasangan pada alamat yang tersedia di subnet yang ditentukan. Sebagai alternatif, Anda dapat menentukan alamat IP untuk target mount. Anda tidak dapat mengubah alamat IP target mount setelah dibuat. Anda perlu menghapus target mount dan membuat yang baru dengan alamat baru. • Grup keamanan — Grup keamanan yang ditetapkan untuk target VPC digunakan secara default. Anda dapat menambah atau menghapus grup keamanan. Untuk mengetahui informasi selengkapnya tentang grup keamanan, lihat Mengubah grup keamanan target mount. Saat Anda membuat sistem file dengan menggunakan AWS CLI, Amazon EFS API, atau AWS SDKs, Anda tidak dapat membuat target pemasangan secara bersamaan. Anda harus menunggu sampai sistem file dibuat, dan kemudian menggunakan operasi CreateMountTarget API untuk membuat target mount.

Pengaturan	Deskripsi
	Untuk informasi lebih lanjut tentang target pemasangan, lihat Mengelola target mount .

Buat cepat menggunakan konsol

Gunakan konsol Amazon EFS untuk membuat sistem file Amazon EFS yang memiliki pengaturan yang disarankan. Jika Anda ingin membuat sistem file dengan konfigurasi yang disesuaikan, lihat [Buat kustom menggunakan konsol](#).

Untuk membuat sistem file Amazon EFS dengan cepat yang memiliki pengaturan yang disarankan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.
2. Pilih Buat sistem file untuk membuka kotak dialog Buat sistem file.
3. (Opsional) Masukkan Nama untuk sistem file Anda.
4. Untuk Virtual Private Cloud (VPC), pilih VPC Anda, atau tetapkan ke VPC default Anda.
5. Pilih Buat untuk membuat sistem file yang menggunakan pengaturan yang direkomendasikan layanan berikut:
 - Ketersediaan wilayah
 - Kebijakan siklus hidup untuk mentransisikan sistem file ke penyimpanan EFS Infrequent Access (IA) setelah 30 hari, ke penyimpanan EFS Archive setelah 90 hari, dan tidak beralih ke penyimpanan EFS Standard
 - Enkripsi data saat istirahat diaktifkan
 - Mode throughput elastis
 - Mode kinerja Tujuan Umum
 - Pasang target yang dikonfigurasi di setiap Availability Zone AWS Region di mana sistem file dibuat, menggunakan yang tersedia IPv4 di subnet yang ditentukan

Halaman sistem File muncul dengan spanduk di bagian atas yang menunjukkan status sistem file yang Anda buat. Tautan untuk mengakses halaman detail sistem file muncul di spanduk saat sistem file tersedia.

Untuk informasi selengkapnya tentang status sistem file, lihat [Memahami status sistem berkas](#).

Buat kustom menggunakan konsol

Bagian ini menjelaskan proses penggunaan konsol Amazon EFS untuk membuat sistem file EFS dengan pengaturan yang disesuaikan alih-alih menggunakan pengaturan yang direkomendasikan layanan. Untuk informasi selengkapnya tentang membuat sistem file dengan menggunakan pengaturan yang disarankan, lihat [Buat cepat menggunakan konsol](#).

Membuat sistem file EFS dengan pengaturan khusus dengan menggunakan konsol adalah proses empat langkah:

- Langkah 1 - Konfigurasi pengaturan sistem file umum, termasuk kelas penyimpanan dan mode throughput.
- Langkah 2 - Konfigurasi pengaturan jaringan sistem file, termasuk virtual private cloud (VPC) dan mount target. Untuk setiap target pemasangan, atur Availability Zone, subnet, alamat IP, dan grup keamanan.
- Langkah 3 - (Opsional) Buat kebijakan sistem file untuk mengontrol akses klien NFS ke sistem file.
- Langkah 4 - Tinjau pengaturan sistem file, buat perubahan apa pun, lalu buat sistem file.

Langkah 1: Konfigurasi pengaturan sistem file

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.
2. Pilih Buat sistem file untuk membuka kotak dialog Buat sistem file.
3. Pilih Sesuaikan untuk membuat sistem file yang disesuaikan alih-alih membuat sistem file dengan menggunakan pengaturan yang disarankan. Halaman pengaturan sistem file terbuka.
4. Untuk pengaturan Umum, lakukan hal berikut.
 - a. (Opsional) Masukkan Nama untuk sistem file.
 - b. Untuk tipe sistem File, Regional dipilih secara default. Pilih Satu Zona jika Anda ingin membuat sistem file yang menyimpan data sistem file dan metadata secara berlebihan dalam satu Availability Zone. Jika Anda memilih One Zone, pilih Availability Zone yang Anda inginkan untuk membuat sistem file, atau pertahankan nilai default.
 - c. Untuk manajemen Siklus Hidup, ubah kebijakan siklus hidup, jika perlu.
 - Transisi ke IA - Pilih kapan harus mentransisikan file ke kelas penyimpanan Akses Jarang (IA), berdasarkan waktu sejak terakhir diakses di penyimpanan Standar.

- Transisi ke Arsip - Pilih kapan harus mentransisikan file ke kelas penyimpanan Arsip, berdasarkan waktu sejak terakhir diakses di penyimpanan Standar.
- Transisi ke Standar — Pilih apakah akan mentransisikan sistem file ke kelas penyimpanan.

Untuk informasi selengkapnya tentang kebijakan siklus hidup, lihat. [Mengelola siklus hidup penyimpanan](#)

- d. Amazon EFS menggunakan AWS Key Management Service (AWS KMS) EFS service key (aws/elasticfilesystem) untuk mengenkripsi data saat istirahat secara default. Untuk memilih kunci KMS yang berbeda untuk digunakan untuk enkripsi, perluas Sesuaikan pengaturan enkripsi dan pilih kunci dari daftar. Atau, masukkan ID kunci KMS atau Nama Sumber Daya Amazon (ARN) untuk kunci KMS yang ingin Anda gunakan.

Jika Anda perlu membuat kunci baru, pilih Buat AWS KMS key untuk meluncurkan AWS KMS konsol dan buat kunci baru.

Anda dapat mematikan enkripsi data saat istirahat dengan membersihkan kotak centang.

Anda tidak dapat mengubah pengaturan enkripsi setelah sistem file dibuat. Untuk informasi selengkapnya, lihat [Enkripsi data di Amazon EFS](#).

5. Untuk pengaturan Kinerja, lakukan hal berikut:

- a. Untuk mode Throughput, mode Elastis dipilih secara default.
 - Untuk menggunakan throughput yang disediakan, pilih Provisioned, dan, di Provisioned Throughput (MIB/s), masukkan jumlah throughput untuk penyediaan permintaan sistem file. Jumlah Throughput Baca Maksimum ditampilkan tiga kali jumlah throughput yang Anda masukkan.
 - Untuk menggunakan throughput meledak, pilih Bursting.

Setelah Anda memilih mode throughput, perkiraan biaya bulanan untuk sistem file ditampilkan. Anda dapat mengubah mode throughput setelah sistem file tersedia.

Untuk informasi selengkapnya tentang memilih mode throughput yang benar untuk kebutuhan kinerja Anda, lihat [Mode throughput](#).

- b. Untuk mode Kinerja, defaultnya adalah Tujuan Umum. Untuk mengubah mode kinerja, perluas Pengaturan tambahan, lalu pilih Max I/O.

Anda tidak dapat mengubah mode kinerja setelah sistem file tersedia. Untuk informasi selengkapnya, lihat [Mode kinerja](#).

 Important

Karena latensi per operasi yang lebih tinggi dengan Max I/O, sebaiknya gunakan mode kinerja Tujuan Umum untuk semua sistem file.

6. (Opsional) Tambahkan pasangan nilai kunci tag ke sistem file Anda.
7. Pilih Berikutnya untuk mengkonfigurasi akses jaringan untuk sistem file.

Langkah 2: Konfigurasi akses jaringan

Pada Langkah 2, Anda mengonfigurasi pengaturan jaringan sistem file, termasuk VPC dan target mount.

1. Pilih Virtual Private Cloud (VPC) di mana Anda ingin instans EC2 terhubung ke sistem file Anda. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).
2. Untuk target Mount, Anda membuat satu atau beberapa target mount untuk sistem file Anda. Untuk setiap target pemasangan, atur properti berikut:
 - Availability Zone — Secara default, target mount dikonfigurasi di setiap Availability Zone dalam file AWS Region. Jika Anda tidak ingin target pemasangan di Availability Zone tertentu, pilih Hapus untuk menghapus target mount untuk zona tersebut. Buat target pemasangan di setiap Availability Zone yang Anda rencanakan untuk mengakses sistem file Anda — tidak ada biaya untuk melakukannya.
 - Subnet ID - Pilih dari subnet yang tersedia di Availability Zone. Subnet default telah dipilih sebelumnya.
 - Jenis alamat IP — Pilih IPv4 hanya untuk mendukung IPv4 alamat saja, IPv6hanya untuk mendukung IPv6 alamat saja, atau Dual-stack untuk mendukung keduanya IPv4 dan IPv6 alamat.
 - IPv4 atau IPv6 alamat - Jika Anda mengetahui alamat IP tempat Anda ingin menempatkan target pemasangan, masukkan di kotak alamat IP yang cocok dengan jenis alamat IP. Jika Anda tidak menentukan nilai, Amazon EFS memilih alamat IP yang tidak digunakan dari subnet yang ditentukan.

- Grup keamanan — Secara default, Amazon EFS memilih grup keamanan default untuk VPC. Untuk mengubah grup keamanan, hapus grup yang ditetapkan, lalu pilih grup dari daftar Pilih grup keamanan. Anda dapat menentukan satu atau beberapa grup keamanan untuk target pemasangan. Untuk informasi selengkapnya, lihat [Menggunakan grup keamanan VPC](#).
3. Pilih Tambahkan target pemasangan untuk membuat target pemasangan untuk Availability Zone yang tidak memilikinya. Jika target pemasangan dikonfigurasi untuk setiap Availability Zone, pilihan ini tidak tersedia.
 4. Pilih Berikutnya untuk mengatur kebijakan sistem file.

Langkah 3: Buat kebijakan sistem file (opsional)

Secara opsional, Anda dapat membuat kebijakan sistem file untuk sistem file Anda. Kebijakan sistem file EFS adalah kebijakan sumber daya IAM yang Anda gunakan untuk mengontrol akses klien NFS ke sistem file. Untuk informasi selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#).

1. Di opsi Kebijakan, Anda dapat memilih kombinasi apa pun dari kebijakan sistem file yang telah dikonfigurasi sebelumnya:
 - Mencegah akses root secara default — Opsi ini menghapus `ClientRootAccess` dari kumpulan tindakan EFS yang diizinkan.
 - Menerapkan akses hanya-baca secara default — Opsi ini menghapus `ClientWriteAccess` dari kumpulan tindakan EFS yang diizinkan.
 - Mencegah akses anonim — Opsi ini menghapus `ClientMount` dari kumpulan tindakan EFS yang diizinkan.
 - Menerapkan enkripsi dalam perjalanan untuk semua klien — Opsi ini menolak akses ke klien yang tidak terenkripsi.

Bila Anda memilih kebijakan yang telah dikonfigurasi sebelumnya, objek JSON kebijakan akan ditampilkan di panel Editor kebijakan.

2. Gunakan Berikan izin tambahan untuk memberikan izin sistem file ke prinsipal IAM tambahan, termasuk yang lain. Akun AWS Pilih Tambah, dan masukkan ARN utama entitas yang Anda berikan izin. Kemudian pilih Izin yang ingin Anda berikan. Izin tambahan ditampilkan di editor Kebijakan.

3. Anda dapat menggunakan editor Kebijakan untuk menyesuaikan kebijakan yang telah dikonfigurasi sebelumnya atau untuk membuat kebijakan sistem file Anda sendiri. Saat Anda menggunakan editor, opsi kebijakan yang telah dikonfigurasi sebelumnya menjadi tidak tersedia. Untuk menghapus kebijakan sistem file saat ini dan mulai membuat kebijakan baru, pilih Hapus.
4. Pilih Berikutnya untuk meninjau dan membuat sistem file.

Langkah 4: Tinjau dan buat

1. Tinjau setiap grup konfigurasi sistem file. Anda dapat membuat perubahan pada setiap grup saat ini dengan memilih Edit.
2. Pilih Buat sistem file untuk membuat sistem file Anda dan kembali ke halaman Sistem file.

Spanduk di bagian atas menunjukkan bahwa sistem file baru sedang dibuat. Tautan untuk mengakses halaman detail sistem file baru muncul di spanduk saat sistem file tersedia.

Buat menggunakan AWS CLI

Saat Anda menggunakan AWS CLI, Anda membuat sumber daya ini secara berurutan. Pertama, Anda membuat sistem file. Kemudian, Anda dapat membuat target mount dan tag opsional tambahan untuk sistem file dengan menggunakan AWS CLI perintah yang sesuai.

Contoh berikut digunakan `adminuser` untuk nilai-nilai `--profile` parameter. Anda harus menggunakan profil pengguna yang sesuai untuk memberikan kredensial Anda. Untuk selengkapnya, lihat [Prasyarat untuk menggunakan AWS CLI dalam Panduan Pengguna](#). AWS Command Line Interface

- Untuk membuat sistem file terenkripsi dengan backup otomatis diaktifkan, gunakan perintah Amazon EFS `create-file-system` CLI (operasi yang sesuai adalah [CreateFileSystem](#)), seperti yang ditunjukkan berikut.

```
aws efs create-file-system \  
--creation-token creation-token \  
--encrypted \  
--backup \  
--performance-mode generalPurpose \  
--throughput-mode elastic \  
--region aws-region \  
--tags Key=key,Value=value Key=key1,Value=value1 \  

```

```
--profile adminuser
```

Misalnya, `create-file-system` perintah berikut membuat sistem file menggunakan throughput Elastis di `us-west-2` AWS Region file. Perintah menentukan `MyFirstFS` sebagai token penciptaan. Untuk daftar Wilayah AWS tempat Anda dapat membuat sistem file Amazon EFS, lihat [titik akhir dan kuota Amazon EFS](#) di file. Referensi Umum Amazon Web Services

```
aws efs create-file-system \  
--creation-token MyFirstFS \  
--backup \  
--encrypted \  
--performance-mode generalPurpose \  
--throughput-mode elastic \  
--region us-west-2 \  
--tags Key=Name,Value="Test File System" Key=developer,Value=rhoward \  
--profile adminuser
```

Setelah berhasil membuat sistem file, Amazon EFS mengembalikan deskripsi sistem file sebagai JSON, seperti yang ditunjukkan pada contoh berikut.

```
{  
  "OwnerId": "123456789abcd",  
  "CreationToken": "MyFirstFS",  
  "Encrypted": true,  
  "FileSystemId": "fs-c7a0456e",  
  "CreationTime": 1422823614.0,  
  "LifecycleState": "creating",  
  "Name": "Test File System",  
  "NumberOfMountTargets": 0,  
  "SizeInBytes": {  
    "Value": 6144,  
    "ValueInIA": 0,  
    "ValueInStandard": 6144  
    "ValueInArchive": 0  
  },  
  "PerformanceMode": "generalPurpose",  
  "ThroughputMode": "elastic",  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "Test File System"  
    }  
  ]  
}
```

```
]
}
```

- Contoh berikut membuat sistem file yang menggunakan throughput Bursting di us-west-2a Availability Zone dengan menggunakan properti. `availability-zone-name`

```
aws efs create-file-system \
--creation-token MyFirstFS \
--availability-zone-name us-west-2a \
--backup \
--encrypted \
--performance-mode generalPurpose \
--throughput-mode bursting \
--region us-west-2 \
--tags Key=Name,Value="Test File System" Key=developer,Value=rhoward \
--profile adminuser
```

Setelah berhasil membuat sistem file, Amazon EFS mengembalikan deskripsi sistem file sebagai JSON, seperti yang ditunjukkan pada contoh berikut.

```
{
  "AvailabilityZoneId": "usw-az1",
  "AvailabilityZoneName": "us-west-2a",
  "OwnerId": "123456789abcd",
  "CreationToken": "MyFirstFS",
  "Encrypted": true,
  "FileSystemId": "fs-c7a0456e",
  "CreationTime": 1422823614.0,
  "LifecycleState": "creating",
  "Name": "Test File System",
  "NumberOfMountTargets": 0,
  "SizeInBytes": {
    "Value": 6144,
    "ValueInIA": 0,
    "ValueInStandard": 6144,
    "ValueInArchive": 0
  },
  "PerformanceMode": "generalPurpose",
  "ThroughputMode": "bursting",
  "Tags": [
    {
      "Key": "Name",
```

```
    "Value": "Test File System"  
  }  
]  
}
```

Amazon EFS juga menyediakan perintah `describe-file-systems` CLI (operasi API yang sesuai [DescribeFileSystems](#)), yang dapat Anda gunakan untuk mengambil daftar sistem file di akun Anda, seperti yang ditunjukkan berikut.

```
aws efs describe-file-systems \  
--region aws-region \  
--profile adminuser
```

Amazon EFS mengembalikan daftar sistem file yang Anda Akun AWS buat di Wilayah yang ditentukan.

Menghapus sistem file EFS

Penghapusan sistem file adalah tindakan destruktif yang tidak dapat Anda batalkan. Anda kehilangan sistem file dan data apa pun yang Anda miliki di dalamnya. Data apa pun yang Anda hapus dari sistem file hilang, dan Anda tidak dapat memulihkan data. Ketika pengguna menghapus data dari sistem file, data tersebut segera dirender tidak dapat digunakan. EFS memaksa menimpa data dengan cara yang akhirnya.

Note

Anda tidak dapat menghapus sistem file yang merupakan bagian dari konfigurasi replikasi. Anda harus menghapus konfigurasi replikasi terlebih dahulu. Untuk informasi selengkapnya, lihat [Menghapus konfigurasi replikasi](#).

Important

Anda harus selalu melepas sistem file sebelum Anda menghapusnya.

Menggunakan konsol

Untuk menghapus sistem file

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Pilih sistem file yang ingin Anda hapus di halaman Sistem file.
3. Pilih Hapus.
4. Dalam kotak dialog Hapus sistem file, masukkan ID sistem file yang ditampilkan, dan pilih Konfirmasi untuk mengonfirmasi penghapusan.

Konsol menyederhanakan penghapusan sistem file untuk Anda. Pertama menghapus target mount terkait, dan kemudian menghapus sistem file.

Menggunakan AWS CLI

Sebelum Anda dapat menggunakan AWS CLI perintah untuk menghapus sistem file, Anda harus menghapus semua target mount dan titik akses yang dibuat untuk sistem file.

Misalnya AWS CLI perintah, lihat [Langkah 4: Membersihkan](#).

Membuat kebijakan sistem file

Anda dapat membuat kebijakan sistem file dengan menggunakan konsol Amazon EFS atau dengan menggunakan AWS CLI. Anda juga dapat membuat kebijakan sistem file secara terprogram dengan menggunakan AWS SDKs atau Amazon EFS API secara langsung. Kebijakan sistem file EFS memiliki batas 20.000 karakter. Untuk informasi selengkapnya tentang menggunakan kebijakan dan contoh sistem file EFS, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#).

Note

Perubahan kebijakan sistem file Amazon EFS dapat memakan waktu beberapa menit untuk diterapkan.

Menggunakan konsol

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Pilih Sistem File.

3. Pada halaman Sistem file, pilih sistem file yang akan diedit atau dibuatkan kebijakan sistem file.
4. Pilih Kebijakan sistem berkas, lalu pilih Edit.
5. Di opsi Kebijakan, Anda dapat memilih kombinasi apa pun dari kebijakan sistem file yang telah dikonfigurasi sebelumnya:
 - Mencegah akses root secara default — Opsi ini menghapus `ClientRootAccess` dari kumpulan tindakan EFS yang diizinkan.
 - Menerapkan akses hanya-baca secara default — Opsi ini menghapus `ClientWriteAccess` dari kumpulan tindakan EFS yang diizinkan.
 - Mencegah akses anonim — Opsi ini menghapus `ClientMount` dari kumpulan tindakan EFS yang diizinkan.
 - Menerapkan enkripsi dalam perjalanan untuk semua klien — Opsi ini menolak akses ke klien yang tidak terenkripsi.

Bila Anda memilih kebijakan yang telah dikonfigurasi sebelumnya, objek JSON kebijakan akan ditampilkan di panel Editor kebijakan.

6. Gunakan Berikan izin tambahan untuk memberikan izin sistem file ke prinsipal IAM tambahan, termasuk yang lain. Akun AWS Pilih Tambah, dan masukkan ARN utama entitas yang Anda berikan izin. Kemudian pilih Izin yang ingin Anda berikan. Izin tambahan ditampilkan di editor Kebijakan.
7. Anda dapat menggunakan editor Kebijakan untuk menyesuaikan kebijakan yang telah dikonfigurasi sebelumnya atau untuk membuat kebijakan sistem file Anda sendiri. Saat Anda menggunakan editor, opsi kebijakan yang telah dikonfigurasi sebelumnya menjadi tidak tersedia. Untuk menghapus kebijakan sistem file saat ini dan mulai membuat kebijakan baru, pilih Hapus.

Saat Anda menghapus editor, kebijakan yang telah dikonfigurasi akan tersedia sekali lagi.

8. Setelah Anda menyelesaikan pengeditan kebijakan, pilih Simpan.

Menggunakan AWS CLI

Dalam contoh berikut, perintah [put-file-system-policy](#) CLI membuat kebijakan sistem file yang memungkinkan akses Akun AWS read-only yang ditentukan ke sistem file EFS. Perintah API yang setara adalah [PutFileSystemPolicy](#).

```
aws efs put-file-system-policy --file-system-id fs-01234567 --policy '{
  "Id": "1",
```



```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "elasticfilesystem:ClientMount"  
    ],  
    "Principal": {  
      "AWS": "arn:aws:iam::111122223333:root"  
    }  
  }  
]  
'
```

Membuat titik akses

Anda dapat membuat dan menghapus jalur akses Amazon EFS menggunakan Konsol Manajemen AWS, the AWS Command Line Interface (AWS CLI), dan Amazon EFS API dan SDKs. Anda tidak dapat mengubah titik akses setelah dibuat. Sistem file dapat memiliki maksimum 10.000 titik akses kecuali Anda meminta peningkatan.

Note

Jika beberapa permintaan untuk membuat titik akses pada sistem file yang sama dikirim secara berurutan, dan sistem file mendekati batas titik akses, Anda mungkin mengalami respons pelambatan untuk permintaan ini. Hal ini untuk memastikan bahwa sistem berkas tidak melebihi kuota titik akses yang disebutkan.

Untuk informasi selengkapnya tentang titik akses EFS, lihat [Bekerja dengan titik akses](#).

Menggunakan konsol

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Pilih Access points untuk membuka jendela Access points.
3. Pilih Buat titik akses untuk menampilkan halaman Buat titik akses.

Anda juga dapat membuka halaman Create Access Point dengan memilih File Systems. Pilih nama sistem file atau ID sistem file dan kemudian pilih Titik akses dan Buat titik akses untuk membuat titik akses untuk sistem file itu.

- a. Masukkan informasi berikut di panel Detail:
- Sistem file — Masukkan nama atau ID sistem file dan pilih sistem file yang cocok. Anda juga dapat memilih sistem file dari daftar yang muncul ketika Anda memilih bidang input.
 - (Opsional) Nama — Masukkan nama untuk titik akses.
 - (Opsional) Jalur direktori Root - Anda dapat menentukan direktori root untuk titik akses; root titik akses default adalah /. Untuk memasukkan jalur direktori root, gunakan formatnya/foo/bar. Untuk informasi selengkapnya, lihat [Menegakkan direktori root dengan titik akses](#).
- b. (Opsional) Di panel pengguna POSIX, Anda dapat menentukan identitas POSIX lengkap yang akan digunakan untuk menegakkan informasi pengguna dan grup untuk semua operasi file oleh klien NFS yang menggunakan titik akses. Untuk informasi selengkapnya, lihat [Menegakkan identitas pengguna menggunakan titik akses](#).
- User ID — Masukkan ID pengguna POSIX numerik untuk pengguna.
 - ID Grup - Masukkan ID grup POSIX numerik untuk pengguna.
 - Kelompok sekunder IDs — Masukkan daftar grup sekunder yang dipisahkan koma opsional. IDs
- c. (Opsional) Untuk izin pembuatan direktori Root, Anda dapat menentukan izin yang akan digunakan saat Amazon EFS membuat jalur direktori root, jika ditentukan dan direktori root belum ada. Untuk informasi selengkapnya, lihat [Menegakkan direktori root dengan titik akses](#).

 Note

Jika Anda tidak menentukan kepemilikan dan izin direktori root apa pun, dan direktori root belum ada, EFS tidak akan membuat direktori root. Setiap upaya untuk me-mount sistem file dengan menggunakan titik akses akan gagal.

- ID pengguna pemilik - Masukkan ID pengguna POSIX numerik untuk digunakan sebagai pemilik direktori root.
- ID grup pemilik - Masukkan ID grup POSIX numerik untuk digunakan sebagai grup pemilik direktori root.

- Izin — Masuk ke mode Unix direktori. Konfigurasi umum adalah 755. Pastikan bit eksekusi diatur untuk pengguna titik akses sehingga mereka dapat me-mount.

4. Pilih Buat titik akses untuk membuat titik akses dengan menggunakan konfigurasi ini.

Menggunakan AWS CLI

Dalam contoh berikut, perintah `create-access-point` CLI menciptakan titik akses untuk sistem file EFS. Perintah API yang setara adalah [CreateAccessPoint](#).

```
aws efs create-access-point --file-system-id fs-abcdef0123456789a --client-token
010102020-3 \
--root-directory "Path=/efs/mobileapp/
east,CreationInfo={OwnerId=0,OwnerGid=11,Permissions=775}" \
--posix-user "Uid=22,Gid=4" \
--tags Key=Name,Value=east-users
```

Jika permintaan berhasil, CLI merespons dengan deskripsi titik akses.

```
{
  "ClientToken": "010102020-3",
  "Name": "east-users",
  "AccessPointId": "fsap-abcd1234ef5678901",
  "AccessPointArn": "arn:aws:elasticfilesystem:us-east-2:111122223333:access-point/
fsap-abcd1234ef5678901",
  "FileSystemId": "fs-01234567",
  "LifecycleState": "creating",
  "OwnerId": "111122223333",
  "PosixUser": {
    "Gid": 4,
    "Uid": 22
  },
  "RootDirectory": {
    "CreationInfo": {
      "OwnerGid": 0,
      "OwnerId": 11,
      "Permissions": "775"
    },
    "Path": "/efs/mobileapp/east",
  },
  "Tags": []
}
```

 Note

Jika beberapa permintaan untuk membuat titik akses pada sistem file yang sama dikirim secara berurutan, dan sistem file mendekati batas titik akses, Anda mungkin mengalami respons pelambatan untuk permintaan ini. Hal ini untuk memastikan bahwa sistem berkas tidak melebihi kuota titik akses yang disebutkan.

Menghapus titik akses

Saat Anda menghapus titik akses, setiap klien yang menggunakan jalur akses kehilangan akses ke sistem file Amazon EFS yang dikonfigurasi untuknya.

Menggunakan konsol

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Titik akses untuk membuka halaman Titik akses.
3. Pilih titik akses yang akan dihapus.
4. Pilih Hapus.
5. Pilih Konfirmasi untuk mengonfirmasi tindakan dan menghapus titik akses.

Menggunakan AWS CLI

Dalam contoh berikut, perintah `delete-access-point` CLI menghapus titik akses yang ditentukan. Perintah API yang setara adalah [DeleteAccessPoint](#). Jika perintah berhasil, layanan mengembalikan respons HTTP 204 dengan badan HTTP kosong.

```
aws efs delete-access-point --access-point-id fsap-092e9f80b3fb5e6f3 --client-token 010102020-3
```

Menandai sumber daya EFS

Untuk membantu mengelola sumber daya EFS, Anda dapat menetapkan metadata Anda sendiri ke setiap sumber daya dalam bentuk tag. Dengan tag, Anda dapat mengkategorikan AWS sumber daya Anda dengan cara yang berbeda, misalnya, berdasarkan tujuan, pemilik, atau lingkungan. Kategorisasi ini berguna ketika Anda memiliki banyak sumber daya dari jenis yang sama—

Anda dapat dengan cepat mengidentifikasi sumber daya tertentu berdasarkan tag yang telah Anda tetapkan padanya. Topik ini menjelaskan penandaan dan menunjukkan kepada Anda cara membuatnya.

Dasar-dasar tanda

Tag adalah label yang Anda tetapkan ke AWS sumber daya. Setiap tag terdiri atas sebuah kunci dan sebuah nilai opsional, dan Anda menentukan sendiri keduanya.

Tag memungkinkan Anda untuk mengkategorikan AWS sumber daya Anda dengan cara yang berbeda, misalnya, berdasarkan tujuan, pemilik, atau lingkungan. Misalnya, Anda dapat menentukan satu set tag untuk sistem file Amazon EFS akun Anda yang membantu Anda melacak setiap pemilik sistem file.

Sebaiknya rancang serangkaian kunci tag yang memenuhi kebutuhan setiap jenis sumber daya. Penggunaan set kunci tag yang konsisten akan memudahkan manajemen sumber daya Anda. Anda dapat mencari dan memfilter sumber daya berdasarkan tag yang Anda tambahkan.

Tag tidak memiliki arti semantik untuk Amazon EFS dan ditafsirkan secara ketat sebagai serangkaian karakter. Selain itu, tag tidak secara otomatis ditetapkan ke sumber daya Anda. Anda dapat mengedit kunci dan nilai tanda, dan dapat menghapus tanda dari sumber daya kapan saja. Anda dapat mengatur nilai tanda ke string kosong, tetapi tidak dapat mengatur nilai tanda ke null. Jika Anda menambahkan tanda yang memiliki kunci yang sama dengan tanda yang telah ada pada sumber daya tersebut, nilai yang baru akan menimpa nilai yang lama. Jika sumber daya dihapus, semua tanda untuk sumber daya tersebut juga akan dihapus.

Pembatasan tag

Batasan dasar berikut berlaku untuk tag:

- Jumlah maksimum tag per sumber daya – 50
- Untuk setiap sumber daya, setiap kunci tag harus unik, dan setiap kunci tag hanya dapat memuat satu nilai.
- Panjang kunci maksimum – 128 karakter Unicode dalam UTF-8
- Panjang nilai maksimum – 256 karakter Unicode di UTF-8
- Meskipun Amazon EFS memungkinkan karakter apa pun dalam tagnya, layanan lain lebih ketat. Karakter yang diperbolehkan pada layanan adalah huruf, angka, dan spasi yang dapat mewakili dalam UTF-8, serta karakter berikut: + - = . _ : / @.

- Kunci dan nilai tag peka terhadap huruf besar dan kecil.
- `aws :` Awalan dicadangkan untuk AWS digunakan. Jika tag memiliki kunci tag dengan awalan ini, Anda tidak dapat mengedit atau menghapus kunci atau nilai tag tersebut. Tag dengan awalan `aws :` tidak dihitung terhadap tag per batas sumber daya.

Anda tidak dapat memperbarui atau menghapus sumber daya hanya berdasarkan tagnya; Anda harus menentukan pengenalan sumber daya. Misalnya, untuk menghapus sistem file yang Anda tag dengan kunci tag yang disebut `DeleteMe`, Anda harus menggunakan `DeleteFileSystem` tindakan dengan pengidentifikasi sumber daya dari sistem file, seperti `fs-1234567890abcdef0`.

Saat Anda menandai sumber daya publik atau bersama, tag yang Anda tetapkan hanya tersedia untuk Anda Akun AWS. Tidak ada yang Akun AWS akan memiliki akses ke tag tersebut. Untuk kontrol akses berbasis tag ke sumber daya bersama, masing-masing Akun AWS harus menetapkan set tag sendiri untuk mengontrol akses ke sumber daya.

Anda dapat menandai sistem file Amazon EFS dan sumber daya titik akses.

Menggunakan tag untuk kontrol akses

Anda dapat menggunakan tag untuk mengontrol akses ke sumber daya Amazon EFS dan menerapkan kontrol akses berbasis atribut (ABAC).

Note

Replikasi tidak mendukung penggunaan tag untuk kontrol akses berbasis atribut (ABAC).

Memberikan tag pada sumber daya Anda

Anda dapat menandai sistem file Amazon EFS dan sumber daya titik akses yang sudah ada di akun Anda.

Menggunakan konsol

- Anda dapat menggunakan konsol Amazon EFS untuk menerapkan tag ke sumber daya yang ada dengan menggunakan tab Tag di layar detail sumber daya. Di konsol Amazon EFS, Anda dapat menentukan tag untuk sumber daya saat membuat sumber daya. Misalnya, Anda dapat menambahkan tag dengan kunci Name dan nilai yang Anda tentukan.

Dalam kebanyakan kasus, konsol menerapkan tag segera setelah sumber daya dibuat (bukan selama pembuatan sumber daya). Meskipun konsol mengatur sumber daya sesuai dengan Name tag, tag ini tidak memiliki arti semantik apa pun untuk layanan Amazon EFS.

Menggunakan AWS CLI

- Jika Anda menggunakan Amazon EFS API, the AWS CLI, atau AWS SDK, Anda dapat menggunakan tindakan TagResource EFS API untuk menerapkan tag ke sumber daya yang ada. Selain itu, beberapa tindakan pembuatan sumber daya memungkinkan Anda menentukan tag untuk sumber daya saat sumber daya tersebut dibuat.

AWS CLI Perintah untuk mengelola tag, dan tindakan Amazon EFS API yang setara, tercantum dalam tabel berikut.

Perintah CLI	Deskripsi	Operasi API yang setara
tag-resource	Tambahkan tag baru atau perbarui tag yang ada	TagResource
list-tags-for-resource	Ambil tag yang ada	ListTagsForResource
untag-resource	Hapus tag yang ada	UntagResource

Tutorial: Membuat subdirektori per pengguna yang dapat ditulis

Setelah Anda membuat sistem file EFS dan memasangnya secara lokal di instans Amazon EC2 (EC2) Anda, itu akan mengekspos direktori kosong yang disebut file. *file system root* Salah satu kasus penggunaan umum untuk direktori root sistem file ini adalah membuat subdirektori “dapat ditulis” untuk setiap pengguna yang Anda buat pada instance EC2 dan memasang subdirektori pada direktori home pengguna. Semua file dan subdirektori yang dibuat pengguna di direktori home mereka kemudian dibuat pada sistem file EFS.

Note

Anda dapat mengikuti [Memulai](#) latihan untuk membuat dan memasang sistem file EFS pada instans EC2 Anda.

Pada langkah-langkah berikut, Anda membuat pengguna, membuat subdirektori untuk pengguna, menjadikan pengguna sebagai pemilik subdirektori, lalu memasang subdirektori Amazon EFS di direktori home pengguna.

1. Buat mike pengguna:

- Masuk ke instans EC2 Anda. Menggunakan hak akses root (dalam hal ini, menggunakan sudo perintah), buat pengguna dan tetapkan kata sandi.

Misalnya, perintah berikut membuat penggunamike.

```
$ sudo useradd -c "Mike Smith" mike  
$ sudo passwd mike
```

Direktori home juga dibuat untuk pengguna. Misalnya, /home/mike.

2. Buat subdirektori di bawah *EFSroot* untuk pengguna.

Misalnya, perintah berikut membuat subdirektori mike di bawah *EFSroot*.

```
$ sudo mkdir /EFSroot/mike
```

Anda harus mengganti *EFSroot* dengan nama direktori lokal Anda.

3. Pengguna root dan grup root adalah pemilik subdirektori (Anda dapat memverifikasi ini dengan menggunakan `ls -l` perintah). Untuk mengaktifkan izin penuh bagi pengguna di subdirektori ini, berikan kepemilikan direktori kepada pengguna.

Contoh:

```
$ sudo chown mike:mike /EFSroot/mike
```

4. Gunakan mount perintah untuk me-mount subdirektori ke direktori home pengguna.

Contoh:

```
$ sudo mount -t nfs -o  
nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport mount-  
target-DNS:/mike /home/mike
```

mount-target-DNS Alamat mengidentifikasi root sistem file EFS jarak jauh.

Jika Anda melepas target pemasangan ini, pengguna tidak dapat mengakses direktori tanpa melakukan remounting, yang memerlukan izin root.

Memasang sistem file EFS

Untuk memasang sistem file EFS, sebaiknya gunakan EFS mount helper. EFS mount helper membantu Anda memasang sistem file EFS di instans EC2 Linux dan Mac yang menjalankan distribusi yang didukung. Mount helper adalah bagian dari kumpulan alat sumber terbuka yang diinstal saat Anda menginstal klien Amazon EFS (`amazon-efs-utils`). Untuk informasi selengkapnya tentang klien Amazon EFS dan distribusi yang didukung, lihat [Menginstal klien Amazon EFS](#).

Atau, Anda dapat memasang sistem file EFS secara manual menggunakan klien Linux NFS standar. Amazon EFS mendukung protokol Network File System versi 4.0 dan 4.1 (NFSv4) saat memasang sistem file Anda di instans Amazon EC2.

Selain itu, Anda dapat menggunakan EFS mount helper atau NFS untuk mengonfigurasi instans EC2 agar secara otomatis memasang sistem file EFS saat instance dimulai.

Topik

- [Pertimbangan pemasangan untuk Linux](#)
- [Memasang sistem file EFS menggunakan EFS mount helper](#)
- [Menggunakan Network File System untuk me-mount sistem file EFS](#)
- [Secara otomatis memasang sistem file EFS](#)
- [Melepaskan sistem file](#)
- [Tutorial: Buat sistem file EFS dan pasang pada instans EC2 menggunakan AWS CLI](#)
- [Tutorial: Pemasangan dengan klien Linux lokal](#)
- [Tutorial: Pasang sistem file dari VPC yang berbeda](#)
- [Memecahkan masalah pemasangan](#)

Pertimbangan pemasangan untuk Linux

Kami merekomendasikan nilai berikut untuk opsi pemasangan di Linux:

- `rsz=1048576`— Menetapkan jumlah maksimum byte data yang klien NFS dapat menerima untuk setiap permintaan READ jaringan. Nilai ini berlaku saat membaca data dari file pada sistem file EFS. Kami menyarankan Anda menggunakan ukuran sebesar mungkin (hingga `1048576`) untuk menghindari penurunan kinerja.

- `wsiz=1048576`— Menetapkan jumlah maksimum byte data yang klien NFS dapat mengirim untuk setiap permintaan WRITE jaringan. Nilai ini berlaku saat menulis data ke file pada sistem file EFS. Kami menyarankan Anda menggunakan ukuran sebesar mungkin (hingga 1048576) untuk menghindari penurunan kinerja.
- `hard`— Menetapkan perilaku pemulihan klien NFS setelah waktu permintaan NFS habis, sehingga permintaan NFS dicoba ulang tanpa batas hingga server membalas. Kami menyarankan Anda menggunakan opsi `hard mount` (`hard`) untuk memastikan integritas data. Jika Anda menggunakan `soft mount`, atur `timeo` parameter ke setidaknya 150 desidetik (15 detik). Melakukannya membantu meminimalkan risiko korupsi data yang melekat pada `soft mount`.
- `timeo=600`— Menetapkan nilai batas waktu yang digunakan klien NFS untuk menunggu respons sebelum mencoba ulang permintaan NFS ke 600 desidetik (60 detik). Jika Anda harus mengubah parameter batas waktu (`timeo`), kami sarankan Anda menggunakan nilai minimal 150, yang setara dengan 15 detik. Melakukannya membantu menghindari penurunan kinerja.
- `retrans=2`— Set ke 2 berapa kali klien NFS mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.
- `noresvport`— Memberitahu klien NFS untuk menggunakan port sumber Transmission Control Protocol (TCP) non-privileged baru ketika koneksi jaringan dibangun kembali. Melakukan hal ini membantu memastikan bahwa sistem file EFS memiliki ketersediaan tanpa gangguan setelah peristiwa pemulihan jaringan.
- `_netdev`— Saat hadir/`etc/fstab`, mencegah klien mencoba memasang sistem file EFS hingga jaringan diaktifkan.

Secara umum, hindari menyetel opsi pemasangan lain yang berbeda dari default, yang dapat menyebabkan penurunan kinerja dan masalah lainnya. Jika Anda tidak menggunakan default sebelumnya, perhatikan hal berikut:

- Mengubah ukuran buffer baca atau tulis atau menonaktifkan caching atribut dapat mengakibatkan penurunan kinerja.
- Amazon EFS mengabaikan port sumber. Jika Anda mengubah port sumber Amazon EFS, itu tidak berpengaruh apa pun.
- Amazon EFS tidak mendukung varian keamanan Kerberos mana pun. Misalnya, perintah `mount` berikut gagal.

```
$ mount -t nfs4 -o krb5p <DNS_NAME>:/ /efs/
```

- Kami menyarankan Anda me-mount sistem file Anda menggunakan nama DNS-nya. Amazon EFS menyelesaikan nama ini ke alamat IP target pemasangan EFS di Availability Zone yang sama dengan instans Amazon EC2 Anda tanpa memanggil sumber daya eksternal. Jika Anda menggunakan target pemasangan di Availability Zone yang berbeda dengan instans EC2, Anda dikenakan biaya EC2 standar untuk data yang dikirim di Availability Zone. Anda juga mungkin melihat peningkatan latensi untuk operasi sistem file.
- Untuk opsi pemasangan lainnya, dan penjelasan rinci tentang default, lihat dokumentasi Linux.

Note

Jika instans EC2 Anda perlu memulai terlepas dari status sistem file EFS yang dipasang, tambahkan `nofail` opsi ke entri sistem file Anda di `/etc/fstab` file Anda.

Memasang sistem file EFS menggunakan EFS mount helper

Setelah menginstal Amazon EFS client (`amazon-efs-utils`), Anda dapat menggunakan EFS mount helper untuk memasang sistem file EFS di instans EC2 Linux dan Mac yang menjalankan distribusi yang didukung. Amazon EFS tidak mendukung pemasangan dari instans Windows EC2.

Important

Kami menyarankan Anda selalu menggunakan versi terbaru `amazon-efs-utils` untuk memastikan pemasangan yang berhasil. Misalnya, versi `amazon-efs-utils` sebelum 2.3 tidak mendukung pemasangan dengan IPv6 alamat.

Saat memasang sistem file, mount helper mendefinisikan jenis sistem file jaringan baru, yang disebut `efs`, yang sepenuhnya kompatibel dengan mount perintah standar di Linux. Mount helper juga mendukung pemasangan sistem file EFS pada waktu boot instance secara otomatis dengan menggunakan entri dalam file `/etc/fstab` konfigurasi pada instans EC2 Linux.

Warning

Gunakan opsi `_netdev`, yang digunakan untuk mengidentifikasi sistem file jaringan, ketika memasang sistem file Anda secara otomatis. Jika `_netdev` hilang, instans EC2 Anda mungkin berhenti merespons. Hasil ini didapatkan karena sistem file jaringan perlu

diinisialisasi setelah instans komputasi memulai jaringannya. Untuk informasi selengkapnya, lihat [Pemasangan otomatis gagal dan instans tidak responsif](#).

Anda dapat me-mount sistem file dengan menentukan salah satu properti berikut:

- Nama DNS sistem file — Jika Anda menggunakan nama DNS sistem file, dan mount helper tidak dapat menyelesaikannya, misalnya ketika Anda memasang sistem file di VPC yang berbeda, itu akan kembali menggunakan alamat IP target mount. Untuk informasi selengkapnya, lihat [Memasang sistem file EFS dari yang lain Akun AWS atau VPC](#).
- ID sistem file - Jika Anda menggunakan ID sistem file, mount helper menyelesaikannya ke alamat IP lokal dari mount target elastic network interface (ENI) tanpa memanggil sumber daya eksternal.
- Pasang alamat IP target — Anda dapat menggunakan alamat IP dari salah satu target pemasangan sistem file.

Anda dapat menemukan nilai untuk semua properti ini di konsol Amazon EFS. Nama DNS sistem file ditemukan di layar Lampirkan.

Ketika enkripsi data dalam perjalanan dideklarasikan sebagai opsi pemasangan untuk sistem file EFS Anda, mount helper menginisialisasi `stunnel` proses klien, dan proses supervisor dipanggil. `amazon-efs-mount-watchdog` proses ini memantau kesehatan pemasangan TLS, dan dimulai secara otomatis saat pertama kali sistem file EFS dipasang melalui TLS. Jika klien Anda berjalan di Linux, proses ini dikelola oleh salah satu `upstart` atau `systemd` tergantung pada distribusi Linux Anda. Untuk klien yang berjalan pada macOS yang didukung, ini dikelola oleh `launchd`.

`Stunnel` adalah relay jaringan multiguna sumber terbuka. `stunnel` proses klien mendengarkan pada port lokal untuk lalu lintas masuk, dan mount helper mengarahkan lalu lintas klien NFS ke port lokal ini.

Mount helper menggunakan TLS versi 1.2 untuk berkomunikasi dengan sistem file Anda. Menggunakan TLS memerlukan sertifikat, dan sertifikat ini ditandatangani oleh Otoritas Sertifikat Amazon terpercaya. Untuk informasi selengkapnya tentang cara kerja enkripsi, lihat [Enkripsi data di Amazon EFS](#).

Topik

- [Pengaturan pemasangan yang digunakan oleh EFS mount helper](#)

- [Mendapatkan log dukungan](#)
- [Prasyarat untuk menggunakan EFS mount helper](#)
- [Pemasangan pada instans EC2 Linux menggunakan EFS mount helper](#)
- [Pemasangan pada instans EC2 Mac menggunakan EFS mount helper](#)
- [Memasang sistem file EFS dari yang berbeda AWS Region](#)
- [Memasang sistem file One Zone](#)
- [Pemasangan dengan otorisasi IAM](#)
- [Pemasangan dengan titik akses EFS](#)
- [Memasang EFS ke beberapa instans EC2](#)
- [Memasang sistem file EFS dari yang lain Akun AWS atau VPC](#)

Pengaturan pemasangan yang digunakan oleh EFS mount helper

Klien mount helper Amazon EFS menggunakan opsi pemasangan berikut yang dioptimalkan untuk Amazon EFS:

- `nfsvers=4.1`— digunakan saat memasang pada instans EC2 Linux
- `nfsvers=4.0`— digunakan saat memasang pada instans EC2 Mac yang didukung yang menjalankan macOS Big Sur, Monterey, dan Ventura
- `rsiz=1048576`— Menetapkan jumlah maksimum byte data yang klien NFS dapat menerima untuk setiap permintaan READ jaringan ke 1048576, terbesar yang tersedia, untuk menghindari kinerja berkurang.
- `wsiz=1048576`— Menetapkan jumlah maksimum byte data yang dapat dikirim klien NFS untuk setiap permintaan WRITE jaringan 1048576, yang terbesar yang tersedia, untuk menghindari penurunan kinerja.
- `hard`— Menetapkan perilaku pemulihan klien NFS setelah waktu permintaan NFS habis, sehingga permintaan NFS dicoba ulang tanpa batas hingga server membalas, untuk memastikan integritas data.
- `timeo=600`— Menetapkan nilai batas waktu yang digunakan klien NFS untuk menunggu respons sebelum mencoba ulang permintaan NFS ke 600 desidetik (60 detik) untuk menghindari penurunan kinerja.
- `retrans=2`— Set ke 2 berapa kali klien NFS mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.

- `noresvport` Memberitahu klien NFS untuk menggunakan port sumber Transmission Control Protocol (TCP) non-privileged baru ketika koneksi jaringan dibangun kembali. Menggunakan `noresvport` opsi ini membantu memastikan bahwa sistem file EFS Anda memiliki ketersediaan tanpa gangguan setelah peristiwa penyambungan ulang atau pemulihan jaringan.
- `mountport=2049`— hanya digunakan saat memasang pada instans EC2 Mac yang menjalankan macOS Big Sur, Monterey, dan Ventura.

Mendapatkan log dukungan

EFS mount helper memiliki pencatatan bawaan untuk sistem file EFS Anda. Anda dapat membagikan log ini dengan AWS Support untuk tujuan pemecahan masalah. Anda dapat menemukan log yang disimpan `/var/log/amazon/efs` di klien menggunakan EFS mount helper. Log ini untuk EFS mount helper, proses stunnel (dinonaktifkan secara default), dan untuk `amazon-efs-mount-watchdog` proses yang memantau proses stunnel.

Note

`amazon-efs-mount-watchdog` Proses ini memastikan bahwa setiap proses stunnel mount berjalan, dan menghentikan proses stunnel saat sistem file EFS dilepas. Jika karena alasan tertentu proses stunnel dihentikan secara tak terduga, proses pengawas akan memulai kembali.

Anda dapat mengubah konfigurasi log di `/etc/amazon/efs/efs-utils.conf`. Agar perubahan log dapat diterapkan, Anda perlu melepas dan memasang kembali sistem file menggunakan EFS mount helper. Kapasitas log untuk mount helper dan watchdog log dibatasi hingga 20 MiB. Log untuk proses stunnel dinonaktifkan secara default.

Important

Anda dapat mengaktifkan logging untuk log proses stunnel. Namun, mengaktifkan log stunnel dapat menggunakan jumlah ruang yang tidak sepele pada sistem file Anda.

Prasyarat untuk menggunakan EFS mount helper

Anda dapat memasang sistem file EFS di instans Amazon EC2 menggunakan helper mount Amazon EFS. Untuk menggunakan mount helper, Anda memerlukan yang berikut:

- ID sistem file dari sistem file yang akan dipasang - Helper mount EFS menyelesaikan ID sistem file ke alamat IP lokal dari mount target elastic network interface (ENI) tanpa memanggil sumber daya eksternal.
- Target pemasangan EFS — Anda membuat target mount di cloud pribadi virtual (VPC) Anda. Jika Anda membuat sistem file di konsol menggunakan pengaturan yang direkomendasikan layanan, target pemasangan dibuat di setiap Availability Zone di AWS Region mana sistem file berada. Untuk petunjuk membuat target pemasangan, lihat [Mengelola target mount](#).

Note

Kami menyarankan Anda menunggu 60 detik setelah status siklus hidup target mount yang baru dibuat tersedia sebelum memasang sistem file melalui DNS. Penantian ini memungkinkan catatan DNS menyebar sepenuhnya di AWS Region tempat sistem file berada.

Jika Anda menggunakan target pemasangan di Availability Zone yang berbeda dengan instans EC2, Anda dikenakan biaya EC2 standar untuk data yang dikirim di Availability Zone. Anda juga mungkin melihat peningkatan latensi untuk operasi sistem file.

- Untuk memasang sistem file One Zone dari Availability Zone yang berbeda:
 - Nama Availability Zone sistem file — Jika Anda memasang sistem file EFS One Zone yang terletak di Availability Zone yang berbeda dari instans EC2.
 - Pasang nama DNS target — Atau, Anda dapat menentukan nama DNS target mount, bukan Availability Zone.
- Instans EC2 yang menjalankan salah satu distribusi Linux atau macOS yang didukung - Distribusi yang didukung untuk memasang sistem file Anda dengan mount helper adalah sebagai berikut:
 - Amazon Linux 2
 - Amazon Linux 2023
 - Amazon Linux 2017.09 dan yang lebih baru
 - macOS Big Sur

- Red Hat Enterprise Linux (dan turunannya seperti CentOS) versi 7 dan yang lebih baru
- Ubuntu 16.04 LTS dan yang lebih baru

 Note

Instans EC2 Mac yang menjalankan macOS Big Sur hanya mendukung NFS 4.0.

- EFS mount helper diinstal pada instans EC2 — Mount helper adalah alat dalam `amazon-efs-utils` paket utilitas. Untuk informasi tentang menginstal `amazon-efs-utils`, lihat [Menginstal klien Amazon EFS](#).
- Instans EC2 ada dalam VPC — Instans EC2 yang menghubungkan harus dalam virtual private cloud (VPC) berdasarkan layanan Amazon VPC. Itu juga harus dikonfigurasi untuk menggunakan server DNS yang disediakan oleh AWS. Untuk informasi tentang server DNS Amazon, lihat [set opsi DHCP di Amazon VPC di Panduan Pengguna Amazon VPC](#).
- VPC mengaktifkan nama host DNS — VPC dari instans EC2 yang menghubungkan harus mengaktifkan nama host DNS. Untuk informasi selengkapnya, lihat [atribut DNS untuk VPC Anda](#) di Panduan Pengguna Amazon VPC.
- Untuk instans EC2 dan sistem file yang berbeda Wilayah AWS — Jika instans EC2 dan sistem file yang Anda pasang terletak berbeda Wilayah AWS, Anda perlu mengedit `region` properti dalam `file.efs-utils.conf`. Untuk informasi selengkapnya, lihat [Memasang sistem file EFS dari yang berbeda AWS Region](#).

Pemasangan pada instans EC2 Linux menggunakan EFS mount helper

Prosedur ini membutuhkan yang berikut:

- Anda telah menginstal `amazon-efs-utils` paket pada instans Amazon EC2. Untuk informasi selengkapnya, lihat [Menginstal klien Amazon EFS secara manual](#).
- Anda telah membuat target mount untuk sistem file. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).

Untuk me-mount sistem file EFS Anda menggunakan mount helper pada instans EC2 Linux

1. Buka jendela terminal pada instans EC2 Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.

2. Buat direktori `efs` yang akan Anda gunakan sebagai titik pemasangan sistem file menggunakan perintah berikut:

```
sudo mkdir efs
```

3. Jalankan salah satu perintah berikut untuk me-mount sistem file Anda.

Note

Jika instans EC2 dan sistem file yang Anda pasang terletak di AWS Region s yang berbeda, lihat [Memasang sistem file EFS dari yang berbeda AWS Region](#) untuk mengedit region properti dalam `efs-utils.conf` file.

- Untuk me-mount menggunakan id sistem file:

```
sudo mount -t efs file-system-id efs-mount-point/
```

Gunakan ID sistem file yang Anda pasang di tempat *file-system-id* dan `efs` di tempat *efs-mount-point*.

```
sudo mount -t efs fs-abcd123456789ef0 efs/
```

Atau, jika Anda ingin menggunakan enkripsi data dalam perjalanan, Anda dapat me-mount sistem file Anda dengan perintah berikut.

```
sudo mount -t efs -o tls fs-abcd123456789ef0:/ efs/
```

- Untuk me-mount menggunakan nama DNS sistem file:

```
sudo mount -t efs -o tls file-system-dns-name efs-mount-point/
```

```
sudo mount -t efs -o tls fs-abcd123456789ef0.efs.us-east-2.amazonaws.com efs/
```

- Untuk me-mount menggunakan alamat IP target mount:

```
sudo mount -t efs -o tls,mounttargetip=mount-target-ip file-system-id efs-mount-point/
```

```
sudo mount -t efs -o tls,mounttargetip=192.0.2.0 fs-abcd123456789ef0 efs/
```

Anda dapat melihat dan menyalin perintah yang tepat untuk me-mount sistem file Anda di kotak dialog Lampirkan.

- a. Di konsol Amazon EFS, pilih sistem file yang ingin Anda pasang untuk menampilkan halaman detailnya.
- b. Untuk menampilkan perintah mount yang akan digunakan untuk sistem file ini, pilih Lampirkan di kanan atas.

Layar Lampirkan menampilkan perintah yang tepat untuk digunakan untuk memasang sistem file dengan cara berikut:

- (Pasang melalui DNS) Menggunakan nama DNS sistem file dengan EFS mount helper atau klien NFS.
- (Pasang melalui IP) Menggunakan alamat IP target mount di Availability Zone yang dipilih dengan klien NFS.

Pemasangan pada instans EC2 Mac menggunakan EFS mount helper

Prosedur ini membutuhkan yang berikut:

- Anda telah menginstal `amazon-efs-utils` paket pada instans Amazon EC2 Mac. Untuk informasi selengkapnya, lihat [Menginstal klien Amazon EFS di instans EC2 Mac yang menjalankan macOS Big Sur, macOS Monterey, atau macOS Ventura](#).
- Anda telah membuat target mount untuk sistem file. Anda dapat membuat target mount pada pembuatan sistem file dan menambahkannya ke sistem file yang ada. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).
- Anda memasang sistem file pada instans EC2 Mac yang menjalankan macOS Big Sur, Monterey, atau Ventura. Versi macOS lainnya tidak didukung.

 Note

Hanya instans EC2 Mac yang menjalankan macOS Big Sur, Monterey, dan Ventura yang didukung. Versi macOS lainnya tidak didukung untuk digunakan dengan Amazon EFS.

Untuk memasang sistem file EFS Anda menggunakan EFS mount helper pada instans EC2 Mac yang menjalankan macOS Big Sur, Monterey, atau Ventura

1. Buka jendela terminal pada instans EC2 Mac Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Buat direktori untuk digunakan sebagai titik pemasangan sistem file menggunakan perintah berikut:

```
sudo mkdir efs
```

3. Jalankan perintah berikut untuk me-mount sistem file Anda.

 Note

Secara default, EFS mount helper menggunakan enkripsi saat transit saat memasang pada instans EC2 Mac, baik Anda menggunakan `tls` opsi dalam perintah mount atau tidak.

```
sudo mount -t efs file-system-id efs-mount-point/
```

```
sudo mount -t efs fs-abcd123456789ef0 efs/
```

Anda juga dapat menggunakan `tls` opsi saat memasang.

```
sudo mount -t efs -o tls fs-abcd123456789ef0:/ efs
```

Untuk memasang sistem file pada instance EC2 Mac tanpa menggunakan enkripsi saat transit, gunakan `notls` opsi, seperti yang ditunjukkan pada perintah berikut.

```
sudo mount -t efs -o noatime file-system-id efs-mount-point/
```

Anda dapat melihat dan menyalin perintah yang tepat untuk memasang sistem file Anda di kotak dialog Lampirkan konsol manajemen, yang dijelaskan sebagai berikut.

- a. Di konsol Amazon EFS, pilih sistem file yang ingin Anda pasang untuk menampilkan halaman detailnya.
- b. Untuk menampilkan perintah mount yang akan digunakan untuk sistem file ini, pilih Lampirkan di kanan atas.

Layar Lampirkan menampilkan perintah yang tepat untuk digunakan untuk memasang sistem file dengan cara berikut:

- (Pasang melalui DNS) Menggunakan nama DNS sistem file dengan EFS mount helper atau klien NFS.
- (Pasang melalui IP) Menggunakan alamat IP target mount di Availability Zone yang dipilih dengan klien NFS.

Memasang sistem file EFS dari yang berbeda AWS Region

Untuk me-mount sistem file EFS Anda dari instans EC2 yang berbeda AWS Region dari sistem file, Anda harus mengedit nilai region properti dalam `efs-utils.conf` file.

Untuk mengedit **region** properti di **efs-utils.conf**

1. Akses terminal untuk instans EC2 Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Temukan `/etc/amazon/efs/efs-utils.conf` file, dan buka menggunakan editor pilihan Anda.
3. Temukan baris berikut:

```
#region = us-east-1
```

- a. Hapus komentar baris.

- b. Jika sistem file tidak terletak di us-east-1 Wilayah, ganti us-east-1 dengan ID Wilayah di mana sistem file berada.
 - c. Simpan perubahan.
4. Tambahkan entri host untuk pemasangan lintas wilayah. Untuk informasi lebih lanjut tentang cara melakukan ini, lihat [Langkah 3: Tambahkan entri host untuk target pemasangan](#).
5. Pasang sistem file menggunakan EFS mount helper untuk instance [Linux](#) atau [Mac](#).

Memasang sistem file One Zone

Sistem file EFS One Zone hanya mendukung satu target mount yang terletak di Availability Zone yang sama dengan sistem file. Anda tidak dapat menambahkan target pemasangan tambahan. Bagian ini menjelaskan hal-hal yang perlu dipertimbangkan saat memasang sistem file One Zone.

Anda dapat menghindari biaya transfer data antar Availability Zone dan mencapai kinerja yang lebih baik dengan mengakses sistem file EFS menggunakan instans komputasi Amazon EC2 yang terletak di Availability Zone yang sama dengan target pemasangan sistem file.

Prosedur di bagian ini membutuhkan yang berikut:

- Anda telah menginstal `amazon-efs-utils` package pada instans EC2. Untuk informasi selengkapnya, lihat [Menginstal klien Amazon EFS](#).
- Anda telah membuat target mount untuk sistem file. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).

Memasang sistem file One Zone pada EC2 di Availability Zone yang berbeda

Jika Anda memasang sistem file One Zone pada instans Amazon EC2 yang terletak di Availability Zone yang berbeda, Anda harus menentukan nama Availability Zone sistem file atau nama DNS dari target pemasangan sistem file dalam perintah mount helper mount.

Buat direktori yang dipanggil `efs` untuk digunakan sebagai titik pemasangan sistem file menggunakan perintah berikut:

```
sudo mkdir efs
```

Gunakan perintah berikut untuk me-mount sistem file menggunakan EFS mount helper. Perintah menentukan nama Availability Zone sistem file.

```
sudo mount -t efs -o az=availability-zone-name,tls file-system-id mount-point/
```

Ini adalah perintah dengan nilai sampel:

```
sudo mount -t efs -o az=us-east-1a,tls fs-abcd1234567890ef efs/
```

Perintah berikut memasang sistem file, menentukan nama DNS dari target mount sistem file.

```
sudo mount -t efs -o tls mount-target-dns-name mount-point/
```

Ini adalah perintah dengan contoh mount nama DNS target.

```
sudo mount -t efs -o tls us-east-1a.fs-abcd1234567890ef9.efs.us-east-1.amazonaws.com  
efs/
```

Memasang sistem file One Zone di Availability Zone yang berbeda secara otomatis dengan EFS mount helper

Jika Anda menggunakan `/etc/fstab` untuk memasang sistem file EFS One Zone pada instans EC2 yang terletak di Availability Zone yang berbeda, Anda harus menentukan nama Availability Zone sistem file atau nama DNS dari target pemasangan sistem file di entri. `/etc/fstab`

```
availability-zone-name.file-system-id.efs.aws-region.amazonaws.com:/ efs-mount-point  
efs defaults,_netdev,noresvport,tls 0 0
```

```
us-east-1a.fs-abc123def456a7890.efs.us-east-1.amazonaws.com:/ efs-one-zone efs  
defaults,_netdev,noresvport,tls 0 0
```

Memasang sistem file One Zone secara otomatis dengan NFS

Jika Anda menggunakan `/etc/fstab` untuk memasang sistem file EFS menggunakan penyimpanan One Zone pada instans EC2 yang terletak di Availability Zone yang berbeda, Anda harus menentukan nama Availability Zone sistem file dengan nama DNS sistem file di entri. `/etc/fstab`

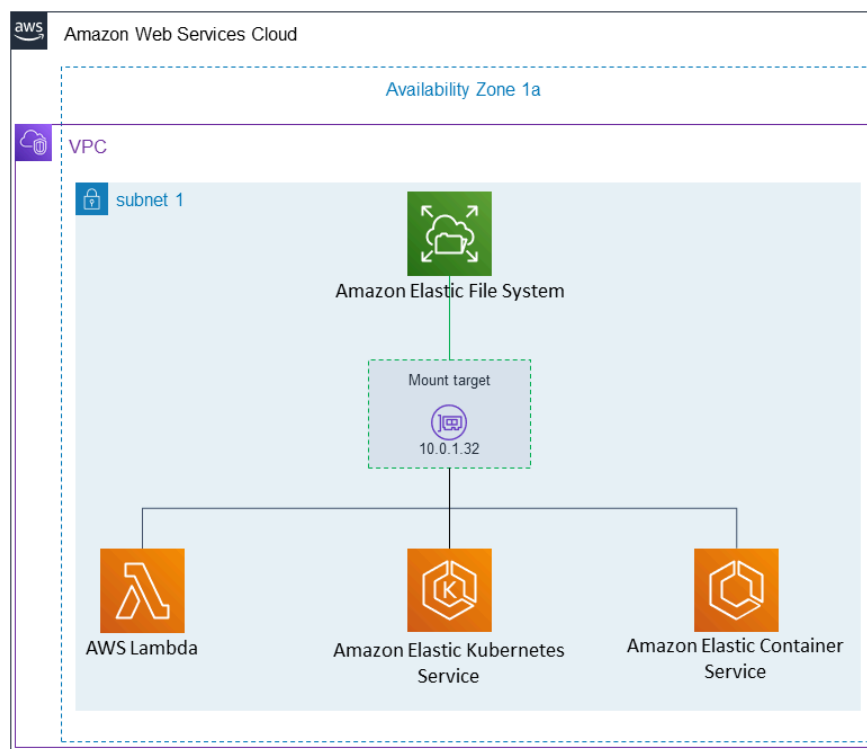
```
availability-zone-name.file-system-id.efs.aws-region.amazonaws.com:/ efs-mount-point  
nfs4  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0  
0
```

```
us-east-1a.fs-abc123def456a7890.efs.us-east-1.amazonaws.com:/ efs-one-zone nfs4
nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
0
```

Untuk informasi selengkapnya tentang cara mengedit `/etc/fstab` file, dan nilai yang digunakan dalam perintah ini, lihat [Secara otomatis memasang sistem file EFS](#).

Memasang sistem file dengan sistem file One Zone pada instance AWS komputasi lainnya

Bila Anda menggunakan sistem file One Zone dengan Amazon Elastic Container Service, Amazon Elastic Kubernetes Service, AWS Lambda atau, Anda perlu mengonfigurasi layanan untuk menggunakan Availability Zone yang sama dengan sistem file EFS, diilustrasikan sebagai berikut, dan dijelaskan di bagian berikut.



Menghubungkan dari Amazon Elastic Container Service

Anda dapat menggunakan sistem file EFS dengan Amazon ECS untuk berbagi data sistem file di seluruh armada instans kontainer sehingga tugas Anda memiliki akses ke penyimpanan persisten yang sama, apa pun instans tempat mereka mendarat. Untuk menggunakan sistem file EFS One

Zone dengan Amazon ECS, Anda harus memilih hanya subnet yang berada di Availability Zone yang sama dengan sistem file Anda saat meluncurkan tugas Anda. Untuk informasi selengkapnya, lihat [volume Amazon EFS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

Menghubungkan dari Amazon Elastic Kubernetes Service

Saat memasang sistem file One Zone dari Amazon EKS, Anda dapat menggunakan driver Amazon EFS [Container Storage Interface](#) (CSI), yang mendukung titik akses EFS, untuk berbagi sistem file antara beberapa pod di Amazon EKS atau cluster Kubernetes yang dikelola sendiri. Driver Amazon EFS CSI diinstal di tumpukan Fargate. Saat menggunakan driver Amazon EFS CSI dengan sistem file EFS One Zone, Anda dapat menggunakan `nodeSelector` opsi ini saat meluncurkan pod untuk memastikan dijadwalkan dalam Availability Zone yang sama dengan sistem file Anda.

Menghubungkan dari AWS Lambda

Anda dapat menggunakan Amazon EFS AWS Lambda untuk berbagi data di seluruh pemanggilan fungsi, membaca file data referensi besar, dan menulis output fungsi ke penyimpanan persisten dan bersama. Lambda menghubungkan instance fungsi dengan aman ke target pemasangan EFS yang berada di Availability Zone dan subnet yang sama. Saat Anda menggunakan Lambda dengan sistem file One Zone, konfigurasi fungsi Anda untuk hanya meluncurkan pemanggilan ke subnet yang berada di Availability Zone yang sama dengan sistem file Anda.

Pemasangan dengan otorisasi IAM

Untuk memasang sistem file EFS Anda di instans Linux menggunakan otorisasi AWS Identity and Access Management (IAM), gunakan EFS mount helper. Untuk informasi selengkapnya tentang otorisasi IAM untuk klien NFS, lihat. [Menggunakan IAM untuk mengontrol akses ke sistem file](#)

Anda perlu membuat direktori untuk digunakan sebagai titik pemasangan sistem file di bagian berikut. Anda dapat menggunakan perintah berikut untuk membuat direktori `mount pointefs`:

```
sudo mkdir efs
```

Anda kemudian dapat mengganti instance *efs-mount-point* dengan `efs`.

Pemasangan dengan IAM menggunakan profil instans EC2

Jika Anda memasang dengan otorisasi IAM ke instans Amazon EC2 dengan profil instans, gunakan opsi `iam` dan pasang, `tls` yang ditunjukkan berikut.

```
$ sudo mount -t efs -o tls,iam file-system-id efs-mount-point/
```

Untuk secara otomatis me-mount dengan otorisasi IAM ke instans EC2 yang memiliki profil instans, tambahkan baris berikut ke `/etc/fstab` file pada instans EC2.

```
file-system-id:/ efs-mount-point efs _netdev,tls,iam 0 0
```

Pemasangan dengan IAM menggunakan profil bernama

Anda dapat me-mount dengan otorisasi IAM menggunakan kredensial IAM yang terletak di file AWS CLI kredensial, atau file konfigurasi. `~/.aws/credentials` AWS CLI `~/.aws/config` Jika tidak "awsprofile" ditentukan, profil "default" digunakan.

Untuk me-mount dengan otorisasi IAM ke instance Linux menggunakan file kredensial, gunakan opsi `awsprofile`, dan `iam mount tlsawsprofile`, yang ditunjukkan berikut.

```
$ sudo mount -t efs -o tls,iam,awsprofile=namedprofile file-system-id efs-mount-point/
```

Untuk secara otomatis me-mount dengan otorisasi IAM ke instance Linux menggunakan file kredensial, tambahkan baris berikut ke `/etc/fstab` file pada instance EC2.

```
file-system-id:/ efs-mount-point efs _netdev,tls,iam,awsprofile=namedprofile 0 0
```

Pemasangan dengan titik akses EFS

Anda dapat memasang sistem file EFS menggunakan titik akses EFS hanya dengan menggunakan EFS mount helper.

Note

Anda harus mengonfigurasi satu atau beberapa target pemasangan untuk sistem file Anda saat memasang sistem file menggunakan titik akses EFS.

Saat Anda memasang sistem file menggunakan titik akses, perintah mount menyertakan opsi `access-point-id` dan `tls mount` selain opsi pemasangan biasa. Contoh ditunjukkan berikut.

```
$ sudo mount -t efs -o tls,accesspoint=access-point-id file-system-id efs-mount-point
```

Untuk secara otomatis memasang sistem file menggunakan titik akses, tambahkan baris berikut ke `/etc/fstab` file pada instance EC2.

```
file-system-id efs-mount-point efs _netdev,tls,accesspoint=access-point-id 0 0
```

Untuk informasi selengkapnya tentang titik akses EFS, lihat [Bekerja dengan titik akses](#).

Memasang EFS ke beberapa instans EC2

Anda dapat memasang sistem file EFS ke beberapa instans Amazon EC2 dari jarak jauh dan aman tanpa harus masuk ke instans menggunakan Perintah Jalankan. AWS Systems Manager Untuk informasi selengkapnya tentang AWS Systems Manager Jalankan Perintah, lihat [AWS Systems Manager Menjalankan Perintah](#) di Panduan AWS Systems Manager Pengguna. Prasyarat berikut diperlukan sebelum memasang sistem file EFS menggunakan metode ini:

1. Instans EC2 diluncurkan dengan profil instans yang menyertakan kebijakan `AmazonElasticFileSystemsUtils` izin. Untuk informasi selengkapnya, lihat [Langkah 1: Konfigurasi profil instans IAM dengan izin yang diperlukan](#).
2. Versi 1.28.1 atau yang lebih baru dari klien Amazon EFS (`amazon-efs-utils` paket) diinstal pada instans EC2. Anda dapat menggunakan AWS Systems Manager untuk menginstal paket secara otomatis pada instans Anda. Untuk informasi selengkapnya, lihat [Langkah 2: Konfigurasi asosiasi yang digunakan oleh State Manager](#).

Untuk memasang beberapa sistem file EFS ke beberapa instans EC2 menggunakan konsol

1. Buka AWS Systems Manager konsol di <https://console.aws.amazon.com/systems-manager/>.
2. Di panel navigasi, pilih Jalankan Perintah.
3. Pilih Run a command (Jalankan perintah).
4. Masukkan **AWS-RunShellScript** di bidang pencarian Perintah.
5. Pilih AWS- RunShellScript.
6. Dalam parameter Command masukkan perintah mount yang akan digunakan untuk setiap sistem file EFS yang ingin Anda pasang. Contoh:

```
sudo mount -t efs -o tls fs-12345678:/ /mnt/efs  
sudo mount -t efs -o tls,accesspoint=fsap-12345678 fs-01233210 /mnt/efs
```

Untuk informasi selengkapnya tentang perintah pemasangan EFS menggunakan klien Amazon EFS, lihat [Pemasangan pada instans EC2 Linux menggunakan EFS mount helper](#) atau [Pemasangan pada instans EC2 Mac menggunakan EFS mount helper](#).

7. Pilih instans EC2 AWS Systems Manager terkelola target yang Anda inginkan untuk menjalankan perintah.
8. Buat pengaturan tambahan lainnya yang Anda inginkan. Kemudian pilih Jalankan untuk menjalankan perintah dan pasang sistem file EFS yang ditentukan dalam perintah.

Setelah Anda menjalankan perintah, Anda dapat melihat statusnya di riwayat perintah.

Memasang sistem file EFS dari yang lain Akun AWS atau VPC

Anda dapat memasang sistem file EFS Anda menggunakan otorisasi IAM untuk klien NFS dan titik akses EFS menggunakan EFS mount helper. Secara default, EFS mount helper menggunakan layanan nama domain (DNS) untuk menyelesaikan alamat IP target pemasangan EFS Anda. Jika Anda memasang sistem file dari akun lain atau virtual private cloud (VPC), Anda harus menyelesaikan target pemasangan EFS secara manual.

Setelah itu, Anda dapat menemukan petunjuk untuk menentukan alamat IP target pemasangan EFS yang benar untuk digunakan untuk klien NFS Anda. Anda juga dapat menemukan instruksi untuk mengonfigurasi klien untuk memasang sistem file EFS menggunakan alamat IP tersebut.

Topik

- [Memasang sistem file EFS dari yang lain Akun AWS](#)
- [Memasang sistem file EFS dari VPC lain](#)

Memasang sistem file EFS dari yang lain Akun AWS

Menggunakan shared VPCs, Anda dapat memasang sistem file EFS yang dimiliki oleh salah satu Akun AWS dari instans Amazon EC2 yang dimiliki oleh instans lain. Akun AWS Untuk informasi selengkapnya tentang menyiapkan VPC bersama, lihat [Membagikan VPC Anda dengan akun lain di Panduan Peering VPC Amazon](#).

Setelah Anda menyiapkan berbagi VPC, instans EC2 dapat memasang sistem file EFS menggunakan resolusi nama Domain Name System (DNS) atau helper mount EFS. Sebaiknya gunakan EFS mount helper untuk memasang sistem file EFS Anda.

Memasang sistem file EFS dari VPC lain

Saat Anda menggunakan koneksi peering VPC atau gateway transit untuk menghubungkan VPC, instans Amazon EC2 yang ada di satu VPC dapat mengakses sistem file EFS di VPC lain, meskipun milik akun yang berbeda. VPCs

Anda tidak dapat menggunakan resolusi nama DNS untuk titik pemasangan EFS di VPC lain. Untuk memasang sistem file EFS Anda, gunakan alamat IP titik pemasangan di Availability Zone yang sesuai.

Atau, Anda dapat menggunakan Amazon Route 53 sebagai layanan DNS Anda. Di Route 53, Anda dapat menyelesaikan alamat IP target pemasangan EFS dari VPC lain dengan membuat zona host pribadi dan kumpulan catatan sumber daya. Untuk informasi selengkapnya tentang cara melakukannya, lihat [Bekerja dengan zona yang dihosting pribadi](#) di Panduan Pengembang Amazon Route 53.

Prasyarat

Sebelum menggunakan prosedur berikut ini, lakukan langkah-langkah ini:

- Instal klien Amazon EFS, bagian dari `amazon-efs-utils` rangkaian utilitas pada instance komputasi tempat Anda memasang sistem file EFS. Anda menggunakan EFS mount helper, yang disertakan dalam `amazon-efs-utils`, untuk me-mount sistem file. Untuk petunjuk tentang menginstal `amazon-efs-utils`, lihat [Menginstal klien Amazon EFS](#).
- Izinkan `ec2:DescribeAvailabilityZones` tindakan dalam kebijakan IAM untuk peran IAM yang Anda lampirkan ke instance. Kami menyarankan Anda melampirkan kebijakan AWS terkelola `AmazonElasticFileSystemsUtils` ke entitas IAM untuk memberikan izin yang diperlukan untuk entitas.
- Saat memasang dari yang lain Akun AWS, perbarui kebijakan sumber daya sistem file untuk memungkinkan `elasticfilesystem:DescribeMountTarget` tindakan untuk ARN utama lainnya. Akun AWS Contoh:

```
{
  "Id": "access-point-example03",
  "Statement": [
    {
      "Sid": "access-point-statement-example03",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::555555555555:root"},

```

```
    "Action": "elasticfilesystem:DescribeMountTargets",
    "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-
system/fs-12345678"
  }
]
```

Untuk informasi selengkapnya tentang kebijakan sumber daya sistem file EFS, lihat [Kebijakan berbasis sumber daya dalam Amazon EFS](#).

- Instal botocore. Klien EFS menggunakan botocore untuk mengambil alamat IP target mount ketika nama DNS sistem file tidak dapat diselesaikan saat memasang sistem file di VPC lain. Untuk informasi selengkapnya, lihat [Menginstal botocore di file](#) amazon-efs-utils README.
- Siapkan koneksi peering VPC atau gateway transit VPC.

Anda menghubungkan VPC klien dan VPC sistem file EFS Anda menggunakan koneksi peering VPC atau gateway transit VPC. Saat Anda menggunakan koneksi peering VPC atau gateway transit untuk menghubungkan VPC, instans Amazon EC2 yang ada di satu VPC dapat mengakses sistem file EFS di VPC lain, meskipun milik akun yang berbeda. VPCs

Gateway transit adalah hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. Untuk informasi selengkapnya tentang penggunaan gateway transit VPC, lihat [Memulai gateway transit di](#) Panduan Gerbang Transit VPC Amazon.

Koneksi peering VPC adalah koneksi jaringan antara dua VPCs. Jenis koneksi ini memungkinkan Anda untuk merutekan lalu lintas di antara mereka menggunakan alamat Internet Protocol pribadi versi 4 (IPv4) atau Internet Protocol versi 6 (IPv6). Anda dapat menggunakan VPC peering untuk terhubung VPCs dalam hal yang sama AWS Region atau di antaranya. Wilayah AWS Untuk informasi selengkapnya tentang peering VPC, lihat [Apa yang dimaksud dengan peering VPC?](#) dalam Panduan Peering Amazon VPC.

Untuk memastikan ketersediaan sistem file Anda yang tinggi, sebaiknya Anda selalu menggunakan alamat IP target pemasangan EFS yang berada di Availability Zone yang sama dengan klien NFS Anda. Jika Anda memasang sistem file EFS yang ada di akun lain, pastikan klien NFS dan target pemasangan EFS berada di ID Availability Zone yang sama. Persyaratan ini berlaku karena nama AZ dapat berbeda dari satu akun ke akun lainnya.

Untuk me-mount sistem file EFS di VPC lain menggunakan IAM atau titik akses

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Buat direktori untuk memasang sistem file menggunakan perintah berikut.

```
$ sudo mkdir /mnt/efs
```

3. Untuk me-mount sistem file menggunakan otorisasi IAM, gunakan perintah berikut:

```
$ sudo mount -t efs -o tls,iam file-system-dns-name /mnt/efs/
```

Untuk informasi selengkapnya tentang penggunaan otorisasi IAM dengan EFS, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#)

Untuk me-mount sistem file menggunakan titik akses EFS, gunakan perintah berikut:

```
$ sudo mount -t efs -o tls,accesspoint=access-point-id file-system-dns-name /mnt/efs/
```

Untuk informasi selengkapnya tentang titik akses EFS, lihat [Bekerja dengan titik akses](#).

Memasang sistem file EFS dari yang berbeda AWS Region

Jika Anda memasang sistem file EFS Anda dari VPC lain yang berbeda AWS Region dari sistem file, Anda perlu mengedit file. `efs-utils.conf` Di `/dist/efs-utils.conf`, cari baris berikut:

```
#region = us-east-1
```

Hapus komentar baris, dan ganti nilai untuk ID wilayah di mana sistem file berada, jika tidak ada. `us-east-1`

Menggunakan Network File System untuk me-mount sistem file EFS

Berikut ini, pelajari cara menginstal klien Network File System (NFS) dan cara memasang sistem file Amazon EFS Anda pada instans Amazon EC2. Anda juga dapat menemukan penjelasan tentang

mount perintah dan opsi yang tersedia untuk menentukan nama Sistem Nama Domain (DNS) sistem file Anda dalam perintah. mount Selain itu, Anda dapat menemukan cara menggunakan file `fstab` untuk secara otomatis memasang kembali sistem file Anda setelah sistem apa pun dimulai ulang.

Note

Di bagian ini, Anda dapat mempelajari cara memasang sistem file Amazon EFS Anda tanpa `amazon-efs-utils` paket. Untuk menggunakan enkripsi data dalam perjalanan dengan sistem file Anda, Anda harus me-mount sistem file Anda dengan Transport Layer Security (TLS). Untuk melakukannya, kami sarankan menggunakan `amazon-efs-utils` paket. Untuk informasi selengkapnya, lihat [Menginstal klien Amazon EFS](#).

Topik

- [Prasyarat](#)
- [Dukungan NFS](#)
- [Menginstal klien NFS](#)
- [Pengaturan pemasangan NFS yang disarankan](#)
- [Pemasangan di Amazon EC2 dengan nama DNS](#)
- [Pemasangan dengan alamat IP](#)

Prasyarat

Sebelum Anda dapat memasang sistem file, pastikan Anda memenuhi persyaratan berikut:

- Buat, konfigurasi, dan luncurkan AWS sumber daya terkait Anda. Untuk petunjuk, lihat [Memulai dengan Amazon EFS](#).
- Buat grup keamanan VPC untuk instans Amazon EC2 Anda dan pasang target dengan akses masuk dan keluar yang diperlukan. Untuk informasi selengkapnya, lihat [Menggunakan grup keamanan VPC](#).

Dukungan NFS

Amazon EFS mendukung protokol Network File System versi 4.0 dan 4.1 (NFSv4) saat memasang sistem file Anda di instans Amazon EC2. Meskipun NFSv4.0 didukung, kami menyarankan Anda

menggunakan NFSv4.1. Memasang sistem file Amazon EFS di instans Amazon EC2 Anda juga memerlukan klien NFS yang mendukung protokol NFSv4 pilihan Anda. Instans Amazon EC2 Mac yang menjalankan macOS Big Sur hanya mendukung NFS v4.0.

Amazon EFS tidak mendukung opsi `nconnect` pemasangan.

Note

Untuk kernel Linux versi 5.4.*, klien Linux NFS menggunakan `readahead_kb` nilai default 128 KB. Kami merekomendasikan untuk meningkatkan nilai ini menjadi 15 MB. Untuk informasi selengkapnya, lihat [Mengoptimalkan ukuran `readahead\_kb` NFS](#).

Untuk kinerja optimal dan untuk menghindari berbagai bug klien NFS yang diketahui, kami sarankan bekerja dengan kernel Linux baru-baru ini. Jika Anda menggunakan distribusi Linux perusahaan, kami merekomendasikan hal berikut:

- Amazon Linux 2
- Amazon Linux 2017.09 atau yang lebih baru
- Red Hat Enterprise Linux (dan turunannya seperti CentOS) versi 8 dan yang lebih baru
- Ubuntu 16.04 LTS dan yang lebih baru
- SLES 12 Sp2 atau yang lebih baru

Jika Anda menggunakan distribusi lain atau kernel khusus, kami merekomendasikan kernel versi 4.3 atau yang lebih baru. Untuk memecahkan masalah yang terkait dengan AMI atau versi kernel tertentu saat menggunakan Amazon EFS dari instans EC2, lihat [Memecahkan masalah AMI dan kernel](#)

Note

Memasang sistem file EFS dengan instans Amazon EC2 yang menjalankan Microsoft Windows tidak didukung.

Menginstal klien NFS

Untuk memasang sistem file EFS Anda di instans Amazon EC2 Anda, pertama-tama Anda harus menginstal klien NFS. Untuk terhubung ke instans EC2 Anda dan menginstal klien NFS, Anda

memerlukan nama DNS publik dari instans EC2 dan nama pengguna untuk masuk. Nama pengguna untuk contoh Anda biasanya `ec2-user`.

Untuk menghubungkan instans EC2 Anda dan menginstal klien NFS

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.

File kunci tidak dapat dilihat secara publik untuk SSH. Anda dapat menggunakan `chmod 400 filename.pem` perintah untuk mengatur izin ini. Untuk informasi selengkapnya, lihat [Membuat key pair untuk instans Amazon EC2 Anda](#).

2. (Opsional) Dapatkan pembaruan dan reboot.

```
$ sudo yum -y update
$ sudo reboot
```

3. Setelah reboot, sambungkan kembali ke instans EC2 Anda.
4. Instal klien NFS.

Jika Anda menggunakan Amazon Linux AMI atau Red Hat Linux AMI, instal klien NFS dengan perintah berikut.

```
$ sudo yum -y install nfs-utils
```

Jika Anda menggunakan Ubuntu Amazon EC2 AMI, instal klien NFS dengan perintah berikut.

```
$ sudo apt-get -y install nfs-common
```

5. Mulai layanan NFS menggunakan perintah berikut:

```
$ sudo service nfs-server start
```

6. Verifikasi bahwa layanan NFS dimulai, sebagai berikut.

```
$ sudo service nfs-server status
Redirecting to /bin/systemctl status nfs.service
# nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor preset: disabled)
   Active: active (exited) since Wed 2019-10-30 16:13:44 UTC; 5s ago
```

```
Process: 29446 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
Process: 29441 ExecStartPre=/bin/sh -c /bin/kill -HUP `cat /run/gssproxy.pid` (code=exited, status=0/SUCCESS)
Process: 29439 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
Main PID: 29446 (code=exited, status=0/SUCCESS)
CGroup: /system.slice/nfs-server.service
```

Jika Anda menggunakan kernel khusus (yaitu, jika Anda membuat AMI khusus), Anda harus menyertakan minimal modul kernel NFSv4 klien.1 dan pembantu pemasangan NFS4 ruang pengguna yang tepat.

Note

Jika Anda memilih Amazon Linux AMI 2016.03.0 atau Amazon Linux AMI 2016.09.0 saat meluncurkan instans Amazon EC2 Anda, Anda tidak perlu menginstal `nfs-utils` karena sudah termasuk dalam AMI secara default.

Berikutnya: Pasang sistem file Anda

Gunakan salah satu prosedur berikut untuk me-mount sistem file Anda.

- [Pemasangan di Amazon EC2 dengan nama DNS](#)
- [Pemasangan dengan alamat IP](#)
- [Secara otomatis memasang sistem file EFS](#)

Pengaturan pemasangan NFS yang disarankan

Kami merekomendasikan nilai berikut untuk opsi pemasangan di Linux:

- `noresvport` Memberitahu klien NFS untuk menggunakan port sumber Transmission Control Protocol (TCP) non-privileged baru ketika koneksi jaringan dibangun kembali. Perangkat lunak klien NFS yang disertakan dalam versi kernel Linux yang lebih lama (versi v5.4 dan di bawahnya) mencakup perilaku yang menyebabkan klien NFS, setelah terputus, mencoba menghubungkan kembali pada port sumber TCP yang sama. Perilaku ini tidak sesuai dengan TCP RFC, dan dapat mencegah klien ini membangun kembali koneksi dengan cepat ke sistem file EFS.

Menggunakan `noresvport` opsi membantu memastikan bahwa klien NFS terhubung kembali secara transparan ke sistem file EFS Anda, menjaga ketersediaan tanpa gangguan saat menyambung kembali setelah peristiwa pemulihan jaringan.

 Important

Kami sangat menyarankan untuk menggunakan opsi `noresvport` pemasangan untuk membantu memastikan bahwa sistem file EFS Anda memiliki ketersediaan tanpa gangguan setelah peristiwa penyambungan ulang atau pemulihan jaringan.

Pertimbangkan untuk menggunakan [EFS mount helper](#) untuk me-mount sistem file Anda. EFS mount helper menggunakan opsi pemasangan NFS yang dioptimalkan untuk sistem file Amazon EFS.

- `rsize=1048576`— Menetapkan jumlah maksimum byte data yang klien NFS dapat menerima untuk setiap permintaan READ jaringan. Nilai ini berlaku saat membaca data dari file pada sistem file EFS. Kami menyarankan Anda menggunakan ukuran sebesar mungkin (hingga `1048576`) untuk menghindari penurunan kinerja.
- `wsiz=1048576`— Menetapkan jumlah maksimum byte data yang klien NFS dapat mengirim untuk setiap permintaan WRITE jaringan. Nilai ini berlaku saat menulis data ke file pada sistem file EFS. Kami menyarankan Anda menggunakan ukuran sebesar mungkin (hingga `1048576`) untuk menghindari penurunan kinerja.
- `hard`— Menetapkan perilaku pemulihan klien NFS setelah waktu permintaan NFS habis, sehingga permintaan NFS dicoba ulang tanpa batas hingga server membalas. Kami menyarankan Anda menggunakan opsi `hard mount` (`hard`) untuk memastikan integritas data. Jika Anda menggunakan `soft mount`, atur `timeo` parameter ke setidaknya 150 desidetik (15 detik). Melakukannya membantu meminimalkan risiko korupsi data yang melekat pada `soft mount`.
- `timeo=600`— Menetapkan nilai batas waktu yang digunakan klien NFS untuk menunggu respons sebelum mencoba ulang permintaan NFS ke 600 desidetik (60 detik). Jika Anda harus mengubah parameter batas waktu (`timeo`), kami sarankan Anda menggunakan nilai minimal 150, yang setara dengan 15 detik. Melakukannya membantu menghindari penurunan kinerja.
- `retrans=2`— Set ke 2 berapa kali klien NFS mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.
- `_netdev`— Saat hadir/`etc/fstab`, mencegah klien mencoba memasang sistem file EFS hingga jaringan diaktifkan.

- `nofail`— Jika instans EC2 Anda perlu memulai terlepas dari status sistem file EFS yang Anda pasang, tambahkan `nofail` opsi ke entri sistem file Anda di `/etc/fstab` file Anda.

Jika Anda tidak menggunakan default sebelumnya, perhatikan hal berikut:

- Secara umum, hindari menyetel opsi pemasangan lain yang berbeda dari default, yang dapat menyebabkan penurunan kinerja dan masalah lainnya. Misalnya, mengubah ukuran buffer baca atau tulis atau menonaktifkan caching atribut dapat mengakibatkan penurunan kinerja.
- Amazon EFS mengabaikan port sumber. Jika Anda mengubah port sumber Amazon EFS, itu tidak berpengaruh apa pun.
- Amazon EFS tidak mendukung opsi `nconnect` pemasangan.
- Amazon EFS tidak mendukung varian keamanan Kerberos mana pun. Misalnya, perintah `mount` berikut gagal.

```
$ mount -t nfs4 -o krb5p <DNS_NAME>:/ /efs/
```

- Kami menyarankan Anda me-mount sistem file Anda menggunakan nama DNS-nya. Nama ini diatasi ke alamat IP target pemasangan Amazon EFS di Availability Zone yang sama dengan instans Amazon EC2 Anda. Jika Anda menggunakan target pemasangan di Availability Zone yang berbeda dengan instans Amazon EC2, Anda dikenakan biaya EC2 standar untuk data yang dikirim di seluruh Availability Zone. Anda juga mungkin melihat peningkatan latensi untuk operasi sistem file.
- Untuk opsi pemasangan lainnya, dan penjelasan rinci tentang default, lihat dokumentasi Linux.

Pemasangan di Amazon EC2 dengan nama DNS

Note

Sebelum memasang sistem file Anda, Anda perlu menambahkan aturan ke grup keamanan target mount yang memungkinkan akses NFS masuk pada port 2049 dari grup keamanan EC2. Untuk informasi selengkapnya, lihat [Menggunakan grup keamanan VPC](#).

- Nama DNS sistem file — Menggunakan nama DNS sistem file adalah opsi pemasangan paling sederhana Anda. Nama DNS sistem file secara otomatis menyelesaikan alamat IP target mount di Availability Zone dari instans Amazon EC2 yang menghubungkan. Anda bisa mendapatkan nama

DNS dari konsol, atau jika Anda memiliki ID sistem file, Anda dapat membuatnya menggunakan konvensi berikut.

```
file-system-id.efs.aws-region.amazonaws.com
```

Note

Resolusi DNS untuk nama DNS sistem file mengharuskan sistem file Amazon EFS memiliki target pemasangan di Availability Zone yang sama dengan instance klien.

- Dengan menggunakan nama DNS sistem file, Anda dapat memasang sistem file pada instans Amazon EC2 Linux Anda dengan perintah berikut.

```
sudo mount -t nfs -o
nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport file-
system-id.efs.aws-region.amazonaws.com:/ efs-mount-point
```

- Dengan menggunakan nama DNS sistem file, Anda dapat memasang sistem file di instans Amazon EC2 Mac yang menjalankan versi macOS yang didukung (Big Sur, Monterey, Ventura) dengan perintah berikut.

```
sudo mount -t nfs -o
nfsvers=4.0,rsz=65536,wsz=65536,hard,timeo=600,retrans=2,noresvport,mountport=2049 fil
system-id.efs.aws-region.amazonaws.com:/ efs
```

Important

Anda harus menggunakan `mountport=2049` agar berhasil terhubung ke sistem file EFS saat memasang pada instans EC2 Mac yang menjalankan mendukung versi macOS.

- Pasang nama DNS target - Pada bulan Desember 2016, kami memperkenalkan nama DNS sistem file. Kami terus memberikan nama DNS untuk setiap target pemasangan Availability Zone untuk kompatibilitas mundur. Bentuk generik dari nama DNS target mount adalah sebagai berikut.

```
availability-zone.file-system-id.efs.aws-region.amazonaws.com
```

 Note

Resolusi nama DNS target mount di seluruh Availability Zones didukung.

Dalam beberapa kasus, Anda mungkin menghapus target mount dan kemudian membuat yang baru di Availability Zone yang sama. Dalam kasus seperti itu, nama DNS untuk target mount baru di Availability Zone tersebut sama dengan nama DNS untuk target mount lama.

Anda dapat melihat dan menyalin perintah yang tepat untuk me-mount sistem file Anda di kotak dialog Lampirkan.

Untuk melihat perintah mount untuk sistem file Anda

1. Di konsol Amazon EFS, pilih sistem file yang ingin Anda pasang untuk menampilkan halaman detailnya.
2. Untuk menampilkan perintah mount yang akan digunakan untuk sistem file ini, pilih Lampirkan di kanan atas.

Layar Lampirkan menampilkan perintah yang tepat untuk digunakan untuk memasang sistem file.

3. Tampilan Mount via DNS default menampilkan perintah untuk me-mount sistem file menggunakan nama DNS sistem file saat memasang dengan EFS mount helper atau klien NFS.

Untuk daftar AWS Region s yang mendukung Amazon EFS, lihat [Amazon Elastic File System](#) di file Referensi Umum AWS.

Untuk menggunakan nama DNS dalam perintah mount, pastikan hal berikut sudah benar:

- Instans EC2 penghubung harus berada di dalam VPC dan harus dikonfigurasi untuk menggunakan server DNS yang disediakan oleh Amazon. Untuk informasi tentang server DNS Amazon, lihat [set opsi DHCP di Amazon VPC di Panduan Pengguna](#) Amazon VPC.
- VPC dari instans EC2 penghubung harus mengaktifkan Resolusi DNS dan Nama Host DNS. Untuk informasi selengkapnya, lihat [atribut DNS untuk VPC Anda](#) di Panduan Pengguna Amazon VPC.

- Instans EC2 penghubung harus berada di dalam VPC yang sama dengan sistem file EFS. Untuk informasi lebih lanjut tentang mengakses dan memasang sistem file dari lokasi lain atau dari VPC yang berbeda, [Prasyarat](#) lihat dan. [Tutorial: Pasang sistem file dari VPC yang berbeda](#)

Note

Kami menyarankan Anda menunggu 90 detik setelah membuat target mount sebelum Anda me-mount sistem file Anda. Penantian ini memungkinkan catatan DNS menyebar sepenuhnya di AWS Region tempat sistem file berada.

Pemasangan dengan alamat IP

Sebagai alternatif untuk memasang sistem file Amazon EFS Anda dengan nama DNS, instans Amazon EC2 dapat memasang sistem file menggunakan alamat IP target mount. Pemasangan berdasarkan alamat IP berfungsi di lingkungan di mana DNS dinonaktifkan, seperti VPCs dengan nama host DNS dinonaktifkan.

Anda juga dapat mengonfigurasi pemasangan sistem file menggunakan alamat IP target mount sebagai opsi fallback untuk aplikasi yang dikonfigurasi untuk me-mount sistem file menggunakan nama DNS-nya secara default. Saat menghubungkan ke alamat IP target mount, instans EC2 harus dipasang menggunakan alamat IP target mount di Availability Zone yang sama dengan instance penghubung.

Anda dapat melihat dan menyalin perintah yang tepat untuk me-mount sistem file Anda di kotak dialog Lampirkan.

Note

Sebelum memasang sistem file Anda, Anda perlu menambahkan aturan untuk grup keamanan target mount untuk memungkinkan akses NFS masuk pada port 2049 dari grup keamanan EC2. Untuk informasi selengkapnya, lihat [Menggunakan grup keamanan VPC](#).

Untuk melihat dan menyalin perintah yang tepat untuk me-mount sistem file EFS Anda menggunakan alamat IP target mount

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.

2. Di konsol Amazon EFS, pilih sistem file yang ingin Anda pasang untuk menampilkan halaman detailnya.
3. Untuk menampilkan perintah mount yang akan digunakan untuk sistem file ini, pilih Lampirkan di kanan atas.
4. Layar Lampirkan menampilkan perintah yang tepat untuk digunakan untuk memasang sistem file.

Pilih Mount via IP untuk menampilkan perintah untuk me-mount sistem file menggunakan alamat IP target mount di Availability Zone yang dipilih dengan klien NFS.

- Menggunakan alamat IP target mount dalam mount perintah, Anda dapat me-mount sistem file pada instans Amazon EC2 Linux Anda dengan perintah berikut.

```
sudo mount -t nfs -o
nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport mount-
target-IP:/ /efs
```

- Dengan menggunakan alamat IP target mount dalam mount perintah, Anda dapat memasang sistem file di instans Amazon EC2 Mac yang menjalankan macOS Big Sur dengan perintah berikut.

```
sudo mount -t nfs -o
nfsvers=4.0,rsz=65536,wsz=65536,hard,timeo=600,retrans=2,noresvport,mountport=2049 mount-
target-IP:/ /efs
```

Important

Anda harus menggunakan mountport=2049 agar berhasil terhubung ke sistem file EFS saat memasang pada instans EC2 Mac yang menjalankan macOS Big Sur.

Pemasangan dengan alamat IP di AWS CloudFormation

Anda juga dapat me-mount sistem file Anda menggunakan alamat IP dalam CloudFormation template. Untuk informasi selengkapnya, lihat [storage-efs-mountfilesystem-ip-addr.config](#) di elastic-beanstalk-samplesawsdocs/ repositori untuk file konfigurasi yang disediakan komunitas. GitHub

Secara otomatis memasang sistem file EFS

Anda dapat menggunakan EFS mount helper atau NFS untuk mengonfigurasi instans Amazon EC2 agar secara otomatis memasang sistem file EFS saat instance dimulai.

- Menggunakan EFS mount helper:
 - Lampirkan sistem file EFS saat Anda membuat instans Linux EC2 baru menggunakan wizard instans peluncuran EC2.
 - Perbarui `/etc/fstab` file EC2 dengan entri untuk sistem file EFS.
- Menggunakan [NFS tanpa EFS mount helper](#) untuk memperbarui `/etc/fstab` file EC2, untuk instans EC2 Linux dan Mac.

Note

Helper pemasangan EFS tidak mendukung pemasangan otomatis pada instans EC2 Mac yang menjalankan macOS Big Sur atau Monterey. Sebagai gantinya, Anda dapat menggunakan [NFS untuk mengonfigurasi file `/etc/fstab` pada instance EC2 Mac untuk secara otomatis memasang sistem file EFS](#).

Topik

- [Mengaktifkan pemasangan otomatis pada instans EC2 Linux baru](#)
- [Mengaktifkan pemasangan otomatis pada instans EC2 Linux yang ada](#)
- [Mengaktifkan pemasangan otomatis pada instans EC2 Linux atau Mac menggunakan NFS](#)

Mengaktifkan pemasangan otomatis pada instans EC2 Linux baru

Saat membuat instans EC2 Linux baru menggunakan wizard instans peluncuran Amazon EC2, Anda dapat mengonfigurasinya untuk memasang sistem file Amazon EFS secara otomatis. Instans EC2 memasang sistem file secara otomatis instans pertama kali diluncurkan dan juga setiap kali restart.

Metode ini menggunakan EFS mount helper untuk me-mount sistem file memperbarui file `/etc/fstab` pada instance EC2. Mount helper adalah bagian dari [amazon-efs-utils](#) seperangkat alat.

Note

Sistem file EFS tidak mendukung pemasangan pada instans EC2 Mac yang menjalankan macOS Big Sur atau Monterey saat peluncuran instans.

Note

Anda tidak dapat menggunakan Amazon EFS dengan instans EC2 berbasis Microsoft Windows.

Sebelum Anda dapat meluncurkan dan terhubung ke instans EC2, Anda perlu membuat key pair. Untuk informasi selengkapnya, lihat [pasangan kunci Amazon EC2 dan instans Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk membuat key pair.

Untuk mengonfigurasi instans EC2 Anda untuk memasang sistem file EFS secara otomatis saat diluncurkan

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Pilih Luncurkan Instans.
3. Pada Langkah 1: Pilih Amazon Machine Image (AMI), temukan Amazon Linux AMI di bagian atas daftar dan pilih Pilih.
4. Pada Langkah 2: Pilih Jenis Instance, pilih Berikutnya: Konfigurasikan Detail Instance.
5. Pada Langkah 3: Konfigurasikan Detail Instance, berikan informasi berikut:
 - Untuk Network, pilih entri untuk VPC yang sama dengan sistem file EFS yang Anda pasang.
 - Untuk Subnet, pilih subnet default di Availability Zone apa pun.
 - Untuk sistem File, pilih sistem file EFS yang ingin Anda pasang. Jalur yang ditampilkan di sebelah ID sistem file adalah titik pemasangan yang akan digunakan instans EC2, yang dapat Anda ubah.
 - Di bawah Detail Lanjutan, data Pengguna dibuat secara otomatis, dan menyertakan perintah yang diperlukan untuk memasang sistem file EFS yang Anda tentukan di bawah Sistem file.
6. Pilih Berikutnya: Tambahkan Penyimpanan.
7. Pilih Berikutnya: Tambahkan Tanda.
8. Beri nama instans Anda dan pilih Berikutnya: Konfigurasi Grup Keamanan.

9. Pada Langkah 6: Konfigurasi Grup Keamanan, atur Tetapkan grup keamanan ke Pilih grup keamanan yang ada. Pilih grup keamanan default untuk memastikan dapat mengakses sistem file EFS Anda.

Anda tidak dapat mengakses instans EC2 Anda dengan Secure Shell (SSH) menggunakan grup keamanan ini. Untuk akses dengan SSH, nanti Anda dapat mengedit keamanan default dan menambahkan aturan untuk mengizinkan SSH atau grup keamanan baru yang memungkinkan SSH. Anda dapat menggunakan pengaturan berikut:

- Jenis: SSH
 - Protokol: TCP
 - Rentang Pelabuhan: 22
 - Sumber: Anywhere 0.0.0.0/0
10. Pilih Tinjau dan Luncurkan.
 11. Pilih Luncurkan.
 12. Pilih kotak centang untuk key pair yang Anda buat, lalu pilih Launch Instances.

Instans EC2 Anda sekarang dikonfigurasi untuk memasang sistem file EFS saat diluncurkan dan kapan pun di-boot ulang.

Mengaktifkan pemasangan otomatis pada instans EC2 Linux yang ada

File `/etc/fstab` berisi informasi tentang sistem file. Perintah `mount -a`, yang berjalan selama start-up instance, memasang semua sistem file yang tercantum di dalamnya. `/etc/fstab` Dalam prosedur ini, Anda akan memperbarui instans Linux `/etc/fstab` Amazon EC2 secara manual sehingga instans menggunakan EFS mount helper untuk secara otomatis menjalankan ulang sistem file EFS saat instance dimulai ulang.

Note

Sistem file EFS tidak mendukung pemasangan otomatis menggunakan `/etc/fstab` EFS mount helper pada instans EC2 Mac yang menjalankan macOS Big Sur atau Monterey. Sebagai gantinya, Anda dapat menggunakan [NFS /etc/fstab](#) untuk secara otomatis memasang sistem file Anda pada instans EC2 Mac yang menjalankan macOS Big Sur dan Monterey.

Metode ini menggunakan EFS mount helper untuk me-mount sistem file. Mount helper adalah bagian dari `amazon-efs-utils` seperangkat alat.

`amazon-efs-utils` Alat ini tersedia untuk instalasi di Amazon Linux dan Amazon Linux 2 Amazon Machine Images (AMIs). Untuk informasi selengkapnya tentang `amazon-efs-utils`, lihat [Menginstal klien Amazon EFS](#). Jika Anda menggunakan distribusi Linux lain, seperti Red Hat Enterprise Linux (RHEL), buat dan instal `amazon-efs-utils` secara manual. Untuk informasi selengkapnya, lihat [Menginstal klien Amazon EFS di distribusi Linux lainnya](#).

Prasyarat

Persyaratan berikut harus ada sebelum Anda berhasil menerapkan prosedur ini:

- Anda telah membuat sistem file EFS yang ingin dipasang ulang secara otomatis. Untuk informasi selengkapnya, lihat [Buat cepat menggunakan konsol](#).
- Anda telah membuat instance EC2 Linux yang ingin Anda konfigurasi untuk secara otomatis memasang kembali sistem file EFS.
- EFS mount helper diinstal pada instans EC2 Linux. Untuk informasi selengkapnya, lihat [Menginstal klien Amazon EFS](#).

Perbarui berkas `/etc/fstab`

Lakukan langkah-langkah berikut untuk memperbarui `/etc/fstab` pada instance EC2 Linux sehingga instance menggunakan EFS mount helper untuk secara otomatis menjalankan ulang sistem file EFS saat instance dimulai ulang.

Untuk memperbarui berkas `/etc/fstab` pada instans EC2 Anda

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Buka `/etc/fstab` file di editor.
3. Untuk pemasangan otomatis menggunakan otorisasi IAM atau titik akses EFS:
 - Untuk secara otomatis me-mount dengan otorisasi IAM ke instans EC2 yang memiliki profil instance, tambahkan baris berikut ke file. `/etc/fstab`

```
file-system-id:/ efs-mount-point efs _netdev,noresvport,tls,iam 0 0
```

- Untuk secara otomatis me-mount dengan otorisasi IAM ke instance Linux menggunakan file kredensyal, tambahkan baris berikut ke file. `/etc/fstab`

```
file-system-id:/ efs-mount-point efs  
_netdev,noresvport,tls,iam,awsprofile=namedprofile 0 0
```

- Untuk memasang sistem file secara otomatis menggunakan titik akses EFS, tambahkan baris berikut ke `/etc/fstab` file.

```
file-system-id:/ efs-mount-point efs _netdev,noresvport,tls,accesspoint=access-  
point-id 0 0
```

Warning

Gunakan opsi `_netdev`, yang digunakan untuk mengidentifikasi sistem file jaringan, ketika memasang sistem file Anda secara otomatis. Jika `_netdev` hilang, instans EC2 Anda mungkin berhenti merespons. Hasil ini didapatkan karena sistem file jaringan perlu diinisialisasi setelah instans komputasi memulai jaringannya. Untuk informasi selengkapnya, lihat [Pemasangan otomatis gagal dan instans tidak responsif](#).

Untuk informasi selengkapnya, lihat [Pemasangan dengan otorisasi IAM](#) dan [Pemasangan dengan titik akses EFS](#).

4. Simpan perubahan pada file.

Note

Dalam beberapa kasus, instans EC2 Anda mungkin perlu memulai terlepas dari status sistem file EFS yang Anda pasang. Dalam kasus seperti itu, tambahkan `nofail` opsi ke entri sistem file Anda di `/etc/fstab` file Anda.

Baris kode yang Anda tambahkan ke `/etc/fstab` file melakukan hal berikut.

Bidang	Deskripsi
<i>file-system-id</i> :/	ID untuk sistem file EFS Anda. Anda bisa mendapatkan ID ini dari konsol atau secara terprogram dari CLI atau SDK. AWS
<i>efs-mount-point</i>	Titik pemasangan untuk sistem file EFS pada instans EC2 Anda.
efs	Jenis sistem file. Saat Anda menggunakan mount helper, tipe ini selaluefs.
mount options	Opsi pemasangan untuk sistem file. Ini adalah daftar opsi berikut yang dipisahkan koma: • <code>_netdev</code>— Opsi ini memberi tahu sistem operasi bahwa sistem file berada pada perangkat yang memerlukan akses jaringan. Opsi ini mencegah instans memasang sistem file sampai jaringan telah diaktifkan pada klien. • <code>noresvport</code> Memberitahu klien NFS untuk menggunakan port sumber Transmission Control Protocol (TCP) baru ketika koneksi jaringan dibangun kembali. Melakukan hal ini membantu memastikan bahwa sistem file EFS memiliki ketersediaan tanpa gangguan setelah peristiwa pemulihan jaringan. • <code>tls</code>— Memungkinkan enkripsi data dalam perjalanan. • <code>iam</code>— Gunakan opsi ini untuk me-mount dengan otorisasi IAM ke instans EC2 yang memiliki profil instans. Menggunakan opsi <code>iam</code> mount juga membutuhkan penggunaan <code>tls</code> opsi. Untuk informasi selengkapnya, lihat Menggunakan IAM untuk mengontrol akses ke sistem file. • <code>awsprofile= <i>namedprofile</i></code> — Gunakan opsi ini dengan <code>tls</code> opsi <code>iam</code> dan untuk memasang dengan otorisasi IAM ke instance Linux menggunakan file kredensial. Untuk informasi selengkapnya tentang titik akses EFS, lihat Menggunakan IAM untuk mengontrol akses ke sistem file. • <code>accesspoint= <i>access-point-id</i></code> — Gunakan opsi ini dengan <code>tls</code> opsi untuk me-mount menggunakan titik akses EFS. Untuk

Bidang	Deskripsi
	informasi selengkapnya tentang titik akses EFS, lihat Bekerja dengan titik akses .
0	Nilai bukan nol menunjukkan bahwa sistem file harus didukung oleh dump. Untuk EFS, nilai ini seharusnya 0.
0	Urutan di mana fsck memeriksa sistem file saat boot. Untuk sistem file EFS, nilai ini 0 harus menunjukkan bahwa tidak fsck boleh berjalan saat start-up.

Mengaktifkan pemasangan otomatis pada instans EC2 Linux atau Mac menggunakan NFS

Menggunakan NFS tanpa EFS mount helper untuk memperbarui file Amazon `/etc/fstab` EC2, untuk instans EC2 Linux dan Mac.

Untuk memperbarui `/etc/fstab` file pada instans EC2 Anda

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Buka `/etc/fstab` file di editor.
3. Untuk secara otomatis memasang sistem file menggunakan NFS, bukan EFS mount helper, tambahkan baris berikut ke file. `/etc/fstab`
 - Ganti `file_system_id` dengan ID sistem file yang Anda pasang.
 - Ganti `aws-region` dengan AWS Region yang ada di sistem file, seperti `us-east-1`.
 - Ganti `mount_point` dengan titik pasang sistem file.

```
file_system_id.efs.aws-region.amazonaws.com:/ mount_point nfs4
nfsvers=4.1,rsiz=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport,_netdev
0 0
```

Baris kode yang Anda tambahkan ke `/etc/fstab` file melakukan hal berikut.

Bidang	Deskripsi
<i>file-system-id</i> :/	ID untuk sistem file EFS Anda. Anda bisa mendapatkan ID ini dari konsol atau secara terprogram dari CLI atau SDK. AWS
<i>efs-mount-point</i>	Titik pemasangan untuk sistem file EFS pada instans EC2 Anda.
nfs4	Menentukan jenis sistem file.
mount options	Daftar opsi mount yang dipisahkan koma untuk sistem file: • <code>nfsvers=4.1</code> — menentukan menggunakan NFS v4.1. • <code>rsz=1048576</code> Untuk meningkatkan kinerja, tetapkan jumlah maksimum byte data yang dapat diterima klien NFS untuk setiap permintaan READ jaringan saat membaca data dari file pada sistem file EFS. 1048576 adalah ukuran terbesar yang mungkin. • <code>wsz=1048576</code> Untuk meningkatkan kinerja, tetapkan jumlah maksimum byte data yang dapat dikirim klien NFS untuk setiap permintaan WRITE jaringan saat menulis data ke file pada sistem file EFS. 1048576 adalah ukuran terbesar yang mungkin. • <code>hard</code>— Menetapkan perilaku pemulihan klien NFS setelah waktu permintaan NFS habis, sehingga permintaan NFS dicoba ulang tanpa batas hingga server membalas. Kami menyarankan Anda menggunakan opsi hard mount (<code>hard</code>) untuk memastikan integritas data. Jika Anda menggunakan soft mount, atur <code>timeo</code> parameter ke setidaknya 150 desidetik (15 detik). Melakukannya membantu meminimalkan risiko korupsi data yang melekat pada soft mount. • <code>timeo=600</code> — Menetapkan nilai batas waktu yang digunakan klien NFS untuk menunggu respons sebelum mencoba ulang permintaan ke 600 desidetik (60 detik). Jika Anda harus mengubah parameter batas waktu (<code>timeo</code>), kami sarankan Anda menggunakan nilai minimal 150, yang setara dengan 15 detik. Melakukannya membantu menghindari penurunan kinerja. • <code>retrans=2</code> — Set ke 2 berapa kali klien NFS mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.

Bidang	Deskripsi
	• <code>noresvport</code> Memberitahu klien NFS untuk menggunakan port sumber Transmission Control Protocol (TCP) baru ketika koneksi jaringan dibangun kembali. Melakukan hal ini membantu memastikan bahwa sistem file EFS memiliki ketersediaan tanpa gangguan setelah peristiwa pemulihan jaringan. • <code>_netdev</code>— Mencegah klien mencoba memasang sistem file EFS hingga jaringan diaktifkan.
0	Menentukan dump nilai; 0 memberitahu dump utilitas untuk tidak membuat cadangan sistem file.
0	Memberitahu <code>fsck</code> utilitas untuk tidak berjalan saat start-up.

Melepaskan sistem file

Sebelum Anda menghapus sistem file, kami sarankan Anda melepaskannya dari setiap instans Amazon EC2 yang terhubung dengannya. Anda dapat melepaskan sistem file pada instans Amazon EC2 Anda dengan menjalankan perintah `umount` pada instans itu sendiri. Anda tidak dapat melepas sistem file EFS melalui AWS CLI, file Konsol Manajemen AWS, atau melalui AWS SDK mana pun. Untuk melepas sistem file EFS yang terhubung ke instans EC2 yang menjalankan Linux, gunakan `umount` perintah sebagai berikut:

```
umount /mnt/efs
```

Kami menyarankan Anda untuk tidak menentukan pilihan `umount` lainnya. Hindari pengaturan pilihan `umount` lainnya yang berbeda dari default.

Anda dapat memverifikasi bahwa sistem file EFS Anda telah dilepas dengan menjalankan `df` perintah. Perintah ini menampilkan statistik penggunaan disk untuk sistem file yang saat ini dipasang pada instans Amazon EC2 berbasis Linux Anda. Jika sistem file EFS yang ingin Anda `umount` tidak tercantum dalam output `df` perintah, ini berarti bahwa sistem file dilepas.

Example— Identifikasi status pemasangan sistem file EFS dan lepaskan

```
$ df -T
Filesystem Type 1K-blocks Used Available Use% Mounted on
```

```
/dev/sda1 ext4 8123812 1138920 6884644 15% /  
availability-zone.file-system-id.efs.aws-region.amazonaws.com :/ nfs4 9007199254740992  
0 9007199254740992 0% /mnt/efs
```

```
$ umount /mnt/efs
```

```
$ df -T
```

```
Filesystem Type 1K-blocks Used Available Use% Mounted on  
/dev/sda1 ext4 8123812 1138920 6884644 15% /
```

Tutorial: Buat sistem file EFS dan pasang pada instans EC2 menggunakan AWS CLI

Buat sistem file EFS terenkripsi, pasang pada instans EC2 di VPC Anda, dan uji penyiapannya menggunakan file. AWS CLI

Note

Dalam [Memulai](#) tutorial, Anda menggunakan konsol untuk membuat sumber daya Amazon EC2 dan EFS. Dalam tutorial ini, Anda menggunakan AWS CLI untuk melakukan hal yang sama—terutama untuk membiasakan diri dengan Amazon EFS API.

Dalam tutorial ini, Anda membuat AWS sumber daya berikut di akun Anda:

- Sumber daya Amazon EC2:
 - Dua grup keamanan (untuk instans EC2 dan sistem file EFS Anda).

Anda menambahkan aturan ke grup keamanan ini untuk mengotorisasi akses masuk/keluar yang sesuai. Melakukan hal ini memungkinkan Anda EC2instance untuk terhubung ke sistem file melalui target mount dengan menggunakan port TCP NFSv4 .1 standar.

- Instans EC2 di VPC Anda.
- Sumber daya Amazon EFS:
 - Sebuah sistem file.
 - Target mount untuk sistem file Anda.

Untuk me-mount sistem file Anda pada instans EC2, Anda perlu membuat target mount di VPC Anda. Anda dapat membuat satu target pemasangan di setiap Availability Zone di VPC Anda. Untuk informasi selengkapnya, lihat [Cara kerja Amazon EFS](#).

Kemudian, Anda menguji sistem file pada instans EC2 Anda. Langkah pembersihan di akhir tutorial memberikan informasi bagi Anda untuk menghapus sumber daya ini.

Tutorial membuat semua sumber daya ini di Wilayah AS Barat (Oregon) (us-west-2). Apapun yang AWS Region Anda gunakan, pastikan untuk menggunakannya secara konsisten. Semua sumber daya Anda—VPC, sumber daya EC2, dan sumber daya EFS Anda—harus sama. AWS Region

Topik

- [Prasyarat](#)
- [Menyiapkan AWS CLI](#)
- [Langkah 1: Buat sumber daya EC2](#)
- [Langkah 2: Buat sumber daya EFS](#)
- [Langkah 3: Pasang sistem file pada instans EC2 dan uji](#)
- [Langkah 4: Membersihkan](#)

Prasyarat

- Anda dapat menggunakan kredensial root Anda Akun AWS untuk masuk ke konsol dan mencoba latihan memulai. Namun, AWS Identity and Access Management (IAM) merekomendasikan agar Anda tidak menggunakan kredensial root Anda. Akun AWS Sebagai gantinya, buat pengguna administrator di akun Anda dan gunakan kredensial tersebut untuk mengelola sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Menetapkan Akun AWS akses untuk pengguna Pusat Identitas IAM di Panduan AWS IAM Identity Center](#) Pengguna.
- Anda dapat menggunakan VPC default atau VPC khusus yang telah Anda buat di akun Anda. Untuk panduan ini, konfigurasi VPC default berfungsi. Namun, jika Anda menggunakan VPC kustom, verifikasi hal berikut:
 - Nama host DNS diaktifkan. Untuk informasi selengkapnya, lihat [atribut DNS di VPC Anda di Panduan Pengguna Amazon VPC](#).
 - Gateway internet terpasang ke VPC Anda. Untuk informasi selengkapnya, lihat [Hubungkan ke internet menggunakan gateway internet](#) di Panduan Pengguna Amazon VPC.

- Subnet VPC dikonfigurasi untuk meminta alamat IP publik untuk instance yang diluncurkan di subnet VPC. Untuk informasi selengkapnya, lihat [Pengalaman IP untuk subnet Anda VPCs dan subnet](#) di Panduan Pengguna Amazon VPC.
- Tabel rute VPC mencakup aturan untuk mengirim semua lalu lintas ke Internet ke gateway Internet.
- Anda perlu mengatur AWS CLI dan menambahkan profil adminuser.

Menyiapkan AWS CLI

Gunakan petunjuk berikut untuk mengatur AWS CLI dan profil pengguna.

Untuk mengatur AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk petunjuk, lihat [Memulai dengan AWS CLI](#) di Panduan AWS Command Line Interface Pengguna.
2. Tetapkan profil.

Anda menyimpan kredensial pengguna dalam file. AWS CLI config Contoh perintah CLI dalam tutorial ini menentukan profil adminuser. Buat profil adminuser dalam file. config Anda juga dapat mengatur profil pengguna administrator sebagai default dalam config file seperti yang ditunjukkan.

```
[profile adminuser]
aws_access_key_id = admin user access key ID
aws_secret_access_key = admin user secret access key
region = us-west-2

[default]
aws_access_key_id = admin user access key ID
aws_secret_access_key = admin user secret access key
region = us-west-2
```

Profil sebelumnya juga menetapkan default. AWS Region Jika Anda tidak menentukan Region dalam perintah CLI, wilayah us-west-2 diasumsikan.

3. Verifikasi pengaturan dengan memasukkan perintah berikut pada prompt perintah. Kedua perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

- Coba perintah bantuan

Anda juga dapat menentukan profil pengguna secara eksplisit dengan menambahkan parameter. `--profile`

```
aws help
```

```
aws help \  
--profile adminuser
```

Langkah 1: Buat sumber daya EC2

Dalam langkah ini, Anda melakukan hal berikut:

- Buat dua grup keamanan.
- Tambahkan aturan ke grup keamanan untuk mengotorisasi akses tambahan.
- Luncurkan instans EC2. Anda membuat dan memasang sistem file EFS pada instance ini di langkah berikutnya.

Langkah 1.1: Buat dua grup keamanan

Di bagian ini, Anda membuat grup keamanan di VPC untuk instans EC2 dan target pemasangan EFS. Kemudian dalam tutorial, Anda menetapkan grup keamanan ini ke instans EC2 dan target pemasangan EFS. Untuk informasi tentang grup keamanan, lihat Grup keamanan [Amazon EC2 untuk instans Linux](#).

Untuk membuat grup keamanan

1. Buat dua grup keamanan menggunakan perintah `create-security-group` CLI:
 - a. Buat grup keamanan (`efs-walkthrough1-ec2-sg`) untuk instans EC2 Anda, dan berikan ID VPC Anda.

```
$ aws ec2 create-security-group \  
--region us-west-2 \  
--group-name efs-walkthrough1-ec2-sg \  
--description "Amazon EFS walkthrough 1, SG for EC2 instance" \  

```

```
--vpc-id vpc-id-in-us-west-2 \
--profile adminuser
```

Tuliskan ID grup keamanan. Berikut ini adalah contoh respons.

```
{
  "GroupId": "sg-aexample"
}
```

Anda dapat menemukan ID VPC menggunakan perintah berikut.

```
$ aws ec2 describe-vpcs
```

- b. Buat grup keamanan (`efs-walkthrough1-mt-sg`) untuk target pemasangan EFS Anda. Anda perlu memberikan ID VPC Anda.

```
$ aws ec2 create-security-group \
--region us-west-2 \
--group-name efs-walkthrough1-mt-sg \
--description "Amazon EFS walkthrough 1, SG for mount target" \
--vpc-id vpc-id-in-us-west-2 \
--profile adminuser
```

Tuliskan ID grup keamanan. Berikut ini adalah contoh respons.

```
{
  "GroupId": "sg-aexample"
}
```

2. Verifikasi grup keamanan.

```
aws ec2 describe-security-groups \
--group-ids list of security group IDs separated by space \
--profile adminuser \
--region us-west-2
```

Keduanya seharusnya hanya memiliki satu aturan keluar yang memungkinkan semua lalu lintas pergi.

Di bagian selanjutnya, Anda mengotorisasi akses tambahan yang memungkinkan hal berikut:

- Memungkinkan Anda untuk terhubung ke instans EC2 Anda.
- Aktifkan lalu lintas antara instans EC2 dan target pemasangan EFS (yang dengannya Anda mengaitkan grup keamanan ini nanti dalam tutorial ini).

Langkah 1.2: Tambahkan aturan ke grup keamanan untuk mengotorisasi akses inbound/outbound

Pada langkah ini, Anda menambahkan aturan ke grup keamanan untuk mengotorisasi akses masuk/keluar.

Untuk menambahkan aturan

1. Otorisasi koneksi Secure Shell (SSH) yang masuk ke grup keamanan untuk instans EC2 (`efs-walkthrough1-ec2-sg`) Anda sehingga Anda dapat terhubung ke instans EC2 menggunakan SSH dari host mana pun.

```
$ aws ec2 authorize-security-group-ingress \
--group-id id of the security group created for EC2 instance \
--protocol tcp \
--port 22 \
--cidr 0.0.0.0/0 \
--profile adminuser \
--region us-west-2
```

Verifikasi bahwa grup keamanan memiliki aturan masuk dan keluar yang Anda tambahkan.

```
aws ec2 describe-security-groups \
--region us-west-2 \
--profile adminuser \
--group-id security-group-id
```

2. Otorisasi akses masuk ke grup keamanan untuk EFS mount target (`efs-walkthrough1-mt-sg`).

Pada prompt perintah, jalankan AWS CLI `authorize-security-group-ingress` perintah berikut menggunakan profil `adminuser` untuk menambahkan aturan masuk.


```
$ aws ec2 authorize-security-group-ingress \
--group-id ID of the security group created for Amazon EFS mount target \
--protocol tcp \
--port 2049 \
--source-group ID of the security group created for EC2 instance \
--profile adminuser \
--region us-west-2
```

3. Verifikasi bahwa kedua grup keamanan sekarang mengotorisasi akses masuk.

```
aws ec2 describe-security-groups \
--group-names efs-walkthrough1-ec2-sg efs-walkthrough1-mt-sg \
--profile adminuser \
--region us-west-2
```

Langkah 1.3: Luncurkan instans EC2

Pada langkah ini, Anda meluncurkan instans EC2.

Cara meluncurkan instans EC2

1. Kumpulkan informasi berikut yang perlu Anda berikan saat meluncurkan instans EC2:
 - Nama pasangan kunci. Untuk petunjuk membuat key pair, lihat [Membuat key pair untuk instans Amazon EC2 Anda di Panduan Pengguna Amazon EC2](#).
 - ID Gambar Mesin Amazon (AMI) yang ingin Anda luncurkan.

AWS CLI Perintah yang Anda gunakan untuk meluncurkan instans EC2 memerlukan ID Amazon Machine Image (AMI) yang ingin Anda gunakan sebagai parameter. Latihan ini menggunakan Amazon Linux HVM AMI.

Note

Anda dapat menggunakan sebagian besar tujuan umum berbasis Linux AMIs. Jika Anda menggunakan AMI Linux lain, pastikan Anda menggunakan manajer paket distribusi Anda untuk menginstal klien NFS pada instance. Juga, Anda mungkin perlu menambahkan paket perangkat lunak saat Anda membutuhkannya.

Untuk Amazon Linux HVM AMI, Anda dapat menemukan yang terbaru di IDs [Amazon Linux AMI](#). Anda memilih nilai ID dari IDs tabel AMI Amazon Linux sebagai berikut:

- Pilih wilayah Oregon Barat AS. Panduan ini mengasumsikan Anda membuat semua sumber daya di Wilayah AS Barat (Oregon) (us-west-2).
- Pilih tipe 64-bit HVM yang didukung EBS (karena dalam perintah CLI Anda menentukan jenis `t2.micro` instance, yang tidak mendukung penyimpanan instance).
- ID grup keamanan yang Anda buat untuk instans EC2.
- AWS Region. Panduan ini menggunakan wilayah us-west-2.
- ID subnet VPC Anda di mana Anda ingin meluncurkan instance. Anda bisa mendapatkan daftar subnet menggunakan `describe-subnets` perintah.

```
$ aws ec2 describe-subnets \
--region us-west-2 \
--filters "Name=vpc-id,Values=vpc-id" \
--profile adminuser
```

Setelah Anda memilih ID subnet, tuliskan nilai berikut dari `describe-subnets` hasilnya:

- Subnet ID - Anda memerlukan nilai ini saat membuat target mount. Dalam latihan ini, Anda membuat target mount di subnet yang sama tempat Anda meluncurkan instans EC2.
- Availability Zone dari subnet — Anda memerlukan nilai ini untuk membangun nama DNS target mount Anda, yang Anda gunakan untuk me-mount sistem file pada instans EC2.

2. Jalankan AWS CLI `run-instances` perintah berikut untuk meluncurkan instance EC2.

```
$ aws ec2 run-instances \
--image-id AMI ID \
--count 1 \
--instance-type t2.micro \
--associate-public-ip-address \
--key-name key-pair-name \
--security-group-ids ID of the security group created for EC2 instance \
--subnet-id VPC subnet ID \
--region us-west-2 \
--profile adminuser
```

3. Tuliskan ID instance yang dikembalikan oleh `run-instances` perintah.

4. Instans EC2 yang Anda buat harus memiliki nama DNS publik yang Anda gunakan untuk terhubung ke instans EC2 dan memasang sistem file di atasnya. Nama DNS publik adalah dari bentuk:

```
ec2-xx-xx-xx-xxx.compute-1.amazonaws.com
```

Jalankan perintah CLI berikut dan tuliskan nama DNS publik.

```
aws ec2 describe-instances \  
--instance-ids EC2 instance ID \  
--region us-west-2 \  
--profile adminuser
```

Jika Anda tidak menemukan nama DNS publik, periksa konfigurasi VPC tempat Anda meluncurkan instans EC2. Untuk informasi selengkapnya, lihat [Prasyarat](#).

5. (Opsional) Tetapkan nama ke instans EC2 yang Anda buat. Untuk melakukannya, tambahkan tag dengan nama kunci dan nilai yang disetel ke nama yang ingin Anda tetapkan ke instance. Anda melakukan ini dengan menjalankan AWS CLI `create-tags` perintah berikut.

```
$ aws ec2 create-tags \  
--resources EC2-instance-ID \  
--tags Key=Name,Value=Provide-instance-name \  
--region us-west-2 \  
--profile adminuser
```

Langkah 2: Buat sumber daya EFS

Dalam langkah ini, Anda melakukan hal berikut:

- Buat sistem file EFS terenkripsi.
- Aktifkan manajemen siklus hidup.
- Buat target pemasangan di Availability Zone tempat instans EFS diluncurkan.

Langkah 2.1: Buat sistem file EFS

Pada langkah ini, Anda membuat sistem file EFS. Tuliskan `FileSystemId` yang akan digunakan nanti saat Anda membuat target mount untuk sistem file di langkah berikutnya.

Untuk membuat sistem file

- Buat sistem file dengan Name tag opsional.
 - a. Pada prompt perintah, jalankan perintah AWS CLI `create-file-system` berikut.

```
$ aws efs create-file-system \  
--encrypted \  
--creation-token FileSystemForWalkthrough1 \  
--tags Key=Name,Value=SomeExampleNameValue \  
--region us-west-2 \  
--profile adminuser
```

Anda mendapatkan tanggapan berikut.

```
{  
  "OwnerId": "111122223333",  
  "CreationToken": "FileSystemForWalkthrough1",  
  "FileSystemId": "fs-c657c8bf",  
  "CreationTime": 1548950706.0,  
  "LifecycleState": "creating",  
  "NumberOfMountTargets": 0,  
  "SizeInBytes": {  
    "Value": 0,  
    "ValueInIA": 0,  
    "ValueInStandard": 0  
  },  
  "PerformanceMode": "generalPurpose",  
  "Encrypted": true,  
  "KmsKeyId": "arn:aws:kms:us-west-2:111122223333:a5c11222-7a99-43c8-9dcc-  
abcdef123456",  
  "ThroughputMode": "bursting",  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "SomeExampleNameValue"  
    }  
  ]  
}
```

- b. Perhatikan `FileSystemId` nilainya. Anda memerlukan nilai ini saat Anda membuat target mount untuk sistem file ini di [Langkah 2.3: Buat target mount](#).

Langkah 2.2: Aktifkan manajemen siklus hidup

Pada langkah ini, Anda mengaktifkan manajemen siklus hidup pada sistem file Anda untuk menggunakan kelas penyimpanan EFS Infrequent Access (IA). Untuk mempelajari selengkapnya, lihat [Mengelola siklus hidup penyimpanan](#) dan [Kelas penyimpanan EFS](#).

Untuk mengaktifkan manajemen siklus hidup

- Pada prompt perintah, jalankan AWS CLI `put-lifecycle-configuration` perintah berikut.

```
$ aws efs put-lifecycle-configuration \  
--file-system-id fs-c657c8bf \  
--lifecycle-policies TransitionToIA=AFTER_30_DAYS \  
--region us-west-2 \  
--profile adminuser
```

Anda mendapatkan tanggapan berikut.

```
{  
  "LifecyclePolicies": [  
    {  
      "TransitionToIA": "AFTER_30_DAYS"  
    }  
  ]  
}
```

Langkah 2.3: Buat target mount

Pada langkah ini, Anda membuat target pemasangan untuk sistem file Anda di Availability Zone tempat instans EC2 diluncurkan.

1. Pastikan Anda memiliki informasi berikut:

- ID sistem file (misalnya, `fs-example`) tempat Anda membuat target mount.
- ID subnet VPC tempat Anda meluncurkan instans EC2. [Langkah 1: Buat sumber daya EC2](#)

Untuk tutorial ini, Anda membuat target mount di subnet yang sama di mana Anda meluncurkan instans EC2, jadi Anda memerlukan ID subnet (misalnya, `subnet-example`).

- ID grup keamanan yang Anda buat untuk target pemasangan pada langkah sebelumnya.
2. Pada prompt perintah, jalankan AWS CLI `create-mount-target` perintah berikut.

```
$ aws efs create-mount-target \  
--file-system-id file-system-id \  
--subnet-id subnet-id \  
--security-group ID-of-the security-group-created-for-mount-target \  
--region us-west-2 \  
--profile adminuser
```

Anda mendapatkan tanggapan berikut.

```
{  
  "MountTargetId": "fsmt-example",  
  "NetworkInterfaceId": "eni-example",  
  "FileSystemId": "fs-example",  
  "PerformanceMode" : "generalPurpose",  
  "LifecycleState": "available",  
  "SubnetId": "fs-subnet-example",  
  "OwnerId": "account-id",  
  "IpAddress": "xxx.xx.xx.xxx"  
}
```

3. Anda juga dapat menggunakan `describe-mount-targets` perintah untuk mendapatkan deskripsi target mount yang Anda buat pada sistem file.

```
$ aws efs describe-mount-targets \  
--file-system-id file-system-id \  
--region us-west-2 \  
--profile adminuser
```

Langkah 3: Pasang sistem file pada instans EC2 dan uji

Dalam langkah ini, Anda melakukan hal berikut:

- Kumpulkan informasi yang diperlukan.
- Instal klien NFS pada instans EC2 Anda.
- Pasang sistem file pada instans dan uji EC2 Anda.

Topik

- [Langkah 3.1: Kumpulkan Informasi](#)
- [Langkah 3.2: Instal klien NFS pada instans EC2 Anda](#)
- [Langkah 3.3: Pasang sistem file pada instans EC2 Anda dan uji](#)

Langkah 3.1: Kumpulkan Informasi

Pastikan Anda memiliki informasi berikut saat Anda mengikuti langkah-langkah di bagian ini:

- Nama DNS publik instans EC2 Anda dalam format berikut:

```
ec2-xx-xxx-xxx-xx.aws-region.compute.amazonaws.com
```

- Nama DNS dari sistem file Anda. Anda dapat membuat nama DNS ini menggunakan formulir generik berikut:

```
file-system-id.efs.aws-region.amazonaws.com
```

Instans EC2 tempat Anda memasang sistem file dengan menggunakan target mount dapat menyelesaikan nama DNS sistem file ke alamat IP target mount.

Note

Amazon EFS tidak mengharuskan instans EC2 Anda memiliki alamat IP publik atau nama DNS publik. Persyaratan yang tercantum sebelumnya hanya untuk contoh panduan ini untuk memastikan bahwa Anda dapat terhubung dengan menggunakan SSH ke instance dari luar VPC.

Langkah 3.2: Instal klien NFS pada instans EC2 Anda

Anda dapat terhubung ke instans EC2 dari Windows atau dari komputer yang menjalankan Linux, atau macOS X, atau varian Unix lainnya.

Untuk menginstal klien NFS

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Jalankan perintah berikut pada instans EC2 dengan menggunakan sesi SSH:
 - a. (Opsional) Dapatkan pembaruan dan reboot.

```
$ sudo yum -y update
$ sudo reboot
```

Setelah reboot, sambungkan kembali ke instans EC2 Anda.

- b. Instal klien NFS.

```
$ sudo yum -y install nfs-utils
```

Note

Jika Anda memilih Amazon Linux AMI 2016.03.0 Amazon Linux AMI saat meluncurkan instans EC2 Anda, Anda tidak perlu `nfs-utils` menginstal karena sudah termasuk dalam AMI secara default.

Langkah 3.3: Pasang sistem file pada instans EC2 Anda dan uji

Sekarang Anda me-mount sistem file pada instans EC2 Anda.

1. Buat direktori ("efs-mount-point").

```
$ mkdir ~/efs-mount-point
```

2. Pasang sistem file EFS.

```
$ sudo mount -t nfs -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport mount-
target-DNS:/ ~/efs-mount-point
```


Instans EC2 dapat menyelesaikan nama DNS target mount ke alamat IP. Anda dapat secara opsional menentukan alamat IP dari target mount secara langsung.

```
$ sudo mount -t nfs -o  
nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport mount-  
target-ip:/ ~/efs-mount-point
```

3. Sekarang setelah Anda memiliki sistem file EFS yang terpasang pada instans EC2 Anda, Anda dapat membuat file.

- a. Ubah direktori.

```
$ cd ~/efs-mount-point
```

- b. Daftar isi direktori.

```
$ ls -al
```

Itu harus kosong.

```
drwxr-xr-x 2 root      root      4096 Dec 29 22:33 .  
drwx----- 4 ec2-user ec2-user 4096 Dec 29 22:54 ..
```

- c. Direktori root dari sistem file, setelah pembuatan, dimiliki oleh dan dapat ditulis oleh pengguna root, jadi Anda perlu mengubah izin untuk menambahkan file.

```
$ sudo chmod go+rw .
```

Sekarang, jika Anda mencoba `ls -al` perintah, Anda melihat bahwa izin telah berubah.

```
drwxrwxrwx 2 root      root      4096 Dec 29 22:33 .  
drwx----- 4 ec2-user ec2-user 4096 Dec 29 22:54 ..
```

- d. Buat file teks.

```
$ touch test-file.txt
```

- e. Daftar konten direktori.

```
$ ls -l
```

Anda sekarang telah berhasil membuat dan memasang sistem file EFS pada instans EC2 Anda di VPC Anda.

Sistem file yang Anda pasang tidak bertahan selama reboot. Untuk secara otomatis memasang kembali direktori, Anda dapat menggunakan file tersebut. `fstab` Jika Anda menggunakan grup Auto Scaling untuk meluncurkan instans EC2, Anda juga dapat mengatur skrip dalam konfigurasi peluncuran.

Langkah 4: Membersihkan

Jika Anda tidak lagi membutuhkan sumber daya yang Anda buat, Anda harus menghapusnya. Anda dapat melakukan ini dengan CLI.

- Hapus sumber daya EC2 (instans EC2 dan dua grup keamanan). Amazon EFS menghapus antarmuka jaringan saat Anda menghapus target pemasangan.
- Hapus sumber daya EFS (sistem file, target pemasangan).

Untuk menghapus AWS sumber daya yang dibuat dalam panduan ini

1. Hentikan instans EC2 yang Anda buat untuk tutorial ini.

```
$ aws ec2 terminate-instances \
--instance-ids instance-id \
--profile adminuser
```

Anda juga dapat menghapus sumber daya EC2 menggunakan konsol. Untuk petunjuk, lihat [Mengakhiri instance](#).

2. Hapus target pemasangan.

Anda harus menghapus target mount yang dibuat untuk sistem file sebelum menghapus sistem file. Anda bisa mendapatkan daftar target mount dengan menggunakan perintah `describe-mount-targets` CLI.

```
$ aws efs describe-mount-targets \
--file-system-id file-system-ID \
```

```
--profile adminuser \  
--region aws-region
```

Kemudian hapus target mount dengan menggunakan perintah `delete-mount-target` CLI.

```
$ aws efs delete-mount-target \  
--mount-target-id ID-of-mount-target-to-delete \  
--profile adminuser \  
--region aws-region
```

3. (Opsional) Hapus dua grup keamanan yang Anda buat. Anda tidak membayar untuk membuat grup keamanan.

Anda harus menghapus grup keamanan target mount terlebih dahulu, sebelum menghapus grup keamanan instans EC2. Grup keamanan target mount memiliki aturan yang mereferensikan grup keamanan EC2. Oleh karena itu, Anda tidak dapat menghapus grup keamanan instans EC2 terlebih dahulu.

Untuk petunjuk, lihat [Menghapus grup keamanan Anda](#) di Panduan Pengguna Amazon EC2.

4. Hapus sistem file dengan menggunakan perintah `delete-file-system` CLI. Anda bisa mendapatkan daftar sistem file Anda dengan menggunakan perintah `describe-file-systems` CLI. Anda bisa mendapatkan ID sistem file dari respons.

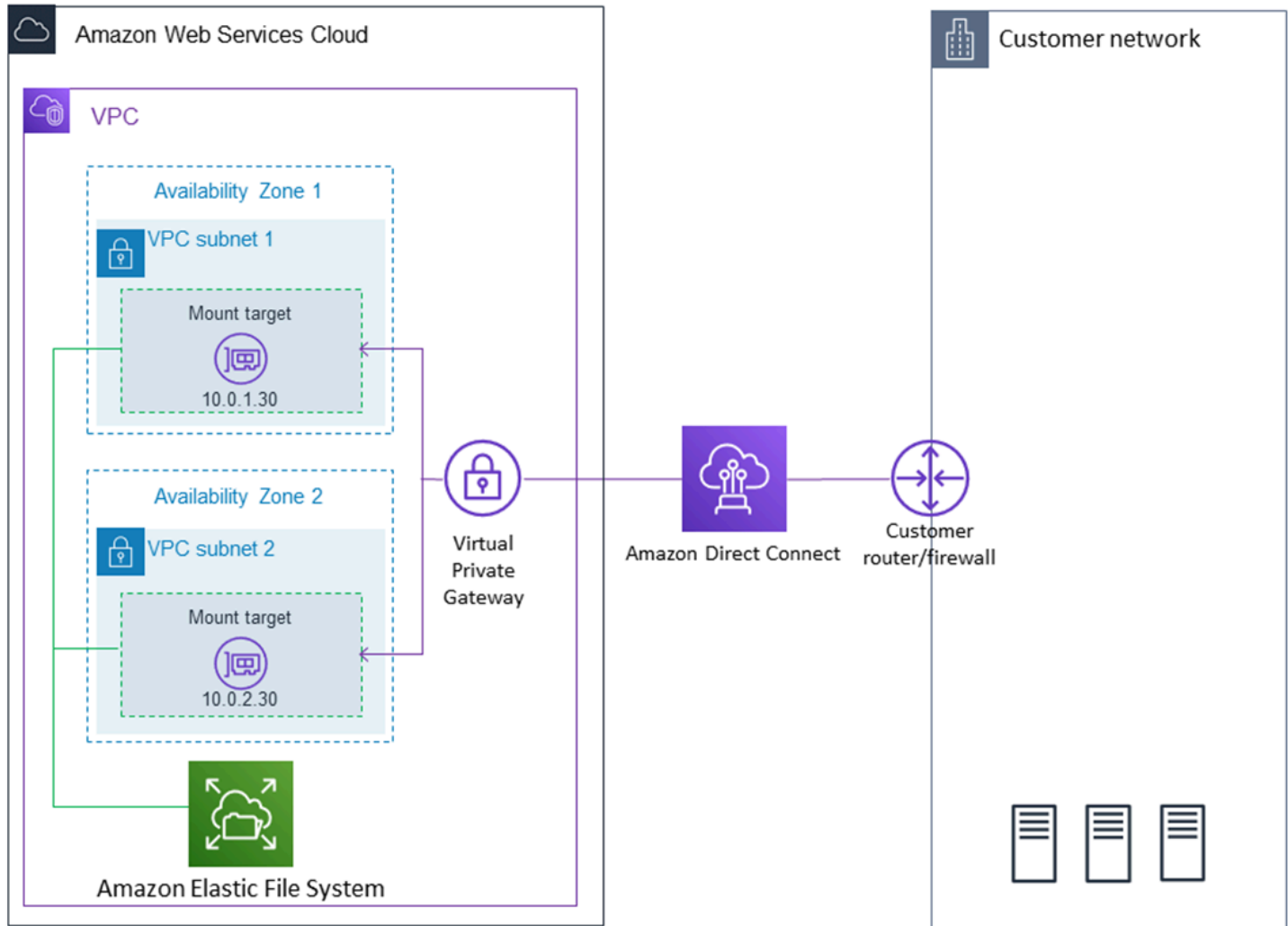
```
aws efs describe-file-systems \  
--profile adminuser \  
--region aws-region
```

Hapus sistem file dengan memberikan ID sistem file.

```
$ aws efs delete-file-system \  
--file-system-id ID-of-file-system-to-delete \  
--region aws-region \  
--profile adminuser
```

Tutorial: Pemasangan dengan klien Linux lokal

Anda dapat memasang sistem file EFS di server pusat data lokal saat tersambung ke Amazon VPC Direct Connect dengan atau VPN. Grafik berikut menunjukkan diagram skematik tingkat tinggi yang Layanan AWS diperlukan dalam pemasangan sistem file EFS dari lokal.



Note

Menggunakan Amazon EFS dengan klien berbasis Microsoft Windows tidak didukung.

Topik

- [Prasyarat](#)
- [Langkah 1: Buat sumber daya EFS Anda](#)

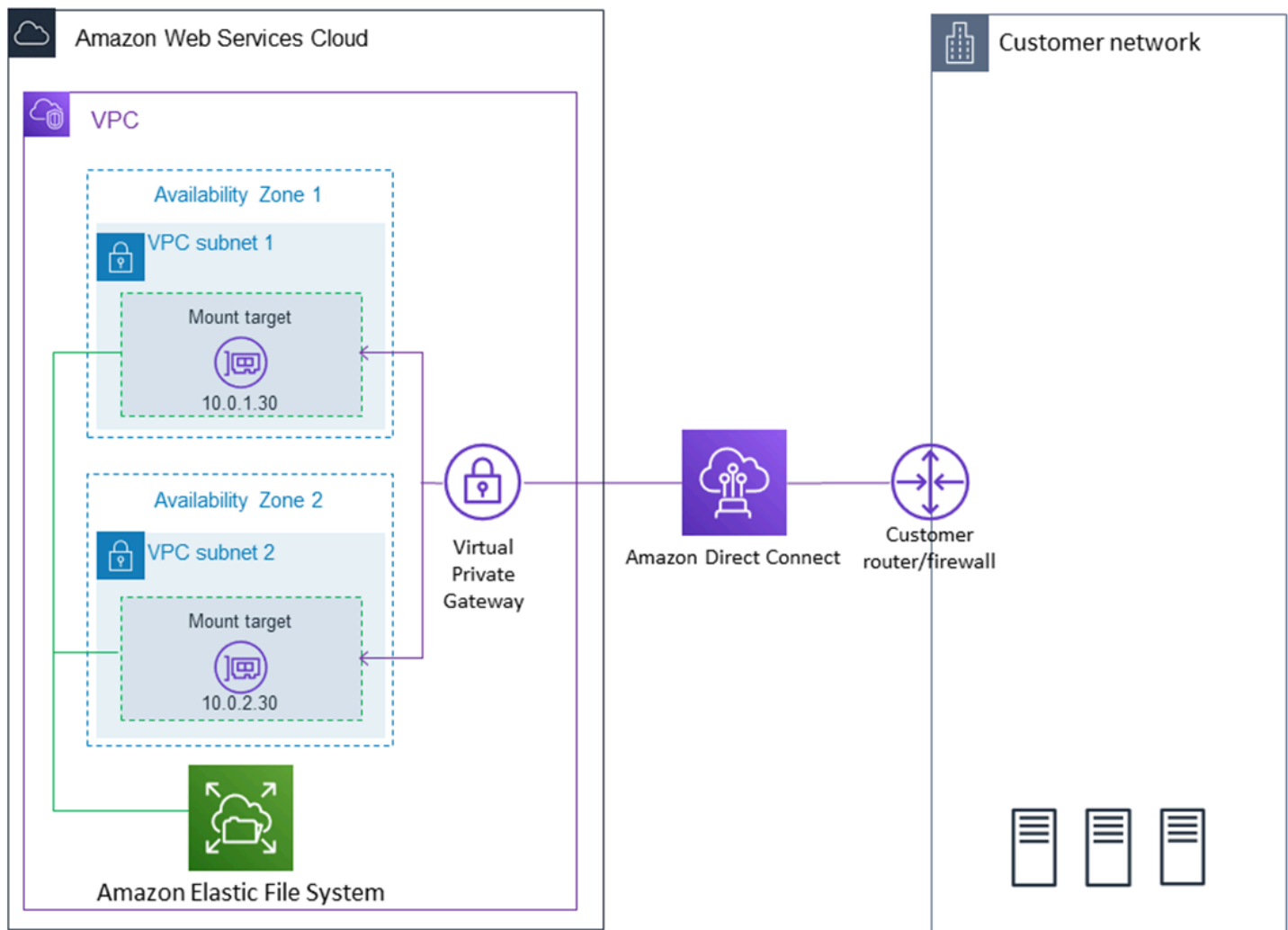
- [Langkah 2: Instal klien NFS](#)
- [Langkah 3: Pasang sistem file EFS di klien lokal](#)
- [Langkah 4: Bersihkan sumber daya dan lindungi AWS akun Anda](#)
- [Opsional: Mengenkripsi data dalam perjalanan](#)

Prasyarat

Pastikan Anda sudah memiliki koneksi Direct Connect atau VPN. Untuk informasi selengkapnya tentang Direct Connect, lihat [Direct Connect Panduan Pengguna](#). Untuk informasi selengkapnya tentang pengaturan koneksi VPN, lihat [Menyambungkan VPC Anda ke jaringan jarak jauh menggunakan Panduan AWS Virtual Private Network](#) Pengguna Amazon VPC.

Setelah Anda memiliki koneksi Direct Connect atau VPN, buat sistem file EFS dan target pemasangan di Amazon VPC Anda. Setelah itu, Anda mengunduh dan menginstal amazon-efs-utils alat. Kemudian, Anda menguji sistem file dari klien lokal Anda. Akhirnya, langkah pembersihan di akhir penelusuran memberikan informasi bagi Anda untuk menghapus sumber daya ini.

Panduan ini menciptakan semua sumber daya ini di Wilayah Barat AS (Oregon) (). us-west-2 Apapun yang AWS Region Anda gunakan, pastikan untuk menggunakannya secara konsisten. Semua sumber daya Anda—VPC, target pemasangan, dan sistem file EFS Anda—harus sama, seperti yang ditunjukkan pada diagram berikut AWS Region.



Note

Dalam beberapa kasus, aplikasi lokal Anda mungkin perlu mengetahui apakah sistem file EFS tersedia. Dalam kasus ini, aplikasi Anda harus dapat menunjuk ke alamat IP mount point yang berbeda jika titik pemasangan pertama menjadi tidak tersedia untuk sementara. Dalam skenario ini, kami menyarankan agar Anda memiliki dua klien lokal yang terhubung ke sistem file Anda melalui Availability Zones (AZs) yang berbeda untuk ketersediaan yang lebih tinggi.

Anda dapat menggunakan kredensial root Anda Akun AWS untuk masuk ke konsol dan mencoba latihan ini. Namun, praktik terbaik AWS Identity and Access Management (IAM) menyarankan agar Anda tidak menggunakan kredensial root Anda. Akun AWS Sebagai gantinya, buat pengguna administrator di akun Anda dan gunakan kredensial tersebut untuk mengelola sumber daya di akun

Anda. Untuk informasi selengkapnya, lihat [Akses masuk tunggal Akun AWS](#) di AWS IAM Identity Center Panduan Pengguna.

Anda dapat menggunakan VPC default atau VPC khusus yang telah Anda buat di akun Anda. Untuk panduan ini, konfigurasi VPC default berfungsi. Namun, jika Anda menggunakan VPC kustom, verifikasi hal berikut:

- Gateway internet terpasang ke VPC Anda. Untuk informasi selengkapnya, lihat [Mengaktifkan akses internet untuk VPC menggunakan gateway internet di Panduan](#) Pengguna Amazon VPC.
- Tabel rute VPC mencakup aturan untuk mengirim semua lalu lintas internet ke gateway Internet.

Langkah 1: Buat sumber daya EFS Anda

Pada langkah ini, Anda membuat sistem file EFS dan memasang target.

Untuk membuat sistem file EFS Anda

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Pilih Buat Sistem File.
3. Pilih VPC default Anda dari daftar VPC.
4. Pilih kotak centang untuk semua Availability Zones. Pastikan bahwa mereka semua memiliki subnet default, alamat IP otomatis, dan grup keamanan default yang dipilih. Ini adalah target mount Anda. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).
5. Pilih Langkah Selanjutnya.
6. Beri nama sistem file Anda, tetap pilih tujuan umum sebagai mode kinerja default Anda, dan pilih Langkah Berikutnya.
7. Pilih Buat Sistem File.
8. Pilih sistem file Anda dari daftar dan catat nilai grup Keamanan. Anda memerlukan nilai ini untuk langkah berikutnya.

Sistem file yang baru saja Anda buat memiliki target mount. Setiap target mount memiliki grup keamanan terkait. Grup keamanan bertindak sebagai firewall virtual yang mengontrol lalu lintas jaringan. Jika Anda tidak menyediakan grup keamanan saat membuat target pemasangan, Amazon EFS mengaitkan grup keamanan default VPC dengannya. Jika Anda mengikuti langkah-langkah sebelumnya dengan tepat, maka target mount Anda menggunakan grup keamanan default.

Selanjutnya, Anda menambahkan aturan ke grup keamanan target mount untuk mengizinkan lalu lintas masuk ke port Network File System (NFS) (2049). Anda dapat menggunakan Konsol Manajemen AWS untuk menambahkan aturan ke grup keamanan target mount Anda di VPC Anda.

Untuk memungkinkan lalu lintas masuk ke port NFS

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Di bawah NETWORK & SECURITY, pilih Grup Keamanan.
3. Pilih grup keamanan yang terkait dengan sistem file Anda. Anda membuat catatan tentang ini di akhir [Langkah 1: Buat sumber daya EFS Anda](#).
4. Di panel tab yang muncul di bawah daftar grup keamanan, pilih tab Masuk.
5. Pilih Edit.
6. Pilih Tambah Aturan, dan pilih aturan dari jenis berikut:
 - Jenis - NFS
 - Sumber - Di mana saja

Kami menyarankan Anda hanya menggunakan sumber Anywhere untuk pengujian. Anda dapat membuat sumber kustom yang disetel ke alamat IP klien lokal, atau menggunakan konsol dari klien itu sendiri, dan memilih IP Saya.

Note

Anda tidak perlu menambahkan aturan keluar, karena aturan keluar default memungkinkan semua lalu lintas untuk pergi. Jika Anda tidak memiliki aturan keluar default ini, tambahkan aturan keluar untuk membuka koneksi TCP pada port NFS 2049, mengidentifikasi grup keamanan target mount sebagai tujuan.

Langkah 2: Instal klien NFS

Pada langkah ini, Anda menginstal klien NFS.

Untuk menginstal klien NFS di server lokal

Note

Jika Anda memerlukan data untuk dienkripsi saat transit, gunakan penolong pemasangan Amazon EFS `amazon-efs-utils`, bukan klien NFS. Untuk informasi tentang penginstalan `amazon-efs-utils`, lihat bagian Opsional: Mengenkripsi Data dalam Transit.

1. Akses terminal untuk klien lokal Anda.
2. Instal NFS.

Jika Anda menggunakan Red Hat Linux, instal NFS dengan perintah berikut.

```
$ sudo yum -y install nfs-utils
```

Jika Anda menggunakan Ubuntu, instal NFS dengan perintah berikut.

```
$ sudo apt-get -y install nfs-common
```

Langkah 3: Pasang sistem file EFS di klien lokal

Untuk membuat direktori mount

1. Buatlah sebuah direktori untuk titik pemasangan dengan perintah berikut ini.

Example

```
mkdir ~/efs
```

2. Pilih alamat IP pilihan Anda dari target pemasangan di Availability Zone. Anda dapat mengukur latensi dari klien Linux lokal Anda. Untuk melakukannya, gunakan alat berbasis terminal seperti `ping` terhadap alamat IP instans EC2 Anda di Availability Zone yang berbeda untuk menemukan yang memiliki latensi terendah.
- Jalankan perintah `mount` untuk me-mount sistem file menggunakan alamat IP dari target mount.

```
$ sudo mount -t nfs -o  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport mount-  
target-IP:/ ~/efs
```

Sekarang setelah Anda memasang sistem file EFS Anda, Anda dapat mengujinya dengan prosedur berikut.

Untuk menguji koneksi sistem file EFS

1. Ubah direktori ke direktori baru yang Anda buat dengan perintah berikut.

```
$ cd ~/efs
```

2. Buat subdirektori dan ubah kepemilikan subdirektori tersebut ke pengguna instans EC2 Anda. Kemudian, arahkan ke direktori baru itu dengan perintah berikut.

```
$ sudo mkdir getting-started  
$ sudo chown ec2-user getting-started  
$ cd getting-started
```

3. Buat file teks dengan perintah berikut.

```
$ touch test-file.txt
```

4. Daftarkan isi direktori dengan perintah berikut.

```
$ ls -al
```

Akibatnya, file berikut dibuat.

```
-rw-rw-r-- 1 username username 0 Nov 15 15:32 test-file.txt
```

Warning

Gunakan opsi `_netdev`, yang digunakan untuk mengidentifikasi sistem file jaringan, ketika memasang sistem file Anda secara otomatis. Jika `_netdev` hilang, instans EC2 Anda mungkin berhenti merespons. Hasil ini didapatkan karena sistem file jaringan perlu

diinisialisasi setelah instans komputasi memulai jaringannya. Untuk informasi selengkapnya, lihat [Pemasangan otomatis gagal dan instans tidak responsif](#).

Langkah 4: Bersihkan sumber daya dan lindungi AWS akun Anda

Setelah Anda menyelesaikan panduan ini, atau jika Anda tidak ingin menjelajahi penelusuran, Anda harus mengikuti langkah-langkah ini untuk membersihkan sumber daya Anda dan melindungi akun Anda. AWS

Untuk membersihkan sumber daya dan melindungi Akun AWS

1. Lepaskan sistem file EFS dengan perintah berikut.

```
$ sudo umount ~/efs
```

2. Buka konsol Amazon EFS di [Amazon EFS Console](#).
3. Pilih sistem file EFS yang ingin Anda hapus dari daftar sistem file.
4. Untuk Tindakan, pilih Hapus sistem file.
5. Di kotak dialog Hapus sistem file secara permanen, ketik ID sistem file untuk sistem file EFS yang ingin Anda hapus, lalu pilih Hapus Sistem File.
6. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
7. Pada panel navigasi, pilih Grup Keamanan.
8. Pilih nama grup keamanan tempat Anda menambahkan aturan untuk panduan ini.

Warning

Jangan hapus grup keamanan default untuk VPC Anda.

9. Untuk Tindakan, pilih Edit aturan inbound.
10. Pilih X di akhir aturan masuk yang Anda tambahkan, dan pilih Simpan.

Opsional: Mengenkripsi data dalam perjalanan

Untuk mengenkripsi data dalam perjalanan, gunakan helper mount Amazon EFS `amazon-efs-utils`, bukan klien NFS.

amazon-efs-utils Paket ini adalah koleksi sumber terbuka alat Amazon EFS. amazon-efs-utilsKoleksi ini dilengkapi dengan mount helper dan tooling yang membuatnya lebih mudah untuk mengenkripsi data dalam perjalanan untuk Amazon EFS. Untuk informasi lebih lanjut tentang paket ini, lihat[Menginstal klien Amazon EFS](#). Paket ini tersedia sebagai unduhan gratis GitHub, yang bisa Anda dapatkan dengan mengkloning repositori paket.

Untuk mengkloning dari amazon-efs-utils GitHub

1. Akses terminal untuk klien lokal Anda.
2. Dari terminal, kloning amazon-efs-utils alat dari GitHub ke direktori pilihan Anda, dengan perintah berikut.

```
git clone https://github.com/aws/efs-utils
```

Sekarang setelah Anda memiliki paket, Anda dapat menginstalnya. Instalasi ini ditangani secara berbeda tergantung pada distribusi Linux klien lokal Anda. Distribusi berikut didukung:

- Amazon Linux 2
- Amazon Linux
- Red Hat Enterprise Linux (dan turunannya seperti CentOS) versi 8 dan yang lebih baru
- Ubuntu 16.04 LTS dan yang lebih baru

Untuk membangun dan menginstal amazon-efs-utils sebagai paket RPM

1. Buka terminal pada klien Anda dan arahkan ke direktori yang memiliki amazon-efs-utils paket kloning dari GitHub.
2. Bangun paket dengan perintah berikut.

```
make rpm
```

Note

Jika Anda belum melakukannya, instal paket rpm-builder dengan perintah berikut.

```
sudo yum -y install rpm-build
```

3. Instal paket dengan perintah berikut.

```
sudo yum -y install build/amazon-efs-utils*rpm
```

Untuk membangun dan menginstal amazon-efs-utils sebagai paket deb

1. Buka terminal pada klien Anda dan arahkan ke direktori yang memiliki amazon-efs-utils paket kloning dari GitHub.
2. Bangun paket dengan perintah berikut.

```
./build-deb.sh
```

3. Instal paket dengan perintah berikut.

```
sudo apt-get install build/amazon-efs-utils*deb
```

Setelah paket diinstal, konfigurasikan amazon-efs-utils untuk digunakan di AWS Region with Direct Connect atau VPN Anda.

Untuk mengkonfigurasi amazon-efs-utils untuk digunakan di AWS Region

1. Akses terminal untuk instans EC2 Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Menggunakan editor teks pilihan Anda, buka `/etc/amazon/efs/efs-utils.conf` file.
3. Temukan garisnya `dns_name_format = {fs_id}.efs.{region}.amazonaws.com`.
4. Ubah `{region}` dengan ID untuk AWS Wilayah Anda, misalnya `us-west-2`.

Untuk memasang sistem file EFS di klien lokal Anda, pertama-tama buka terminal di klien Linux lokal Anda. Untuk me-mount sistem, Anda memerlukan ID sistem file, alamat IP target mount untuk salah satu target mount Anda, dan sistem file AWS Region. Jika Anda membuat beberapa target mount untuk sistem file Anda, maka Anda dapat memilih salah satu dari ini.

Ketika Anda memiliki informasi itu, Anda dapat me-mount sistem file Anda dalam tiga langkah:

Untuk membuat direktori mount

1. Buatlah sebuah direktori untuk titik pemasangan dengan perintah berikut ini.

Example

```
mkdir ~/efs
```

2. Pilih alamat IP pilihan Anda dari target pemasangan di Availability Zone. Anda dapat mengukur latensi dari klien Linux lokal Anda. Untuk melakukannya, gunakan alat berbasis terminal seperti ping terhadap alamat IP instans EC2 Anda di Availability Zone yang berbeda untuk menemukan yang memiliki latensi terendah.

Untuk memperbarui **/etc/hosts**

- Tambahkan entri ke **/etc/hosts** file lokal Anda dengan ID sistem file dan alamat IP target mount, dalam format berikut.

```
mount-target-IP-Address file-system-ID.efs.region.amazonaws.com
```

Example

```
192.0.2.0 fs-12345678.efs.us-west-2.amazonaws.com
```

Untuk membuat direktori mount

1. Buatlah sebuah direktori untuk titik pemasangan dengan perintah berikut ini.

Example

```
mkdir ~/efs
```

2. Jalankan perintah mount untuk me-mount sistem file.

Example

```
sudo mount -t efs fs-12345678 ~/efs
```

Jika Anda ingin menggunakan enkripsi data dalam perjalanan, perintah mount Anda terlihat seperti berikut ini.

Example

```
sudo mount -t efs -o tls fs-12345678 ~/efs
```

Tutorial: Pasang sistem file dari VPC yang berbeda

Dalam tutorial ini, Anda menyiapkan instans EC2 untuk memasang sistem file EFS yang ada di cloud pribadi virtual (VPC) yang berbeda. Anda melakukan ini menggunakan EFS mount helper. Mount helper adalah bagian dari `amazon-efs-utils` seperangkat alat. Untuk informasi selengkapnya tentang `amazon-efs-utils`, lihat [Menginstal klien Amazon EFS](#).

VPC klien dan VPC sistem file EFS Anda harus terhubung menggunakan koneksi peering VPC atau gateway transit VPC. Saat Anda menggunakan koneksi peering VPC atau gateway transit untuk menghubungkan VPC, instans EC2 yang ada di satu VPC dapat mengakses sistem file EFS di VPC lain, meskipun milik akun yang berbeda. VPCs

Note

Menggunakan Amazon EFS dengan klien berbasis Microsoft Windows tidak didukung.

Topik

- [Prasyarat](#)
- [Langkah 1: Tentukan ID Availability Zone target mount](#)
- [Langkah 2: Tentukan alamat IP target mount](#)
- [Langkah 3: Tambahkan entri host untuk target pemasangan](#)
- [Langkah 4: Pasang sistem file Anda menggunakan EFS mount helper](#)
- [Langkah 5: Bersihkan sumber daya dan lindungi Anda Akun AWS](#)

Prasyarat

Untuk menyelesaikan tutorial ini, Anda harus memiliki yang berikut:

- amazon-efs-utils Set alat diinstal pada instans EC2 sebelum menggunakan prosedur ini. Untuk petunjuk tentang menginstal amazon-efs-utils, lihat [Menginstal klien Amazon EFS](#).
- Salah satu dari yang berikut:
 - Koneksi peering VPC antara VPC tempat sistem file EFS berada dan VPC tempat instans EC2 berada. Koneksi peering VPC adalah koneksi jaringan antara dua VPCs. Jenis koneksi ini memungkinkan Anda untuk merutekan lalu lintas di antara mereka menggunakan alamat Internet Protocol pribadi versi 4 (IPv4) atau Internet Protocol versi 6 (IPv6). Anda dapat menggunakan VPC peering untuk terhubung VPCs dalam hal yang sama AWS Region atau di antaranya. Wilayah AWS Untuk informasi selengkapnya, lihat [Membuat dan Menerima Koneksi Peering VPC di Panduan Peering VPC Amazon](#).
 - Gateway transit yang menghubungkan VPC tempat sistem file EFS berada dan VPC tempat instans EC2 berada. Gateway transit adalah hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. Untuk informasi selengkapnya, lihat [Memulai menggunakan Gateway Transit VPC Amazon di Panduan Gateway Transit VPC Amazon](#).

Langkah 1: Tentukan ID Availability Zone target mount

Untuk memastikan ketersediaan sistem file Anda yang tinggi, kami sarankan Anda selalu menggunakan alamat IP target pemasangan EC2 yang berada di Availability Zone yang sama dengan klien NFS Anda. Jika Anda memasang sistem file EFS yang ada di akun lain, pastikan klien NFS dan target pemasangan EFS berada dalam ID Availability Zone yang sama. Persyaratan ini berlaku karena nama Availability Zone dapat berbeda antar akun.

Untuk menentukan ID Availability Zone dari instans EC2

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke instans EC2 Anda](#) di Panduan Pengguna Amazon EC2.
2. Tentukan ID Availability Zone bahwa instans EC2 menggunakan perintah `describe-availability-zones` CLI sebagai berikut.

```
[ec2-user@ip-10.0.0.1] $ aws ec2 describe-availability-zones --zone-name
{
  "AvailabilityZones": [
    {
      "State": "available",
      "ZoneName": "us-east-2b",
      "Messages": [],
```



```

        "ZoneId": "use2-az2",
        "RegionName": "us-east-2"
    }
]
}

```

ID Availability Zone dikembalikan di ZoneId properti `use2-az2`.

Langkah 2: Tentukan alamat IP target mount

Sekarang setelah Anda mengetahui ID Availability Zone dari instans EC2, Anda sekarang dapat mengambil alamat IP dari target mount yang ada di ID Availability Zone yang sama.

Untuk menentukan alamat IP target mount di ID Availability Zone yang sama

- Ambil alamat IP target mount untuk sistem file Anda di ID `use2-az2` AZ menggunakan perintah `describe-mount-targets` CLI, sebagai berikut.

```

$ aws efs describe-mount-targets --file-system-id file_system_id
{
  "MountTargets": [
    {
      "OwnerId": "111122223333",
      "MountTargetId": "fsmt-11223344",
      =====> "AvailabilityZoneId": "use2-az2",
      "NetworkInterfaceId": "eni-048c09a306023eeec",
      "AvailabilityZoneName": "us-east-2b",
      "FileSystemId": "fs-01234567",
      "LifecycleState": "available",
      "SubnetId": "subnet-06eb0da37ee82a64f",
      "OwnerId": "958322738406",
      =====> "IpAddress": "10.0.2.153"
    },
    ...
    {
      "OwnerId": "111122223333",
      "MountTargetId": "fsmt-667788aa",
      "AvailabilityZoneId": "use2-az3",
      "NetworkInterfaceId": "eni-0edb579d21ed39261",
      "AvailabilityZoneName": "us-east-2c",
      "FileSystemId": "fs-01234567",
      "LifecycleState": "available",

```

```
        "SubnetId": "subnet-0ee85556822c441af",  
        "OwnerId": "958322738406",  
        "IpAddress": "10.0.3.107"  
    }  
]  
}
```

Target pemasangan di ID use2-az2 Availability Zone memiliki alamat IP 10.0.2.153.

Langkah 3: Tambahkan entri host untuk target pemasangan

Anda sekarang dapat membuat entri dalam `/etc/hosts` file pada instans EC2 yang memetakan alamat IP target mount ke nama host sistem file EFS Anda.

Untuk menambahkan entri host untuk target pemasangan

1. Tambahkan baris untuk alamat IP target mount ke `/etc/hosts` file instans EC2. Entri menggunakan format `mount-target-IP-Address file-system-ID.efs.region.amazonaws.com`. Gunakan perintah berikut untuk menambahkan baris ke file.

```
echo "10.0.2.153 fs-01234567.efs.us-east-2.amazonaws.com" | sudo tee -a /etc/hosts
```

2. Pastikan bahwa grup keamanan VPC untuk instans EC2 dan target mount memiliki aturan yang memungkinkan akses ke sistem file EFS, sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Menggunakan grup keamanan VPC](#).

Langkah 4: Pasang sistem file Anda menggunakan EFS mount helper

Untuk me-mount sistem file EFS Anda, pertama-tama Anda membuat direktori mount pada instans EC2. Kemudian, menggunakan EFS mount helper, Anda dapat me-mount sistem file dengan otorisasi AWS Identity and Access Management (IAM) atau titik akses EFS. Untuk informasi selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#) dan [Bekerja dengan titik akses](#).

Untuk membuat direktori mount

- Buat direktori untuk memasang sistem file menggunakan perintah berikut.

```
$ sudo mkdir /mnt/efs/
```

Untuk me-mount sistem file menggunakan otorisasi IAM

- Gunakan perintah berikut untuk me-mount sistem file menggunakan otorisasi IAM.

```
$ sudo mount -t efs -o tls,iam file-system-id /mnt/efs/
```

Untuk me-mount sistem file menggunakan titik akses EFS

- Gunakan perintah berikut untuk me-mount sistem file menggunakan titik akses EFS.

```
$ sudo mount -t efs -o tls,accesspoint=access-point-id file-system-id /mnt/efs/
```

Sekarang setelah Anda memasang sistem file EFS Anda, Anda dapat mengujinya dengan prosedur berikut.

Untuk menguji koneksi sistem file EFS

1. Ubah direktori ke direktori baru yang Anda buat dengan perintah berikut.

```
$ cd ~/mnt/efs
```

2. Buat subdirektori dan ubah kepemilikan subdirektori tersebut ke pengguna instans EC2 Anda. Kemudian arahkan ke direktori baru itu dengan perintah berikut.

```
$ sudo mkdir getting-started  
$ sudo chown ec2-user getting-started  
$ cd getting-started
```

3. Buat file teks dengan perintah berikut.

```
$ touch test-file.txt
```

4. Daftar isi direktori dengan perintah berikut.

```
$ ls -al
```

Akibatnya, file berikut dibuat.

```
-rw-rw-r-- 1 username username 0 Nov 15 15:32 test-file.txt
```

Anda juga dapat me-mount sistem file Anda secara otomatis dengan menambahkan entri ke `/etc/fstab` file. Untuk informasi selengkapnya, lihat [Mengaktifkan pemasangan otomatis pada instans EC2 Linux yang ada](#).

Warning

Gunakan opsi `_netdev`, yang digunakan untuk mengidentifikasi sistem file jaringan, ketika memasang sistem file Anda secara otomatis. Jika `_netdev` hilang, instans EC2 Anda mungkin berhenti merespons. Hasil ini didapatkan karena sistem file jaringan perlu diinisialisasi setelah instans komputasi memulai jaringannya. Untuk informasi selengkapnya, lihat [Pemasangan otomatis gagal dan instans tidak responsif](#).

Langkah 5: Bersihkan sumber daya dan lindungi Anda Akun AWS

Setelah Anda menyelesaikan tutorial ini, lakukan langkah-langkah berikut untuk membersihkan sumber daya Anda dan melindungi sumber daya Anda Akun AWS.

Untuk membersihkan sumber daya dan melindungi Anda Akun AWS

1. Lepaskan sistem file EFS dengan perintah berikut.

```
$ sudo umount ~/efs
```

2. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
3. Pilih sistem file EFS yang ingin Anda hapus dari daftar sistem file.
4. Untuk Tindakan, pilih Hapus sistem file.
5. Di kotak dialog Hapus sistem file secara permanen, ketik ID sistem file untuk sistem file EFS yang ingin Anda hapus, lalu pilih Hapus Sistem File.
6. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>

7. Pada panel navigasi, pilih Grup Keamanan.
8. Pilih nama grup keamanan yang Anda tambahkan aturan untuk tutorial ini.

**Warning**

Jangan hapus grup keamanan default untuk VPC Anda.

9. Untuk Tindakan, pilih Edit aturan inbound.
10. Pilih X di akhir aturan masuk yang Anda tambahkan, dan pilih Simpan.

Memecahkan masalah pemasangan

Berikut ini, Anda dapat menemukan informasi tentang pemecahan masalah pemasangan sistem file EFS.

Pemasangan sistem file pada instance Windows gagal

Pemasangan sistem file pada instans Amazon EC2 di Microsoft Windows gagal.

Tindakan yang harus diambil

Jangan gunakan Amazon EFS dengan instans Windows EC2, yang tidak didukung.

Akses ditolak oleh server

Pemasangan sistem file gagal dengan pesan berikut:

```
/efs mount.nfs4: access denied by server while mounting 127.0.0.1:/
```

Masalah ini dapat terjadi jika klien NFS Anda tidak memiliki izin untuk memasang sistem file.

Tindakan yang harus diambil

Jika Anda mencoba memasang sistem file menggunakan IAM, pastikan Anda menggunakan opsi `-o iam` or `-o tls` di perintah mount Anda. Ini memberi tahu EFS mount helper untuk meneruskan kredensial Anda ke target pemasangan EFS. Jika Anda masih belum memiliki akses, periksa kebijakan sistem file dan kebijakan identitas Anda untuk memastikan tidak ada klausa DENY yang berlaku untuk koneksi Anda, dan setidaknya ada satu klausa ALLOW yang berlaku untuk koneksi.

Untuk informasi selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#) dan [Membuat kebijakan sistem file](#).

Pemasangan otomatis gagal dan instans tidak responsif

Masalah ini dapat terjadi jika sistem file dipasang secara otomatis pada sebuah instance dan `_netdev` opsi tidak dideklarasikan. Jika `_netdev` hilang, instans EC2 Anda mungkin berhenti merespons. Hasil ini didapatkan karena sistem file jaringan perlu diinisialisasi setelah instans komputasi memulai jaringannya.

Tindakan yang harus diambil

Jika masalah ini terjadi, hubungi AWS Support.

Pemasangan beberapa sistem file Amazon EFS di `/etc/fstab` gagal

Untuk contoh yang menggunakan sistem init `systemd` dengan dua atau lebih entri Amazon EFS di `/etc/fstab`, mungkin ada saat-saat di mana beberapa atau semua entri ini tidak dipasang. Dalam hal ini, `dmesg` output menunjukkan satu atau lebih baris yang mirip dengan yang berikut ini.

```
NFS: nfs4_discover_server_trunking unhandled error -512. Exiting with error EIO
```

Tindakan yang harus diambil

Dalam hal ini, kami menyarankan Anda membuat file layanan `systemd` baru di `/etc/systemd/system/mount-nfs-sequentially.service`. Kode yang akan disertakan dalam file tergantung pada apakah Anda memasang sistem file secara manual atau menggunakan helper mount Amazon EFS.

- Jika Anda memasang sistem file secara manual, maka `ExecStart` perintah harus menunjuk ke Network File System (NFS4). Sertakan kode berikut dalam file:

```
[Unit]
Description=Workaround for mounting NFS file systems sequentially at boot time
After=remote-fs.target

[Service]
Type=oneshot
ExecStart=/bin/mount -avt nfs4
RemainAfterExit=yes
```

```
[Install]
WantedBy=multi-user.target
```

- Jika Anda menggunakan helper mount Amazon EFS, maka ExecStart perintah harus menunjuk ke EFS alih-alih NFS4 menggunakan Transport Layer Security (TLS). Sertakan kode berikut dalam file:

```
[Unit]
Description=Workaround for mounting NFS file systems sequentially at boot time
After=remote-fs.target

[Service]
Type=oneshot
ExecStart=/bin/mount -avt efs
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Setelah Anda membuat file, jalankan dua perintah berikut:

1. `sudo systemctl daemon-reload`
2. `sudo systemctl enable mount-nfs-sequentially.service`

Kemudian mulai ulang instans Amazon EC2 Anda. Sistem file dipasang sesuai permintaan, umumnya dalam satu detik.

Perintah mount gagal dengan pesan kesalahan “jenis fs yang salah”

Perintah mount gagal dengan pesan kesalahan berikut.

```
mount: wrong fs type, bad option, bad superblock on 10.1.25.30:/,
missing codepage or helper program, or other error (for several filesystems
(e.g. nfs, cifs) you might need a /sbin/mount.<type> helper program)
In some cases useful info is found in syslog - try dmesg | tail or so.
```

Tindakan yang harus diambil

Jika Anda menerima pesan ini, instal paket `nfs-utils` (atau `nfs-common` di Ubuntu). Untuk informasi selengkapnya, lihat [Menginstal klien NFS](#).

Perintah mount gagal dengan pesan kesalahan “opsi pemasangan yang salah”

Perintah mount gagal dengan pesan kesalahan berikut.

```
mount.nfs: an incorrect mount option was specified
```

Tindakan yang harus diambil

Pesan kesalahan ini kemungkinan besar berarti bahwa distribusi Linux Anda tidak mendukung Sistem File Jaringan versi 4.0 dan 4.1 (NFSv4). Untuk mengonfirmasi hal ini terjadi, Anda dapat menjalankan perintah berikut.

```
$ grep CONFIG_NFS_V4_1 /boot/config*
```

Jika perintah sebelumnya kembali `# CONFIG_NFS_V4_1 is not set`, NFSv4.1 tidak didukung pada distribusi Linux Anda. Untuk daftar Amazon Machine Images (AMI) untuk Amazon Elastic Compute Cloud (Amazon EC2) yang mendukung .1, lihat. NFSv4 [Dukungan NFS](#)

Pemasangan dengan titik akses gagal

Perintah mount gagal saat memasang dengan titik akses, dengan pesan kesalahan berikut:

```
mount.nfs4: mounting access_point failed, reason given by server: No such file or directory
```

Tindakan yang harus diambil

Pesan kesalahan ini menunjukkan bahwa jalur EFS yang ditentukan tidak ada. Pastikan Anda memberikan kepemilikan dan izin untuk direktori root titik akses. EFS tidak akan membuat direktori root tanpa informasi ini. Untuk informasi selengkapnya, lihat [Bekerja dengan titik akses](#).

Jika Anda tidak menentukan kepemilikan dan izin direktori root apa pun, dan direktori root belum ada, EFS tidak akan membuat direktori root. Ketika ini terjadi, setiap upaya untuk me-mount sistem file menggunakan titik akses akan gagal.

Pemasangan sistem file gagal segera setelah pembuatan sistem file

Diperlukan waktu hingga 90 detik setelah membuat target pemasangan untuk data Domain Name Service (DNS) untuk disebarkan sepenuhnya dalam file. AWS Region

Tindakan yang harus diambil

Jika Anda membuat dan memasang sistem file secara terprogram, misalnya dengan CloudFormation template, sebaiknya Anda menerapkan kondisi tunggu.

Pemasangan sistem file hang dan kemudian gagal dengan kesalahan timeout

Perintah pemasangan sistem file hang selama satu atau dua menit, dan kemudian gagal dengan kesalahan timeout. Kode berikut menunjukkan contoh.

```
$ sudo mount -t nfs -o  
nfsvers=4.1,rsiz=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport mount-  
target-ip:/ mnt  
  
[2+ minute wait here]  
mount.nfs: Connection timed out  
$
```

Tindakan yang harus diambil

Kesalahan ini dapat terjadi karena instans Amazon EC2 atau grup keamanan target mount tidak dikonfigurasi dengan benar. Pastikan grup keamanan target mount memiliki aturan masuk yang memungkinkan akses NFS pada port 2049 dari grup keamanan EC2. Untuk informasi selengkapnya, lihat [Menggunakan grup keamanan VPC](#).

Verifikasi bahwa alamat IP target mount yang Anda tentukan valid. Jika Anda menentukan alamat IP yang salah dan tidak ada yang lain di alamat IP itu untuk menolak pemasangan, Anda mungkin mengalami masalah ini.

Pemasangan sistem file dengan NFS menggunakan nama DNS gagal

Upaya untuk me-mount sistem file menggunakan klien NFS (tidak menggunakan `amazon-efs-utils` klien) menggunakan nama DNS sistem file gagal, seperti yang ditunjukkan dalam contoh berikut:

```
$ sudo mount -t nfs -o  
nfsvers=4.1,rsiz=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport file-  
system-id.efs.aws-region.amazonaws.com:/ mnt
```

```
mount.nfs: Failed to resolve server file-system-id.efs.aws-region.amazonaws.com:
  Name or service not known.

$
```

Tindakan yang harus diambil

Periksa konfigurasi VPC Anda. Jika Anda menggunakan VPC kustom, pastikan bahwa pengaturan DNS diaktifkan. Untuk informasi selengkapnya, lihat [Atribut DNS untuk VPC Anda](#) dalam Panduan Pengguna Amazon VPC. Selain itu, sistem file dan nama DNS target mount tidak dapat diselesaikan dari luar VPC tempat mereka ada.

Sebelum Anda dapat memasang sistem file menggunakan nama DNS-nya dalam mount perintah, Anda harus melakukan hal berikut:

- Pastikan ada target pemasangan Amazon EFS di Availability Zone yang sama dengan instans Amazon EC2.
- Pastikan ada target pemasangan di VPC yang sama dengan instans Amazon EC2. Jika tidak, Anda tidak dapat menggunakan resolusi nama DNS untuk target pemasangan EFS yang ada di VPC lain. Untuk informasi selengkapnya, lihat [Memasang sistem file EFS dari yang lain Akun AWS atau VPC](#).
- Hubungkan instans Amazon EC2 Anda di dalam VPC Amazon yang dikonfigurasi untuk menggunakan server DNS yang disediakan oleh Amazon. Untuk informasi selengkapnya, lihat [set opsi DHCP di Amazon VPC](#) di Panduan Pengguna Amazon VPC.
- Pastikan VPC Amazon dari instans Amazon EC2 yang menghubungkan mengaktifkan nama host DNS. Untuk informasi selengkapnya, lihat [atribut DNS di VPC Anda di](#) Panduan Pengguna Amazon VPC.

Pemasangan sistem file gagal dengan “nfs tidak merespons”

Pemasangan sistem file Amazon EFS gagal pada peristiwa koneksi ulang Transmission Control Protocol (TCP) dengan. "nfs: server_name still not responding"

Tindakan yang harus diambil

Gunakan opsi `noresvport` mount untuk memastikan bahwa klien NFS menggunakan port sumber TCP baru ketika koneksi jaringan dibangun kembali. Melakukan hal ini membantu memastikan ketersediaan tanpa gangguan setelah peristiwa pemulihan jaringan.

Status siklus hidup target mount macet

Status siklus hidup target mount macet dalam status pembuatan atau penghapusan.

Tindakan yang harus diambil

Coba lagi `CreateMountTarget` atau `DeleteMountTarget` hubungi.

Status siklus hidup target pemasangan menunjukkan kesalahan

Status siklus hidup target mount ditampilkan sebagai kesalahan.

Tindakan yang harus diambil

Amazon EFS tidak dapat membuat data Domain Name System (DNS) yang diperlukan untuk target pemasangan sistem file baru jika virtual private cloud (VPC) memiliki zona host yang bertentangan. Amazon EFS tidak dapat membuat catatan baru dalam zona host milik pelanggan. Jika Anda perlu mempertahankan zona yang dihosting dengan rentang `efs.<region>.amazonaws.com` DNS yang bertentangan, buat zona yang dihosting di VPC terpisah. Untuk informasi selengkapnya tentang pertimbangan DNS untuk VPC, lihat [atribut DNS](#) untuk VPC Anda.

Untuk mengatasi masalah ini, hapus `efs.<region>.amazonaws.com` host yang bertentangan dari VPC dan buat target pemasangan lagi. Untuk informasi selengkapnya tentang menghapus target pemasangan, lihat [Mengelola target mount](#).

Mount tidak merespons

Mount Amazon EFS tampak tidak responsif. Misalnya, perintah seperti `ls` hang.

Tindakan yang harus diambil

Kesalahan ini dapat terjadi jika aplikasi lain menulis data dalam jumlah besar ke sistem file. Akses ke file yang sedang ditulis mungkin diblokir sampai operasi selesai. Secara umum, perintah atau aplikasi apa pun yang mencoba mengakses file yang sedang ditulis mungkin tampak hang. Misalnya, `ls` perintah mungkin hang ketika sampai ke file yang sedang ditulis. Hasil ini karena beberapa distribusi Linux alias `ls` perintah sehingga mengambil atribut file selain mencantumkan isi direktori.

Untuk mengatasi masalah ini, verifikasi bahwa aplikasi lain sedang menulis file ke mount Amazon EFS, dan berada dalam status `Uninterruptible sleep (D)`, seperti pada contoh berikut:

```
$ ps aux | grep large_io.py
```

```
root 33253 0.5 0.0 126652 5020 pts/3 D+ 18:22 0:00 python large_io.py /efs/large_file
```

Setelah Anda memverifikasi bahwa ini masalahnya, Anda dapat mengatasi masalah dengan menunggu operasi penulisan lainnya selesai, atau dengan menerapkan solusi. Dalam contoh `ls`, Anda dapat menggunakan `/bin/ls` perintah secara langsung, bukan alias. Melakukan hal ini memungkinkan perintah untuk melanjutkan tanpa menggantung pada file yang sedang ditulis. Secara umum, jika aplikasi yang menulis data dapat memaksa data flush secara berkala, mungkin dengan menggunakan `fsync(2)`, hal itu dapat membantu meningkatkan daya tanggap sistem file Anda untuk aplikasi lain. Namun, peningkatan ini mungkin mengorbankan kinerja ketika aplikasi menulis data.

Klien yang dipasang akan terputus

Klien yang dipasang ke sistem file Amazon EFS kadang-kadang dapat terputus karena sejumlah penyebab. Klien NFS dirancang untuk menyambung kembali secara otomatis jika terjadi gangguan untuk meminimalkan dampak pemutusan rutin pada kinerja dan ketersediaan aplikasi. Dalam kebanyakan kasus, klien secara transparan terhubung kembali dalam hitungan detik.

Namun, perangkat lunak klien NFS yang disertakan dalam versi kernel Linux yang lebih lama (versi v5.4 dan di bawahnya) menyertakan perilaku yang menyebabkan klien NFS, setelah terputus, mencoba menghubungkan kembali pada port sumber TCP yang sama. Perilaku ini tidak sesuai dengan TCP RFC, dan dapat mencegah klien ini dengan cepat membangun kembali koneksi ke server NFS mereka (dalam hal ini, sistem file EFS).

Untuk mengatasi masalah ini, kami sangat menyarankan Anda menggunakan helper mount Amazon EFS untuk memasang sistem file EFS Anda. EFS mount helper menggunakan pengaturan mount yang dioptimalkan untuk sistem file Amazon EFS. Untuk informasi selengkapnya tentang klien EFS dan mount helper, lihat [Menginstal klien Amazon EFS](#).

Jika Anda tidak dapat menggunakan EFS mount helper, kami sangat menyarankan untuk menggunakan opsi pemasangan `noresvport` NFS, yang menginstruksikan klien NFS untuk membangun kembali koneksi menggunakan port sumber TCP baru untuk menghindari masalah ini. Untuk informasi selengkapnya, lihat [Pengaturan pemasangan NFS yang disarankan](#).

Operasi pada sistem file yang baru dipasang mengembalikan Kesalahan “pegangan file buruk”

Operasi yang dilakukan pada sistem file yang baru dipasang mengembalikan `bad file handle` kesalahan.

Kesalahan ini dapat terjadi jika instans Amazon EC2 terhubung ke satu sistem file dan satu target pemasangan dengan alamat IP tertentu, dan kemudian sistem file dan target pemasangan dihapus. Jika Anda membuat sistem file baru dan memasang target untuk terhubung ke instans Amazon EC2 dengan alamat IP target mount yang sama, masalah ini dapat terjadi.

Tindakan yang harus diambil

Anda dapat mengatasi kesalahan ini dengan melepas sistem file, dan kemudian memasang ulang sistem file pada instans Amazon EC2. Untuk informasi selengkapnya tentang melepas sistem file Amazon EFS Anda, lihat [Melepaskan sistem file](#).

Melepas sistem file gagal

Jika sistem file Anda sibuk, Anda tidak dapat melepasnya.

Tindakan yang harus diambil

Anda dapat mengatasi masalah ini dengan cara berikut:

- Gunakan lazy unmount, `umount -l` yang melepaskan sistem file dari hierarki sistem file saat dijalankan, lalu bersihkan semua referensi ke sistem file segera setelah tidak sibuk lagi.
- Tunggu semua operasi baca dan tulis selesai, lalu coba `umount` perintahnya lagi.
- Paksa unmount menggunakan `umount -f` perintah.

Warning

Memaksa unmount mengganggu operasi baca atau tulis data apa pun yang saat ini sedang dalam proses untuk sistem file. Lihat [halaman manual umount untuk](#) informasi dan panduan lebih lanjut saat menggunakan opsi ini.

Mentransfer data masuk dan keluar dari Amazon EFS

Anda dapat menggunakan AWS DataSync dan AWS Transfer Family mentransfer data masuk dan keluar dari sistem file Amazon EFS Anda. AWS DataSync adalah layanan transfer data online yang dapat menyalin data antara Network File System (NFS), server file Server Message Block (SMB), penyimpanan objek yang dikelola sendiri, dan juga antar layanan. AWS Untuk informasi selengkapnya tentang penggunaan DataSync dengan Amazon EFS, lihat [Menggunakan AWS DataSync untuk mentransfer data](#).

AWS Transfer Family adalah AWS layanan yang dikelola sepenuhnya yang dapat Anda gunakan untuk mentransfer file masuk dan keluar dari sistem file Amazon EFS melalui protokol Secure File Transfer Protocol (SFTP), File Transfer Protocol (FTP), dan FTP melalui protokol Secure Sockets Layer (FTPS). Menggunakan Transfer Family, Anda dapat memberi mitra bisnis Anda akses ke file yang disimpan dalam sistem file Amazon EFS Anda untuk kasus penggunaan seperti distribusi data, rantai pasokan, manajemen konten, dan aplikasi penayangan web. Untuk informasi selengkapnya tentang penggunaan Transfer Family dengan Amazon EFS, lihat [Menggunakan AWS Transfer Family untuk mentransfer data](#).

Topik

- [Menggunakan AWS DataSync untuk mentransfer data](#)
- [Menggunakan AWS Transfer Family untuk mentransfer data](#)

Menggunakan AWS DataSync untuk mentransfer data

AWS DataSync adalah layanan transfer data online yang menyederhanakan, mengotomatisasi, dan mempercepat pemindahan dan replikasi data antara sistem penyimpanan lokal, dan juga antara layanan penyimpanan. AWS DataSync dapat menyalin data antara Network File System (NFS), server file Server Message Block (SMB), penyimpanan objek yang dikelola sendiri,, bucket Amazon AWS Snowball Edge S3, sistem file EFS, FSx dan untuk sistem file Windows File Server.

Anda juga dapat menggunakan DataSync untuk mentransfer file antara dua sistem file EFS, termasuk sistem file dalam AWS Region s yang berbeda dan sistem file yang dimiliki oleh Akun AWS s yang berbeda. Menggunakan DataSync untuk menyalin data antara sistem file EFS, Anda dapat melakukan migrasi data satu kali, konsumsi data berkala untuk beban kerja terdistribusi, dan mengotomatiskan replikasi untuk perlindungan dan pemulihan data.

Untuk informasi selengkapnya, lihat [Memulai dengan Amazon EFS](#) dan [Panduan Pengguna AWS DataSync](#).

Menggunakan AWS Transfer Family untuk mentransfer data

AWS Transfer Family adalah AWS layanan terkelola penuh yang dapat Anda gunakan untuk mentransfer file masuk dan keluar dari sistem file Amazon EFS melalui protokol berikut:

- Protokol Transfer File Secure Shell (SSH) (SFTP) (AWS Transfer for SFTP)
- Protokol Transfer File Aman (FTPS) (AWS Transfer for FTPS)
- Protokol Transfer File (FTP) (AWS Transfer for FTP)

Menggunakan Transfer Family, Anda dapat dengan aman mengaktifkan akses pihak ketiga seperti vendor, mitra, atau pelanggan Anda ke file Anda melalui protokol yang didukung secara global, tanpa perlu mengelola infrastruktur apa pun. Selain itu, Anda sekarang dapat dengan mudah mengakses sistem file EFS Anda dari lingkungan Windows, macOS, dan Linux menggunakan klien SFTP, FTPS, dan FTP. Ini membantu memperluas aksesibilitas data Anda di luar klien NFS dan titik akses, kepada pengguna di berbagai lingkungan.

Menggunakan Transfer Family untuk mentransfer data dalam sistem file Amazon EFS diperhitungkan dengan cara yang sama seperti penggunaan klien lainnya. Untuk informasi selengkapnya, lihat [Mode throughput](#) dan [Kuota Amazon EFS](#).

Untuk mempelajari selengkapnya AWS Transfer Family, lihat [Panduan AWS Transfer Family Pengguna](#).

Note

Menggunakan Transfer Family dengan Amazon EFS dinonaktifkan secara default untuk Akun AWS yang memiliki sistem file Amazon EFS dengan kebijakan yang memungkinkan akses publik yang dibuat sebelum 6 Januari 2021. Untuk mengaktifkan penggunaan Transfer Family untuk mengakses sistem file Anda, hubungi Dukungan.

Topik

- [Prasyarat untuk digunakan dengan Amazon EFS AWS Transfer Family](#)

- [Mengkonfigurasi sistem file EFS Anda untuk AWS Transfer Family](#)

Prasyarat untuk digunakan dengan Amazon EFS AWS Transfer Family

Untuk menggunakan Transfer Family untuk mengakses file sistem file Amazon EFS Anda, konfigurasi Anda harus memenuhi ketentuan berikut:

- Server Transfer Family dan sistem file Amazon EFS Anda berada di tempat yang sama AWS Region.
- Peran IAM yang digunakan oleh Transfer Family harus memiliki izin Amazon EFS (seperti `elasticfilesystem:ClientMount` dan `elasticfilesystem:ClientWrite`) yang dilampirkan. Untuk informasi selengkapnya tentang izin dan kebijakan, lihat [Membuat peran dan kebijakan IAM](#) di AWS Transfer Family Panduan Pengguna.
- (Opsional) Jika server Transfer Family dimiliki oleh akun lain, aktifkan akses lintas akun.
 - Pastikan kebijakan sistem file Anda tidak mengizinkan akses publik. Untuk informasi selengkapnya, lihat [Memblokir akses publik ke sistem file EFS](#).
 - Ubah kebijakan sistem file untuk mengaktifkan akses lintas akun. Untuk informasi selengkapnya, lihat [Mengkonfigurasi akses lintas akun untuk Transfer Family](#).

Mengkonfigurasi sistem file EFS Anda untuk AWS Transfer Family

Mengonfigurasi sistem file Amazon EFS agar berfungsi dengan Transfer Family memerlukan langkah-langkah berikut:

- Langkah 1. Dapatkan daftar POSIX IDs yang dialokasikan untuk pengguna Transfer Family.
- Langkah 2. Pastikan direktori sistem file Anda dapat diakses oleh pengguna Transfer Family dengan menggunakan POSIX yang IDs dialokasikan untuk pengguna Transfer Family.
- Langkah 3. Konfigurasikan IAM untuk mengaktifkan akses ke peran IAM yang digunakan oleh Transfer Family.

Mengatur izin berkas dan direktori untuk pengguna Transfer Family

Pastikan bahwa pengguna Transfer Family memiliki akses ke file dan direktori yang diperlukan pada sistem file EFS Anda. Tetapkan izin akses ke direktori menggunakan daftar POSIX yang IDs dialokasikan untuk pengguna Transfer Family. Dalam contoh ini, pengguna membuat direktori

bernama `transferFam` di bawah titik pemasangan Amazon EFS. Membuat direktori adalah opsional, tergantung pada kasus penggunaan Anda. Jika perlu, Anda dapat memilih nama dan lokasinya di sistem file EFS.

Untuk menetapkan izin file dan direktori ke pengguna POSIX untuk Transfer Family

1. Hubungkan ke instans Amazon EC2 Anda. Amazon EFS hanya mendukung pemasangan dengan instans EC2 berbasis Linux.
2. Pasang sistem file EFS Anda jika belum dipasang pada instans EC2. Untuk informasi selengkapnya, lihat [Memasang sistem file EFS](#).
3. Contoh berikut membuat direktori pada sistem file EFS, dan mengubah grupnya menjadi ID grup POSIX untuk pengguna Transfer Family, yaitu 1101 dalam contoh ini.
 - a. Buat direktori `efs/transferFam` menggunakan perintah berikut. Dalam praktiknya, Anda dapat menggunakan nama dan lokasi pada sistem file yang Anda pilih.

```
[ec2-user@ip-192-0-2-0 ~]$ ls
efs  efs-mount-point  efs-mount-point2
[ec2-user@ip-192-0-2-0 ~]$ ls efs
[ec2-user@ip-192-0-2-0 ~]$ sudo mkdir efs/transferFam
[ec2-user@ip-192-0-2-0 ~]$ ls -l efs
total 0
drwxr-xr-x 2 root root 6 Jan  6 15:58 transferFam
```

- b. Gunakan perintah berikut untuk mengubah grup ke POSIX GID yang ditetapkan untuk pengguna Transfer Family. `efs/transferFam`

```
[ec2-user@ip-192-0-2-0 ~]$ sudo chown :1101 efs/transferFam/
```

- c. Konfirmasikan perubahan.

```
[ec2-user@ip-192-0-2-0 ~]$ ls -l efs
total 0
drwxr-xr-x 2 root 1101 6 Jan  6 15:58 transferFam
```

Aktifkan akses ke peran IAM yang digunakan oleh Transfer Family

Di Transfer Family, Anda membuat kebijakan IAM berbasis sumber daya dan peran IAM yang menentukan akses pengguna ke sistem file EFS. Untuk informasi selengkapnya, lihat [Membuat](#)

[peran dan kebijakan IAM](#) di Panduan AWS Transfer Family Pengguna. Anda harus memberikan akses peran Transfer Family IAM ke sistem file EFS Anda menggunakan kebijakan identitas IAM atau kebijakan sistem file.

Berikut ini adalah contoh kebijakan sistem file yang memberikan ClientMount (baca) dan ClientWrite akses ke peran IAM. EFS-role-for-transfer

JSON

```
{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-8698b356-4212-4d30-901e-ad2030b57762",
  "Statement": [
    {
      "Sid": "Grant-transfer-role-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/EFS-role-for-transfer"
      },
      "Action": [
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/*"
    }
  ]
}
```

Untuk informasi selengkapnya tentang membuat kebijakan sistem file, lihat [Membuat kebijakan sistem file](#). Untuk informasi selengkapnya tentang penggunaan kebijakan IAM berbasis identitas untuk mengelola akses ke sumber daya EFS, lihat [Kebijakan berbasis identitas untuk Amazon EFS](#)

Mengonfigurasi akses lintas akun untuk Transfer Family

Jika server Transfer Family yang digunakan untuk mengakses sistem file Anda milik yang berbeda Akun AWS, Anda harus memberikan akses akun tersebut ke sistem file Anda. Selain itu, kebijakan sistem file Anda harus non-publik. Untuk informasi selengkapnya tentang memblokir akses publik ke sistem file Anda, lihat [Memblokir akses publik ke sistem file EFS](#).

Anda dapat memberikan Akun AWS akses berbeda ke sistem file Anda dalam kebijakan sistem file. Di konsol Amazon EFS, gunakan bagian Berikan izin tambahan pada editor kebijakan sistem File untuk menentukan Akun AWS dan tingkat akses sistem file yang Anda berikan. Untuk informasi selengkapnya tentang membuat atau mengedit kebijakan sistem file, lihat [Membuat kebijakan sistem file](#).

Anda dapat menentukan akun menggunakan ID akun atau akun Nama Sumber Daya Amazon (ARN). Untuk informasi selengkapnya ARNs, lihat [IAM ARNs](#) di Panduan Pengguna IAM.

Contoh berikut adalah kebijakan sistem file non-publik yang memberikan akses lintas akun ke sistem file. Ini memiliki dua pernyataan berikut:

1. Pernyataan pertama, `NFS-client-read-write-via-fsmt`, memberikan hak baca, tulis, dan root kepada klien NFS yang mengakses sistem file menggunakan target pemasangan sistem file.
2. Pernyataan kedua, `Grant-cross-account-access`, hanya memberikan hak baca dan tulis ke Akun AWS 111122223333, yang merupakan akun yang memiliki server Transfer Family yang memerlukan akses ke sistem file EFS ini di akun Anda.

```
{
  "Statement": [
    {
      "Sid": "NFS-client-read-write-via-fsmt",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ],
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    },
    {
      "Sid": "Grant-cross-account-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
```

```

        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
    ]
}
]
}

```

Kebijakan sistem berkas berikut menambahkan pernyataan yang memberikan akses ke peran IAM yang digunakan oleh Transfer Family.

```

{
  "Statement": [
    {
      "Sid": "NFS-client-read-write-via-fsmt",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ],
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    },
    {
      "Sid": "Grant-cross-account-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ]
    },
    {
      "Sid": "Grant-transfer-role-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/EFS-role-for-transfer"
      },
    }
  ]
}

```

```
    "Action": [  
      "elasticfilesystem:ClientWrite",  
      "elasticfilesystem:ClientMount"  
    ]  
  }  
]  
}
```

Mengelola sistem file EFS

Tugas manajemen sistem file termasuk mengelola aksesibilitas jaringan sistem file dengan target mount, mengubah mode throughputnya, memperbarui kebijakan siklus hidupnya, dan mengelola enkripsi.

Anda dapat melakukan tugas manajemen sistem file ini menggunakan Konsol Manajemen AWS, atau secara terprogram menggunakan AWS Command Line Interface (AWS CLI) atau API, seperti yang dibahas di bagian berikut.

Topik

- [Memahami status sistem berkas](#)
- [Mengelola target mount](#)
- [Mengelola throughput sistem file](#)
- [Mengelola siklus hidup penyimpanan](#)

Memahami status sistem berkas

Anda dapat melihat status sistem file Amazon EFS menggunakan konsol Amazon EFS atau AWS CLI. Sistem file Amazon EFS dapat memiliki salah satu nilai status yang dijelaskan dalam tabel berikut.

Status sistem berkas	Deskripsi
TERSEDIA	Sistem file dalam keadaan sehat, dan dapat dijangkau dan tersedia untuk digunakan.
CREATING	Amazon EFS sedang dalam proses pembuatan sistem file baru.
DELETING	Amazon EFS menghapus sistem file sebagai respons terhadap permintaan penghapusan yang diprakarsai pengguna. Untuk informasi selengkapnya, lihat Menghapus sistem file EFS .

Status sistem berkas	Deskripsi
DELETED	Amazon EFS telah menghapus sistem file sebagai tanggapan atas permintaan penghapusan yang diprakarsai pengguna. Untuk informasi selengkapnya, lihat Menghapus sistem file EFS .
UPDATING	Sistem file sedang menjalani pembaruan sebagai tanggapan atas permintaan pembaruan yang diprakarsai pengguna.
ERROR	Berlaku untuk sistem file One Zone, termasuk sistem file dalam konfigurasi replikasi. Sistem file dalam keadaan gagal dan tidak dapat dipulihkan. Untuk mengakses data sistem file, kembalikan cadangan sistem file ini ke sistem file baru. Untuk informasi selengkapnya, lihat Mencadangkan dan mereplikasi data di Amazon EFS

Mengelola target mount

Anda memasang sistem file di Amazon EC2 atau instans AWS komputasi lainnya di virtual private cloud (VPC) menggunakan satu atau beberapa target mount yang Anda buat untuk sistem file. Anda dapat membuat target mount ketika Anda membuat sistem file atau setelah Anda membuat sistem file.

Setelah target pemasangan dibuat untuk sistem file, Anda dapat membuat target pemasangan tambahan, menghapus target pemasangan, dan memodifikasi grup keamanan untuk target pemasangan. Jika Anda ingin memodifikasi VPC untuk target mount, maka Anda harus terlebih dahulu menghapus target mount yang ada.

Note

Anda tidak dapat mengubah alamat IP dari target mount yang ada. Untuk mengubah alamat IP, Anda perlu menghapus target mount dan membuat yang baru dengan alamat baru.

Topik

- [Pasang target dan Availability Zone](#)

- [Pembuatan target pemasangan](#)
- [Menghapus target mount](#)
- [Mengubah VPC target mount](#)
- [Mengubah grup keamanan target mount](#)

Pasang target dan Availability Zone

Untuk sistem file EFS yang menggunakan kelas penyimpanan Regional, Anda dapat membuat target pemasangan di setiap Availability Zone dalam file AWS Region.

Untuk sistem file One Zone, Anda hanya dapat membuat target mount tunggal di Availability Zone yang sama dengan sistem file. Kemudian Anda dapat memasang sistem file pada instans komputasi, termasuk Amazon EC2, Amazon ECS AWS Lambda , dan di cloud pribadi virtual (VPC) Anda.

Diagram berikut menunjukkan sistem file Regional dengan target mount yang dibuat di semua Availability Zone di VPC. Ilustrasi menunjukkan tiga instans EC2 diluncurkan di subnet VPC berbeda yang mengakses sistem file EFS. Ilustrasi juga menunjukkan satu target pemasangan di setiap Availability Zone (terlepas dari jumlah subnet di setiap Availability Zone).

Anda hanya dapat membuat satu target mount per Availability Zone. Jika Availability Zone memiliki beberapa subnet, seperti yang ditunjukkan di salah satu zona dalam ilustrasi, Anda membuat target mount hanya di salah satu subnet. Selama Anda memiliki satu target pemasangan di Availability Zone, instans EC2 yang diluncurkan di salah satu subnetnya dapat berbagi target pemasangan yang sama.

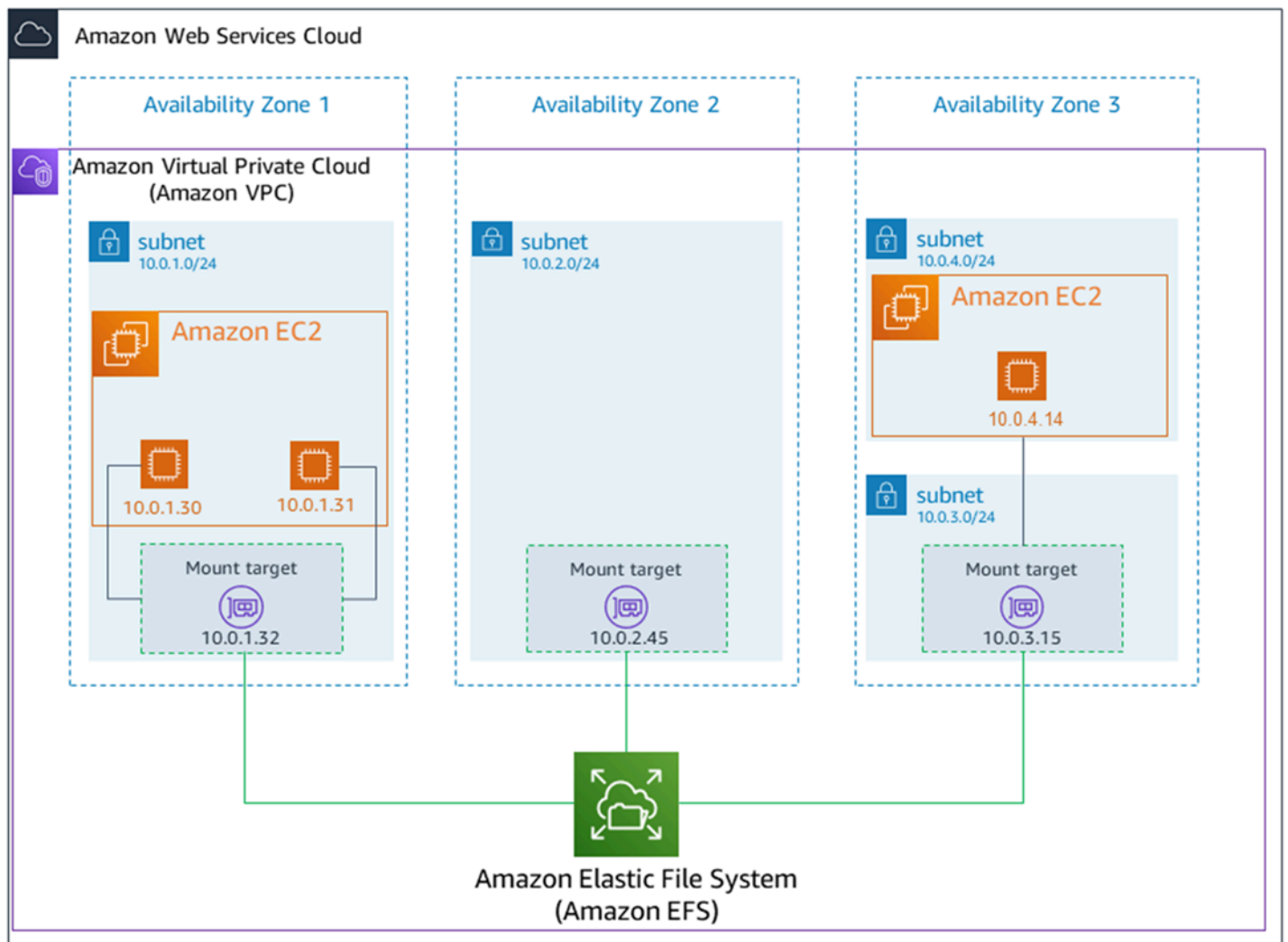
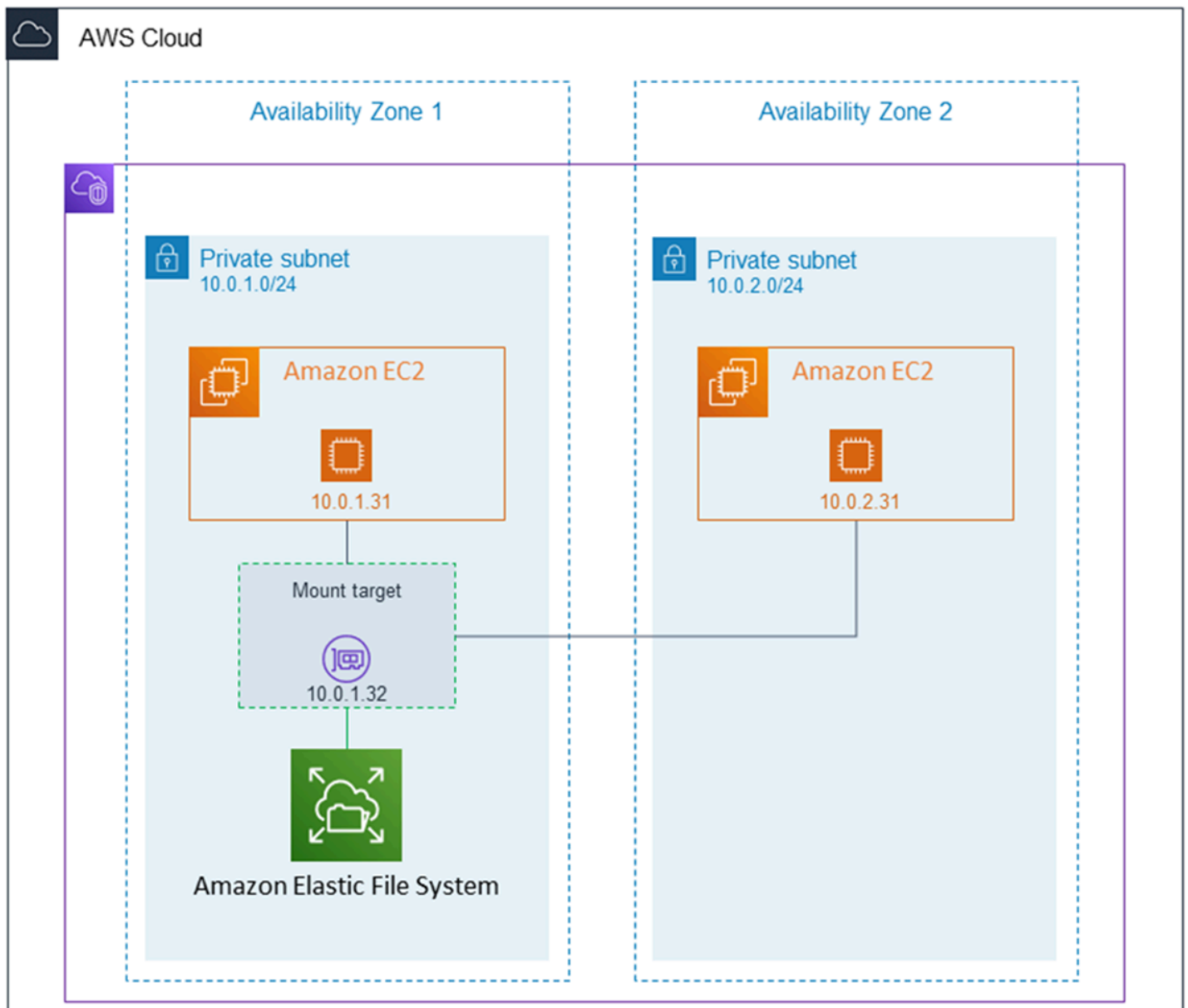


Diagram berikut menunjukkan sistem file One Zone, dengan target mount tunggal yang dibuat di Availability Zone yang sama dengan sistem file. Mengakses sistem file dengan menggunakan instans EC2 di us-west2c Availability Zone menimbulkan biaya akses data karena terletak di Availability Zone yang berbeda dari target mount.



Grup keamanan target mount bertindak sebagai firewall virtual yang mengontrol lalu lintas. Misalnya, ini menentukan klien mana yang dapat mengakses sistem file. Bagian ini menjelaskan hal berikut:

- Mengelola grup keamanan target mount dan mengaktifkan lalu lintas.
- Memasang sistem file pada klien Anda.
- Pertimbangan izin tingkat NFS.

Awalnya, hanya pengguna root pada instans Amazon EC2 yang memiliki read-write-execute izin pada sistem file. Topik ini membahas izin tingkat NFS dan memberikan contoh yang menunjukkan

cara memberikan izin dalam skenario umum. Untuk informasi selengkapnya, lihat [Network File System \(NFS\) tingkat pengguna, grup, dan izin](#).

Pembuatan target pemasangan

Untuk mengakses sistem file EFS di VPC, Anda perlu membuat target mount untuk sistem file.

Untuk sistem file EFS, berikut ini benar:

- Anda dapat membuat target mount untuk sistem file dalam satu VPC sekaligus. Jika Anda ingin mengakses sistem file dari VPC lain, Anda perlu menghapus target mount dari VPC saat ini dan kemudian membuat target mount baru di VPC lainnya. Untuk informasi selengkapnya, lihat [Mengubah VPC target mount](#).
- Jika VPC memiliki beberapa subnet di Availability Zone, Anda dapat membuat target mount hanya di salah satu subnet tersebut. Semua instans EC2 di Availability Zone dapat berbagi target pemasangan tunggal.
- Minimal, Anda harus membuat target mount di setiap Availability Zone tempat Anda ingin mengakses sistem file.

Note

Ada pertimbangan biaya untuk memasang sistem file pada instans EC2 di Availability Zone melalui target mount yang dibuat di Availability Zone lain. Untuk informasi selengkapnya, lihat [Harga Amazon EFS](#). Selain itu, dengan selalu menggunakan target mount lokal ke Availability Zone instance, Anda menghapus skenario kegagalan sebagian. Jika zona target mount turun, Anda tidak dapat mengakses sistem file Anda melalui target mount itu.

Anda dapat membuat target mount untuk sistem file dengan menggunakan Konsol Manajemen AWS, AWS CLI, atau secara terprogram dengan menggunakan file. AWS SDKs Di konsol, Anda dapat membuat target mount saat Anda membuat sistem file atau setelah sistem file dibuat. Untuk petunjuk cara membuat target mount saat membuat sistem file, lihat [Buat kustom menggunakan konsol](#).

Menggunakan konsol

Gunakan prosedur berikut untuk menambahkan target mount ke sistem file EFS yang ada.

Untuk membuat target mount pada sistem file EFS

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file, lalu pilih sistem file yang ingin Anda ubah VPC.
3. Pilih Jaringan dan kemudian pilih Kelola untuk menampilkan target pemasangan untuk sistem file.
4. Pilih sistem file yang ingin Anda tambahkan target mount dengan memilih Namanya atau ID sistem File.

Note

Untuk sistem file One Zone, Anda hanya dapat membuat target mount tunggal yang berada di Availability Zone yang sama dengan sistem file.

5. Untuk sistem file yang menggunakan kelas penyimpanan EFS Regional, pilih Tambahkan target pemasangan untuk setiap target pemasangan yang ingin Anda buat untuk sistem file.
6. Tentukan pengaturan target pemasangan:
 - a. Pilih Availability Zone dan subnet ID untuk target mount.
 - b. Untuk jenis alamat IP, pilih IPv4 hanya untuk mendukung IPv4 alamat saja, IPv6 hanya untuk mendukung IPv6 alamat saja, atau Dual-stack untuk mendukung keduanya IPv4 dan IPv6 alamat.

Note

Jenis alamat IP harus sesuai dengan jenis IP subnet. Selain itu, jenis alamat IP mengesampingkan atribut pengalamatan IP subnet Anda. Misalnya, jika jenis alamat IP IPv4 -only dan atribut IPv6 pengalamatan diaktifkan untuk subnet Anda, antarmuka jaringan yang dibuat di subnet menerima IPv4 alamat dari kisaran subnet. Untuk informasi selengkapnya, lihat [Memodifikasi atribut pengalamatan IP subnet Anda](#).

- c. Jika Anda mengetahui alamat IP tempat Anda ingin menempatkan target mount, masukkan di kotak alamat IP yang cocok dengan jenis alamat IP. Jika Anda tidak menentukan nilai, Amazon EFS memilih alamat IP yang tidak digunakan dari subnet yang ditentukan.

Note

Anda tidak dapat mengubah alamat IP target mount setelah dibuat. Untuk mengubah alamat IP, Anda perlu menghapus target mount dan membuat yang baru dengan alamat baru.

7. Pilih setidaknya satu grup keamanan untuk dikaitkan dengan target pemasangan. Anda dapat [memodifikasi grup keamanan](#) nanti.
8. Pilih Simpan.

Menggunakan AWS CLI

Bagian ini memberikan contoh untuk membuat target mount dalam AWS CLI menggunakan `create-mount-target` perintah. Perintah API yang setara adalah [CreateMountTarget](#).

- Jika Anda tidak menentukan jenis alamat IP untuk target mount, maka IPv4 -only digunakan.
- Jika Anda tidak menentukan alamat IP untuk target pemasangan, Amazon EFS akan menetapkan alamat yang tersedia di subnet yang ditentukan.
- Jenis alamat IP mengesampingkan atribut pengalamatan IP subnet Anda. Misalnya, jika jenis alamat IP IPv4 -only dan atribut IPv6 pengalamatan diaktifkan untuk subnet Anda, antarmuka jaringan yang dibuat di subnet menerima IPv4 alamat dari kisaran subnet. Untuk informasi selengkapnya, lihat [Memodifikasi atribut pengalamatan IP subnet Anda](#).

Note

Untuk sistem file One Zone, Anda hanya dapat membuat target mount tunggal yang berada di Availability Zone yang sama dengan sistem file.

Contoh: Buat target mount di IPv4 alamat yang tersedia di subnet

Perintah berikut menentukan sistem file, subnet, dan grup keamanan untuk target mount. Target dibuat di IPv4 alamat yang tersedia di subnet yang ditentukan.

```
$ aws efs create-mount-target \  
--file-system-id file-system-id \  
--subnet-id subnet-id \  
--security-groups security-groups
```

```
--security-group ID-of-the-security-group-created-for-mount-target \  
--region aws-region \
```

Contoh berikut menunjukkan perintah dengan data sampel.

```
$ aws efs create-mount-target \  
--file-system-id fs-0123456789abcdef1 \  
--subnet-id subnet-b3983dc4 \  
--security-group sg-01234567 \  
--region us-east-2 \
```

Setelah berhasil membuat target mount, Amazon EFS mengembalikan deskripsi target mount sebagai JSON seperti yang ditunjukkan pada contoh berikut.

```
{  
  "OwnerId": "111122223333"  
  "MountTargetId": "fsmt-f9a14450",  
  "FileSystemId": "fs-0123456789abcdef1",  
  "SubnetId": "subnet-b3983dc4",  
  "LifeCycleState": "available",  
  "IpAddress": "10.0.1.24",  
  "NetworkInterfaceId": "eni-3851ec4e",  
  "AvailabilityZoneId": "use2-az1",  
  "AvailabilityZoneName": "us-east-2a",  
  "VpcId": "vpc-3c39ef57"  
}
```

Contoh: Buat target pemasangan di IPv4 alamat tertentu

Perintah berikut menentukan sistem file, subnet, grup keamanan, dan IPv4 alamat yang akan digunakan untuk target mount. Target dibuat pada IPv4 alamat yang ditentukan pada subnet yang ditentukan.

```
$ aws efs create-mount-target \  
--file-system-id file-system-id \  
--subnet-id subnet-id \  
--security-group ID-of-the-security-group-created-for-mount-target \  
--ip-address IPv4-address \  
--region aws-region \
```

Contoh berikut menunjukkan perintah dengan data sampel.

```
$ aws efs create-mount-target \  
--file-system-id fs-0123456789abcdef1 \  
--subnet-id subnet-b3983dc4 \  
--security-group sg-01234567 \  
--ip-address 10.0.1.24 \  
--region us-east-2 \  

```

Setelah berhasil membuat target mount, Amazon EFS mengembalikan deskripsi target mount sebagai JSON seperti yang ditunjukkan pada contoh berikut.

```
{  
  "OwnerId": "111122223333"  
  "MountTargetId": "fsmt-f9a14450",  
  "FileSystemId": "fs-0123456789abcdef1",  
  "SubnetId": "subnet-b3983dc4",  
  "LifecycleState": "available",  
  "IpAddress": "10.0.1.24",  
  "NetworkInterfaceId": "eni-3851ec4e",  
  "AvailabilityZoneId": "use2-az1",  
  "AvailabilityZoneName": "us-east-2a",  
  "VpcId": "vpc-3c39ef57"  
}
```

Contoh: Buat target pemasangan di IPv6 alamat tertentu

Perintah berikut menentukan sistem file, subnet, grup keamanan, dan IPv6 alamat yang akan digunakan untuk target mount. Target dibuat pada IPv6 alamat yang ditentukan pada subnet yang ditentukan.

```
$ aws efs create-mount-target \  
--file-system-id file-system-id \  
--subnet-id subnet-id \  
--security-group ID-of-the-security-group-created-for-mount-target \  
--ip-address-type IP-address-type \  
--ipv6-address IPv6-address \  
--region aws-region \  

```

Contoh berikut menunjukkan perintah dengan data sampel.

```
$ aws efs create-mount-target \  

```

```
--file-system-id fs-0123456789abcdef1 \  
--subnet-id subnet-b3983dc4 \  
--security-group sg-01234567 \  
--ip-address-type IPV6_ONLY \  
--ipv6-address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 \  
--region us-east-2 \  

```

Setelah berhasil membuat target mount, Amazon EFS mengembalikan deskripsi target mount sebagai JSON seperti yang ditunjukkan pada contoh berikut.

```
{  
  "OwnerId": "111122223333"  
  "MountTargetId": "fsmt-f9a14450",  
  "FileSystemId": "fs-0123456789abcdef1",  
  "SubnetId": "subnet-b3983dc4",  
  "LifecycleState": "available",  
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",  
  "NetworkInterfaceId": "eni-3851ec4e",  
  "AvailabilityZoneId": "use2-az1",  
  "AvailabilityZoneName": "us-east-2a",  
  "VpcId": "vpc-3c39ef57"  
}
```

Contoh: Buat target pemasangan di IPv4 alamat yang tersedia di subnet dual-stack

Perintah menentukan sistem file, subnet, grup keamanan, tipe alamat IP dual-stack, dan IPv6 alamat untuk target mount. Target dibuat di alamat yang tersedia dan IPv4 alamat yang ditentukan pada IPv6 subnet dual-stack.

```
$ aws efs create-mount-target \  
--file-system-id file-system-id \  
--subnet-id subnet-id \  
--security-group ID-of-the-security-group-created-for-mount-target \  
--ip-address-type IP-address-type \  
--ipv6-address IPv6-address \  
--region aws-region \  

```

Contoh berikut menunjukkan perintah dengan data sampel.

```
$ aws efs create-mount-target \  
--file-system-id fs-0123456789abcdef1 \  

```



```
--subnet-id subnet-b3983dc4 \  
--security-group sg-01234567 \  
--ip-address-type DUAL_STACK \  
--ipv6-address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 \  
--region us-east-2 \  

```

Setelah berhasil membuat target mount, Amazon EFS mengembalikan deskripsi target mount sebagai JSON seperti yang ditunjukkan pada contoh berikut.

```
{  
  "OwnerId": "111122223333"  
  "MountTargetId": "fsm-t-f9a14450",  
  "FileSystemId": "fs-0123456789abcdef1",  
  "SubnetId": "subnet-b3983dc4",  
  "LifeCycleState": "available",  
  "IpAddress": "10.0.1.24",  
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",  
  "NetworkInterfaceId": "eni-3851ec4e",  
  "AvailabilityZoneId": "us-east-2a",  
  "AvailabilityZoneName": "us-east-2a",  
  "VpcId": "vpc-3c39ef57"  
}
```

Menghapus target mount

Saat Anda menghapus target pemasangan, operasi secara paksa merusak setiap mount sistem file, yang mungkin mengganggu instance atau aplikasi yang menggunakan mount tersebut. Untuk menghindari gangguan aplikasi, hentikan aplikasi dan lepaskan sistem file sebelum menghapus target pemasangan. Untuk informasi selengkapnya, lihat [Melepaskan sistem file](#).

Anda dapat menghapus target mount untuk sistem file dengan menggunakan Konsol Manajemen AWS, AWS CLI, atau secara terprogram dengan menggunakan file. AWS SDKs

Menggunakan konsol

Gunakan prosedur berikut untuk menghapus target mount untuk sistem file EFS yang ada.

Untuk menghapus target mount pada sistem file EFS

1. Lepaskan sistem file. Untuk petunjuk, lihat [Melepaskan sistem file](#).
2. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.

3. Di panel navigasi kiri, pilih Sistem file, lalu pilih sistem file yang ingin Anda hapus target mount.
4. Pilih Jaringan dan kemudian pilih Kelola untuk menampilkan target pemasangan untuk sistem file.
5. Untuk setiap target pemasangan yang ingin Anda hapus, pilih Hapus.
6. Pilih Simpan.

Menggunakan AWS CLI

Untuk menghapus target mount yang ada, gunakan `delete-mount-target` AWS CLI perintah (operasi yang sesuai adalah [DeleteMountTarget](#)), seperti yang ditunjukkan berikut.

Note

Sebelum menghapus target mount, pertama unmount sistem file.

```
$ aws efs delete-mount-target \  
--mount-target-id mount-target-ID-to-delete \  
--region aws-region-where-mount-target-exists
```

Berikut ini adalah contoh dengan data sampel.

```
$ aws efs delete-mount-target \  
--mount-target-id fsmt-5751852e \  
--region us-east-2 \  

```

Mengubah VPC target mount

Anda dapat menggunakan sistem file EFS dalam satu VPC berdasarkan layanan Amazon VPC sekaligus. Artinya, Anda membuat target mount di VPC untuk sistem file Anda, dan menggunakan target mount tersebut untuk menyediakan akses ke sistem file.

Anda dapat memasang sistem file EFS dari target berikut:

- Instans Amazon EC2 dalam VPC yang sama
- Instans EC2 dalam VPC yang dihubungkan oleh peering VPC

- Server lokal dengan menggunakan AWS Direct Connect
- Server lokal melalui jaringan pribadi AWS virtual (VPN) dengan menggunakan Amazon VPC

Koneksi peering VPC adalah koneksi jaringan antara dua VPCs yang memungkinkan Anda untuk mengarahkan lalu lintas di antara mereka. Koneksi dapat menggunakan Private Internet Protocol versi 4 (IPv4) atau versi 6 (IPv6). Untuk informasi selengkapnya tentang cara kerja Amazon EFS dengan VPC peering, lihat [Memasang sistem file EFS dari yang lain Akun AWS atau VPC](#)

Untuk mengubah VPC untuk sistem file

Berikut ini adalah langkah-langkah yang akan Anda lakukan untuk mengubah VPC untuk konfigurasi jaringan sistem file EFS.

1. Hapus setiap target mount yang ditetapkan ke sistem file. Untuk petunjuk, lihat [Menghapus target mount](#).
2. Saat status target Mount untuk setiap target pemasangan Dihapus, tetapkan VPC baru dan buat target pemasangan baru untuk sistem file. Untuk petunjuk, lihat [Pembuatan target pemasangan](#).

Mengubah grup keamanan target mount

Grup keamanan menentukan akses masuk dan keluar. Saat Anda mengubah grup keamanan yang terkait dengan target pemasangan, pastikan Anda mengotorisasi akses masuk dan keluar yang diperlukan. Melakukannya memungkinkan instans EC2 Anda untuk berkomunikasi dengan sistem file. Untuk mengetahui informasi selengkapnya tentang grup keamanan, lihat [Menggunakan grup keamanan VPC](#).

Anda dapat menambahkan atau menghapus grup keamanan untuk target pemasangan sistem file dengan menggunakan Konsol Manajemen AWS, AWS CLI, atau secara terprogram dengan menggunakan AWS SDKs

Menggunakan konsol

Untuk memodifikasi grup keamanan untuk target pemasangan

Gunakan prosedur berikut untuk menambah atau menghapus grup keamanan target mount untuk sistem file EFS yang ada.

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.

2. Di panel navigasi kiri, pilih Sistem file, lalu pilih sistem file yang ingin Anda kelola target mount.
3. Pilih Jaringan dan kemudian pilih Kelola untuk menampilkan target pemasangan untuk sistem file.
4. Untuk menghapus grup keamanan dari target pemasangan, pilih X di sebelah ID grup keamanan.
5. Untuk menambahkan grup keamanan ke target pemasangan, pilih keamanan dari daftar Grup keamanan.
6. Pilih Simpan.

Menggunakan AWS CLI

Untuk memodifikasi grup keamanan yang berlaku untuk target mount, gunakan `modify-mount-target-security-group` AWS CLI perintah (operasi yang sesuai adalah [ModifyMountTargetSecurityGroups](#)) untuk mengganti grup keamanan yang ada, seperti yang ditunjukkan berikut.

```
$ aws efs modify-mount-target-security-groups \  
--mount-target-id mount-target-ID-whose-configuration-to-update \  
--security-groups security-group-ids-separated-by-space \  
--region aws-region-where-mount-target-exists \  
--profile adminuser
```

Berikut ini adalah contoh dengan data sampel.

```
$ aws efs modify-mount-target-security-groups \  
--mount-target-id fsmt-5751852e \  
--security-groups sg-1004395a sg-1114433a \  
--region us-east-2
```

Mengelola throughput sistem file

Throughput elastis adalah mode throughput default di konsol Amazon EFS dan direkomendasikan untuk sebagian besar kasus penggunaan. Dengan throughput Elastis, kinerja secara otomatis menskalakan naik atau turun untuk memenuhi kebutuhan aktivitas beban kerja Anda. Namun, jika Anda mengetahui pola akses spesifik untuk beban kerja Anda (termasuk throughput, latensi, dan kebutuhan penyimpanan), maka Anda dapat memilih untuk mengubah mode throughput.

Note

Meskipun throughput Elastic dirancang untuk menskalakan secara elastis dengan throughput Anda, kami merekomendasikan penerapan tata kelola yang tepat melalui metrik pemantauan dengan CloudWatch (TerukurIOBytes) dan peringatan penggunaan sebagai bagian dari praktik terbaik operasional Anda. Ini membantu Anda mempertahankan pemanfaatan sumber daya yang optimal dan tetap berada dalam parameter operasional yang direncanakan. Untuk informasi selengkapnya, lihat [Memantau metrik dengan Amazon CloudWatch](#).

Mode throughput lain yang dapat Anda pilih meliputi:

- Throughput yang disediakan - Anda menentukan tingkat throughput yang dapat didorong oleh sistem file secara independen dari ukuran sistem file atau saldo kredit burst.
- Throughput yang meledak — Throughput menskalakan dengan jumlah penyimpanan di sistem file Anda dan mendukung ledakan ke tingkat yang lebih tinggi hingga 12 jam per hari.

Untuk informasi selengkapnya tentang mode throughput Amazon EFS, lihat [Mode throughput](#).

Note

Anda dapat mengubah mode throughput dan jumlah throughput yang disediakan setelah sistem file tersedia. Mengubah mode throughput tidak menyebabkan downtime aplikasi. Namun, setiap kali Anda mengubah sistem file ke throughput yang disediakan atau meningkatkan jumlah throughput yang disediakan, Anda harus menunggu setidaknya 24 jam sebelum Anda dapat mengubah mode throughput lagi atau mengurangi jumlah yang disediakan.

Anda dapat mengelola mode throughput sistem file dengan menggunakan konsol Amazon EFS, AWS Command Line Interface (AWS CLI), dan Amazon EFS API.

Menggunakan konsol

Untuk mengelola throughput sistem file

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file untuk menampilkan daftar sistem file EFS di akun Anda.

3. Pilih sistem file yang ingin Anda ubah mode throughput.
4. Pada halaman detail sistem file, di bagian Umum, pilih Edit. Halaman Edit ditampilkan.
5. Ubah pengaturan mode Throughput.
 - Untuk menggunakan Elastic throughput atau Provisioned throughput, pilih Enhanced, lalu pilih Elastic atau Provisioned.

Jika Anda memilih Provisioned, maka, di Provisioned Throughput (MIB/s), masukkan jumlah throughput untuk penyediaan permintaan sistem file. Jumlah Throughput Baca Maksimum ditampilkan tiga kali jumlah throughput yang Anda masukkan. EFS file systems meteran membaca permintaan pada sepertiga tingkat permintaan lainnya. Setelah Anda memasukkan throughput, perkiraan biaya bulanan untuk sistem file ditampilkan.

Note

Anda dapat mengubah mode throughput dan jumlah throughput yang disediakan setelah sistem file tersedia. Namun, setiap kali Anda mengubah throughput sistem file ke Provisioned atau meningkatkan jumlah throughput yang disediakan, Anda harus menunggu setidaknya 24 jam sebelum Anda dapat mengubah mode throughput lagi atau mengurangi jumlah yang disediakan.

- Untuk menggunakan throughput Bursting, pilih Bursting.

Untuk informasi selengkapnya tentang memilih mode throughput yang benar untuk kebutuhan kinerja Anda, lihat [Mode throughput](#).

6. Pilih Simpan perubahan untuk menerapkan perubahan Anda.

Menggunakan AWS CLI

Gunakan perintah [update-file-system](#) CLI, atau tindakan [UpdateFileSystem](#) API untuk mengubah mode throughput sistem file.

Mengelola siklus hidup penyimpanan

Anda dapat mengelola sistem file Anda sehingga mereka memiliki penyimpanan hemat biaya sepanjang siklus hidup mereka. Gunakan manajemen siklus hidup untuk secara otomatis mentransisikan data antar kelas penyimpanan sesuai dengan konfigurasi siklus hidup untuk sistem

file. Konfigurasi siklus hidup terdiri dari tiga kebijakan siklus hidup yang Anda tetapkan untuk sistem file.

Kebijakan siklus hidup menginstruksikan manajemen siklus hidup kapan harus mentransisikan file masuk dan keluar dari kelas penyimpanan EFS Infrequent Access (IA) dan EFS Archive. Waktu transisi didasarkan pada kapan file terakhir diakses di kelas penyimpanan Standar. Untuk menentukan waktu akses terakhir di kelas penyimpanan Standar, timer internal melacak kapan file terakhir diakses (bukan atribut sistem file POSIX yang dapat dilihat publik). Setiap kali file dalam Standar diakses, timer manajemen siklus hidup diatur ulang.

Kebijakan siklus hidup berlaku untuk seluruh sistem file EFS.

Kebijakan siklus hidup EFS adalah:

- Transisi ke IA - Menginstruksikan manajemen siklus hidup kapan harus memindahkan file ke kelas penyimpanan Akses Jarang (IA), yang dioptimalkan biaya untuk data yang diakses hanya beberapa kali setiap kuartal. Secara default, file yang tidak diakses di penyimpanan Standar fclass atau 30 hari dialihkan ke IA.
- Transisi ke Arsip - Menginstruksikan manajemen siklus hidup kapan harus memindahkan file dari kelas penyimpanan Standar atau IA ke kelas penyimpanan Arsip, yang dioptimalkan biaya untuk data yang diakses hanya beberapa kali setiap tahun atau kurang. Secara default, file yang tidak diakses di kelas penyimpanan Standar selama 90 hari dialihkan ke kelas penyimpanan Arsip.
- Transisi ke Standar - Menginstruksikan manajemen siklus hidup apakah akan mentransisikan file dari kelas penyimpanan IA atau Arsip dan kembali ke kelas penyimpanan Standar ketika file diakses di kelas penyimpanan IA atau Arsip. Secara default, file tidak dipindahkan kembali ke kelas penyimpanan Standar, dan mereka tetap berada di kelas penyimpanan IA atau Arsip saat diakses.

Untuk kasus penggunaan yang sensitif terhadap kinerja yang menuntut kinerja latensi tercepat (seperti aplikasi yang bekerja dengan volume besar file kecil), pilih untuk mentransisikan file ke Penyimpanan standar Pada akses pertama.

Untuk informasi selengkapnya tentang mengonfigurasi kebijakan siklus hidup untuk sistem file, lihat.

[Mengkonfigurasi kebijakan siklus hidup](#)

Operasi sistem file untuk manajemen siklus hidup

Operasi sistem file untuk manajemen siklus hidup memiliki prioritas yang lebih rendah daripada operasi untuk beban kerja sistem file EFS. Waktu yang diperlukan untuk mentransisikan file masuk

ke atau keluar dari penyimpanan IA dan Arsip bervariasi tergantung pada ukuran file dan beban kerja sistem file. Misalnya, transisi jutaan file kecil mungkin memakan waktu lebih lama daripada mentransisikan lebih sedikit file yang lebih besar dengan ukuran penyimpanan total yang sama.

Metadata file, termasuk nama file, informasi kepemilikan, dan struktur direktori sistem file, selalu disimpan dalam Standar untuk membantu memastikan kinerja metadata yang konsisten.

Operasi metadata untuk sistem file di IA atau penyimpanan Arsip, seperti mencantumkan konten direktori, tidak dihitung sebagai akses file. Selama proses transisi konten file ke kelas penyimpanan IA atau Arsip, file tersebut disimpan di kelas penyimpanan Standar dan ditagih pada tingkat penyimpanan tersebut.

Semua operasi tulis ke file di kelas penyimpanan IA atau Arsip sistem file pertama kali ditulis ke kelas penyimpanan Standar, dan kemudian memenuhi syarat untuk dialihkan ke kelas penyimpanan yang berlaku setelah 24 jam.

Mengkonfigurasi kebijakan siklus hidup

Saat Anda membuat sistem file EFS yang memiliki pengaturan yang disarankan menggunakan Konsol Manajemen AWS, sistem file secara otomatis dikonfigurasi dengan konfigurasi siklus hidup default berikut:

- Transisi ke IA diatur ke 30 hari sejak akses terakhir.
- Transisi ke Arsip diatur ke 90 hari sejak akses terakhir.
- Transisi ke Standar diatur ke None.

Anda dapat mengubah kebijakan siklus hidup default saat membuat sistem file dengan setelan khusus menggunakan Konsol Manajemen AWS atau saat membuat sistem file menggunakan file. AWS CLI Sebagai alternatif, Anda dapat mengubah kebijakan setelah sistem file dibuat, seperti yang dijelaskan dalam prosedur berikut.

Menggunakan konsol

Anda dapat menggunakan Konsol Manajemen AWS untuk mengatur kebijakan siklus hidup untuk sistem file yang ada.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.

2. Pilih Sistem file untuk menampilkan daftar sistem file di akun Anda.
3. Pilih sistem file tempat Anda ingin mengubah kebijakan siklus hidup.
4. Pada halaman detail sistem file, di bagian Umum, pilih Edit. Halaman Edit ditampilkan.
5. Untuk manajemen Siklus Hidup, konfigurasi kebijakan siklus hidup:
 - Atur Transisi ke IA ke salah satu opsi yang tersedia. Untuk berhenti memindahkan file ke penyimpanan IA, pilih Tidak Ada.
 - Atur Transisi ke Arsip ke salah satu opsi yang tersedia. Untuk berhenti memindahkan file ke penyimpanan Arsip, pilih Tidak Ada.
 - Setel Transisi ke Akses pertama Standar ke Aktif untuk memindahkan file yang ada di penyimpanan IA ke penyimpanan standar saat diakses untuk operasi non-metadata.

Untuk berhenti memindahkan file dari IA atau Arsip ke penyimpanan Standar pada akses pertama, pilih Tidak Ada.

6. Pilih Simpan perubahan untuk menyimpan perubahan Anda.

Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI untuk mengatur atau memodifikasi kebijakan siklus hidup sistem file.

- Jalankan [put-lifecycle-configuration](#) AWS CLI perintah atau perintah [PutLifecycleConfiguration](#) API, tentukan ID sistem file dari sistem file yang Anda kelola manajemen siklus hidup.

```
$ aws efs put-lifecycle-configuration \
--file-system-id File-System-ID \
--lifecycle-policies "[{\"TransitionToIA\": \"AFTER_60_DAYS\"},
{ \"TransitionToPrimaryStorageClass\": \"AFTER_1_ACCESS\" }, { \"TransitionToArchive\":
\"AFTER_90_DAYS\" }]" \
--region us-west-2 \
--profile adminuser
```

Anda mendapatkan tanggapan berikut.

```
{
  "LifecyclePolicies": [
```

```
{
  "TransitionToIA": "AFTER_60_DAYS"
},
{
  "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
},
{
  "TransitionToArchive": "AFTER_90_DAYS"
}
]
```

Untuk menghentikan manajemen siklus hidup untuk sistem file yang ada (CLI)

- Jalankan `put-lifecycle-configuration` perintah yang menentukan ID sistem file dari sistem file yang Anda hentikan manajemen siklus hidup. Biarkan `--lifecycle-policies` properti tetap kosong.

```
$ aws efs put-lifecycle-configuration \
--file-system-id File-System-ID \
--lifecycle-policies \
--region us-west-2 \
--profile adminuser
```

Anda mendapatkan tanggapan berikut.

```
{
  "LifecyclePolicies": []
}
```

Pemantauan Amazon EFS

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon EFS dan AWS solusi Anda. Kami menyarankan Anda mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. Namun, sebelum Anda mulai memantau Amazon EFS, buat rencana pemantauan yang mencakup jawaban atas pertanyaan-pertanyaan berikut:

- Apa saja sasaran pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?
- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu saat terjadi kesalahan?

Langkah selanjutnya adalah menetapkan dasar untuk kinerja Amazon EFS normal di lingkungan Anda, dengan mengukur kinerja pada berbagai waktu dan dalam kondisi beban yang berbeda. Saat Anda memantau Amazon EFS, pertimbangkan untuk menyimpan data pemantauan historis. Data yang disimpan ini akan memberi Anda garis dasar untuk dibandingkan dengan data kinerja saat ini, mengidentifikasi pola kinerja normal dan anomali kinerja, dan merancang metode untuk mengatasi masalah.

Misalnya, dengan Amazon EFS, Anda dapat memantau throughput jaringan, I/O untuk operasi baca, tulis, dan metadata, koneksi klien, dan saldo kredit burst untuk sistem file Anda. Jika kinerja berada di luar garis dasar yang ditetapkan, Anda mungkin perlu mengubah ukuran sistem file Anda atau jumlah klien yang terhubung untuk mengoptimalkan sistem file untuk beban kerja Anda.

Untuk menetapkan baseline, Anda harus, setidaknya, memantau item-item berikut:

- Throughput jaringan sistem file Anda.
- Jumlah koneksi klien ke sistem file.
- Jumlah byte untuk setiap operasi sistem file, termasuk pembacaan data, penulisan data dan operasi metadata.

Topik

- [Alat-alat pemantauan](#)
- [Bagaimana Amazon EFS melaporkan sistem file dan ukuran objek](#)
- [Melihat ukuran kelas penyimpanan](#)
- [Memantau metrik dengan Amazon CloudWatch](#)
- [Mencatat panggilan Amazon EFS API dengan AWS CloudTrail](#)

Alat-alat pemantauan

AWS menyediakan berbagai alat yang dapat Anda gunakan untuk memantau Amazon EFS. Anda dapat mengonfigurasi beberapa alat ini untuk melakukan pemantauan untuk Anda, tetapi beberapa alat memerlukan intervensi manual. Sebaiknya Anda mengotomatisasi tugas pemantauan sebanyak mungkin.

Alat pemantauan otomatis

Anda dapat menggunakan alat pemantauan otomatis berikut untuk menonton Amazon EFS dan melaporkan ketika ada sesuatu yang salah:

- CloudWatch Alarm Amazon — Tonton satu metrik selama periode waktu yang Anda tentukan, dan lakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. Tindakannya adalah pemberitahuan yang dikirim ke topik Amazon Simple Notification Service (Amazon SNS) atau kebijakan Amazon EC2 Auto Scaling. CloudWatch alarm tidak memanggil tindakan hanya karena mereka berada dalam keadaan tertentu; negara harus telah berubah dan dipertahankan untuk sejumlah periode tertentu. Untuk informasi selengkapnya, lihat [Memantau metrik dengan Amazon CloudWatch](#).
- Amazon CloudWatch Logs — Pantau, simpan, dan akses file log Anda dari AWS CloudTrail atau sumber lain. Untuk informasi selengkapnya, lihat [Apa itu Amazon CloudWatch Logs?](#) di Panduan CloudWatch Pengguna Amazon.
- CloudWatch Acara Amazon — Cocokkan acara dan arahkan ke satu atau beberapa fungsi atau aliran target untuk membuat perubahan, menangkap informasi status, dan mengambil tindakan korektif. Untuk informasi selengkapnya, lihat [Apa itu CloudWatch Acara Amazon](#) di Panduan CloudWatch Pengguna Amazon.
- AWS CloudTrail Pemantauan Log - Bagikan file log antar akun, pantau file CloudTrail log secara real time dengan mengirimkannya ke CloudWatch Log, menulis aplikasi pemrosesan log di Java, dan validasi bahwa file log Anda tidak berubah setelah pengiriman oleh CloudTrail. Untuk informasi selengkapnya, lihat [Bekerja dengan file CloudTrail log](#) di Panduan AWS CloudTrail Pengguna.

Alat-alat pemantauan manual

Bagian penting lainnya dari pemantauan Amazon EFS melibatkan pemantauan secara manual item-item yang tidak dicakup oleh CloudWatch alarm Amazon. Amazon EFS CloudWatch,, dan Konsol Manajemen AWS dasbor lainnya memberikan at-a-glance pandangan tentang keadaan AWS lingkungan Anda. Kami menyarankan Anda juga memeriksa file log untuk sistem file.

- Dari konsol Amazon EFS, Anda dapat menemukan item berikut untuk sistem file Anda:
 - Ukuran terukur saat ini
 - Jumlah target mount
 - Status siklus hidup
- CloudWatch halaman rumah menunjukkan:
 - Alarm dan status saat ini
 - Grafik alarm dan sumber daya
 - Status kesehatan layanan

Selain itu, Anda dapat menggunakan CloudWatch untuk melakukan hal berikut:

- Buat [dasbor khusus](#) untuk memantau layanan yang Anda gunakan.
- Data metrik grafik untuk memecahkan masalah dan mengungkap tren.
- Cari dan telusuri semua metrik AWS sumber daya Anda.
- Membuat dan mengedit alarm untuk menerima notifikasi terkait masalah.

Bagaimana Amazon EFS melaporkan sistem file dan ukuran objek

Bagian berikut menjelaskan bagaimana Amazon EFS melaporkan ukuran sistem file, ukuran objek dalam sistem file, dan throughput sistem file.

Metering objek sistem file EFS

Objek yang dapat Anda lihat dalam sistem file EFS termasuk file biasa, direktori, tautan simbolik, dan file khusus (FIFOs dan soket). Masing-masing objek ini diukur untuk 2 kibibyte (KiB) metadata (untuk inode) dan satu atau lebih penambahan 4 KiB data. Daftar berikut menjelaskan ukuran data terukur untuk berbagai jenis objek sistem file:

- File reguler — Ukuran data terukur dari file biasa adalah ukuran logis dari file yang dibulatkan ke kenaikan 4-KiB berikutnya, kecuali bahwa itu mungkin kurang untuk file yang jarang.

File jarang adalah file yang datanya tidak ditulis ke semua posisi file sebelum ukuran logisnya tercapai. Untuk file yang jarang, dalam beberapa kasus penyimpanan aktual yang digunakan kurang dari ukuran logis yang dibulatkan ke kenaikan 4-KiB berikutnya. Dalam kasus ini, Amazon EFS melaporkan penyimpanan aktual yang digunakan sebagai ukuran data terukur.

- Direktori — Ukuran data terukur dari direktori adalah penyimpanan aktual yang digunakan untuk entri direktori dan struktur data yang menahannya, dibulatkan ke kenaikan 4-KiB berikutnya. Ukuran data terukur tidak termasuk penyimpanan aktual yang digunakan oleh data file.
- Tautan simbolis dan file khusus - Ukuran data terukur untuk objek ini selalu 4 KiB.

Saat Amazon EFS melaporkan ruang yang digunakan untuk objek, melalui NFSv4 `space_used` atribut.¹, itu menyertakan ukuran data terukur objek saat ini tetapi bukan ukuran metadatanya. Anda dapat menggunakan dua utilitas untuk mengukur penggunaan disk file, `du` dan `stat` utilitas. Berikut ini adalah contoh bagaimana menggunakan `du` utilitas pada file kosong yang mencakup `-k` opsi untuk mengembalikan output dalam kilobyte.

```
$ du -k file
4      file
```

Berikut contoh menunjukkan bagaimana menggunakan `stat` utilitas pada file kosong untuk mengembalikan penggunaan disk file.

```
$ /usr/bin/stat --format="%b%B" file | bc
4096
```

Untuk mengukur ukuran direktori, gunakan `stat` utilitas. Temukan `Blocks` nilainya, lalu kalikan nilai itu dengan ukuran blok. Berikut ini adalah contoh cara menggunakan `stat` utilitas pada direktori kosong:

```
$ /usr/bin/stat --format="%b%B" . | bc
4096
```

Ukuran terukur dari sistem file EFS

Ukuran terukur dari sistem file EFS mencakup jumlah ukuran semua objek saat ini di semua kelas penyimpanan EFS. Ukuran setiap objek dihitung dari pengambilan sampel representatif dari ukuran objek selama jam terukur, misalnya dari jam 8 pagi hingga 9 pagi.

File kosong memberikan kontribusi 6 KiB (2 KiB metadata+4 KiB data) ke ukuran terukur dari sistem file. Setelah pembuatan, sistem file memiliki satu direktori root kosong dan karenanya memiliki ukuran meteran 6 KiB.

Ukuran terukur dari sistem file tertentu menentukan penggunaan akun pemilik yang ditagih untuk sistem file tersebut selama jam itu.

Note

Ukuran terukur yang dihitung tidak mewakili snapshot yang konsisten dari sistem file pada waktu tertentu selama jam itu. Sebaliknya, ini mewakili ukuran objek yang ada dalam sistem file pada waktu yang bervariasi dalam setiap jam, atau mungkin satu jam sebelumnya. Ukuran ini dijumlahkan untuk menentukan ukuran meteran sistem file selama satu jam. Ukuran terukur dari sistem file dengan demikian akhirnya konsisten dengan ukuran terukur dari objek yang disimpan ketika tidak ada penulisan ke sistem file.

Anda dapat melihat ukuran terukur untuk sistem file EFS dengan cara berikut:

- Menggunakan [describe-file-systems](#) AWS CLI perintah dan operasi [DescribeFileSystem](#) API, responsnya mencakup yang berikut:

```
"SizeInBytes":{
    "Timestamp": 1403301078,
    "Value": 29313744866,
    "ValueInIA": 675432,
    "ValueInStandard": 29312741784
    "ValueInArchive": 327650
}
```

[Di mana ukuran terukur juga ValueInStandard digunakan untuk menentukan baseline I/O throughput dan burst rate Anda untuk sistem file menggunakan throughput Bursting.](#)

- Lihat StorageBytes CloudWatch metrik, yang menampilkan ukuran total data terukur di setiap kelas penyimpanan. Untuk informasi lebih lanjut tentang StorageBytes metrik, lihat [CloudWatch metrik untuk Amazon EFS](#).
- Jalankan df perintah di Linux pada prompt terminal dari instance EC2.

Jangan gunakan du perintah pada root sistem file untuk tujuan pengukuran penyimpanan karena respon tidak mencerminkan set data lengkap yang digunakan untuk metering sistem file Anda.

Note

Ukuran terukur juga `ValueInStandard` digunakan untuk menentukan baseline throughput I/O dan laju ledakan Anda. Untuk informasi selengkapnya, lihat [Throughput yang melonjak](#).

Mengukur kelas penyimpanan Akses dan Arsip yang Jarang

Kelas EFS Infrequent Access (IA) dan penyimpanan Arsip diukur dalam peningkatan 4 KiB dan memiliki biaya penagihan minimum per file 128 KiB. Metadata file IA dan Arsip (2 KiB per file) selalu disimpan dan diukur di kelas penyimpanan Standar. Support untuk file yang lebih kecil dari 128 KiB hanya tersedia untuk kebijakan siklus hidup yang diperbarui pada atau setelah 12:00 PM PT, 26 November 2023. Akses data untuk penyimpanan IA dan Arsip diukur dalam peningkatan 128 KiB.

Anda dapat menggunakan `StorageBytes` CloudWatch metrik untuk melihat ukuran data terukur di setiap kelas penyimpanan. Metrik ini juga menampilkan jumlah total byte yang dikonsumsi oleh pembulatan file kecil dalam kelas penyimpanan IA dan Arsip. Untuk informasi selengkapnya tentang melihat CloudWatch metrik, lihat [Mengakses CloudWatch metrik untuk Amazon EFS](#). Untuk informasi lebih lanjut tentang `StorageBytes` metrik, lihat [CloudWatch metrik untuk Amazon EFS](#).

Throughput pengukuran

Amazon EFS mengukur throughput untuk permintaan baca pada sepertiga tingkat operasi sistem I/O file lainnya. Misalnya, jika Anda menggerakkan 30 mebibyte per detik (MiBps) dari throughput baca dan tulis, bagian baca dihitung sebagai 10 MiBps dari throughput efektif, bagian tulis dihitung sebagai 30 MiBps, dan throughput terukur gabungan adalah 40. MiBps Hasil gabungan yang disesuaikan dengan tingkat konsumsi ini tercermin dalam `MeteredIOBytes` CloudWatch metrik.

Pengukuran Throughput elastis

Ketika mode throughput elastis diaktifkan untuk sistem file, Anda hanya membayar untuk jumlah metadata dan data yang dibaca dari atau ditulis ke sistem file. Sistem file EFS menggunakan pengukur mode throughput elastis dan metadata tagihan dibaca sebagai operasi baca dan metadata ditulis sebagai operasi tulis. Operasi metadata diukur dalam kenaikan 1 KiB setelah 4 KiB pertama. Operasi data diukur dalam kenaikan 1 KiB setelah 32 KiB pertama.

 Note

Meskipun throughput Elastic dirancang untuk menskalakan secara elastis dengan throughput Anda, kami merekomendasikan penerapan tata kelola yang tepat melalui metrik pemantauan dengan CloudWatch (TerukurIOBytes) dan peringatan penggunaan sebagai bagian dari praktik terbaik operasional Anda. Ini membantu Anda mempertahankan pemanfaatan sumber daya yang optimal dan tetap berada dalam parameter operasional yang direncanakan. Untuk informasi selengkapnya, lihat [Memantau metrik dengan Amazon CloudWatch](#).

Pengukuran Throughput yang disediakan

Untuk sistem file yang menggunakan mode throughput Provisioned, Anda hanya membayar untuk jumlah waktu throughput diaktifkan. Amazon EFS meteran sistem file dengan mode throughput yang disediakan diaktifkan setiap jam sekali. Untuk pengukuran saat mode throughput yang disediakan disetel kurang dari satu jam, Amazon EFS menghitung rata-rata waktu menggunakan presisi milidetik.

Melihat ukuran kelas penyimpanan

Anda dapat melihat berapa banyak data yang disimpan di setiap kelas penyimpanan sistem file Anda menggunakan konsol Amazon EFS, EFS API AWS CLI, atau EFS API.

Menggunakan konsol

Tab Ukuran terukur pada halaman detail sistem File menampilkan ukuran terukur saat ini dari sistem file dalam kelipatan biner byte (kibibytes, mebibytes, gibibytes, dan tebibytes). Metrik dipancarkan setiap 15 menit dan memungkinkan Anda melihat ukuran terukur sistem file Anda dari waktu ke waktu. Ukuran terukur menampilkan informasi berikut untuk ukuran penyimpanan sistem file:

- Ukuran total adalah ukuran (dalam byte biner) data yang disimpan dalam sistem file, termasuk semua kelas penyimpanan.
- Ukuran dalam Standar adalah ukuran (dalam byte biner) data yang disimpan di kelas penyimpanan EFS Standard.
- Ukuran di IA adalah ukuran (dalam byte biner) data yang disimpan di kelas penyimpanan EFS Infrequent Access. File yang lebih kecil dari 128KiB dibulatkan hingga 128KiB.
- Size in Archive adalah ukuran (dalam byte biner) data yang disimpan di kelas penyimpanan EFS Archive. File yang lebih kecil dari 128KiB dibulatkan hingga 128KiB.

Anda juga dapat melihat Storage bytes metrik pada tab Monitoring pada halaman detail sistem File di konsol Amazon EFS. Untuk informasi selengkapnya, lihat [Mengakses CloudWatch metrik untuk Amazon EFS](#).

Menggunakan AWS CLI

Anda dapat melihat berapa banyak data yang disimpan di setiap kelas penyimpanan sistem file Anda menggunakan AWS CLI atau EFS API. Lihat detail penyimpanan data dengan memanggil perintah `describe-file-systems` CLI (operasi API yang sesuai adalah [DescribeFileSystems](#)).

```
$ aws efs describe-file-systems \
--region us-west-2 \
--profile adminuser
```

Sebagai tanggapan, `ValueInIA` menampilkan ukuran terukur terakhir dalam byte di kelas penyimpanan Akses Jarang sistem file. `ValueInStandard` menampilkan ukuran terukur terakhir dalam byte di kelas penyimpanan Standar. `ValueInArchive` menampilkan ukuran terukur terakhir dalam byte di kelas penyimpanan Arsip. Jumlah dari tiga nilai sama dengan ukuran seluruh sistem file, yang ditampilkan di `Value`.

```
{
  "FileSystems":[
    {
      "OwnerId":"251839141158",
      "CreationToken":"MyFileSystem1",
      "FileSystemId":"fs-47a2c22e",
      "PerformanceMode" : "generalPurpose",
      "CreationTime": 1403301078,
      "LifecycleState":"created",
      "NumberOfMountTargets":1,
      "SizeInBytes":{"
        "Value": 29313746702,
        "ValueInIA": 675432,
        "ValueInStandard": 29312741784,
        "ValueInArchive":329486
      },
      "ThroughputMode": "elastic"
    }
  ]
}
```

Untuk cara tambahan untuk melihat dan mengukur penggunaan disk, lihat [Metering objek sistem file EFS](#).

Memantau metrik dengan Amazon CloudWatch

Anda dapat memantau sistem file menggunakan Amazon CloudWatch, yang mengumpulkan dan memproses data mentah dari Amazon EFS menjadi metrik hampir real-time yang dapat dibaca. Statistik ini dicatat untuk jangka waktu 15 bulan, sehingga Anda dapat memperoleh perspektif yang lebih baik tentang kinerja aplikasi atau layanan web Anda.

Secara default, data metrik Amazon EFS dikirim secara otomatis CloudWatch pada periode 1 menit, kecuali dicatat untuk beberapa metrik individual. Konsol Amazon EFS menampilkan serangkaian grafik berdasarkan data mentah dari Amazon CloudWatch. Bergantung pada kebutuhan Anda, Anda mungkin lebih suka mendapatkan data untuk sistem file Anda CloudWatch daripada grafik di konsol.

Untuk informasi selengkapnya tentang Amazon CloudWatch, lihat [Apa itu Amazon CloudWatch?](#) di Panduan CloudWatch Pengguna Amazon.

CloudWatch Metrik Amazon EFS dilaporkan sebagai byte mentah. Byte tidak dibulatkan baik ke desimal atau biner ganda unit.

Topik

- [CloudWatch metrik untuk Amazon EFS](#)
- [Mengakses CloudWatch metrik untuk Amazon EFS](#)
- [Menggunakan CloudWatch metrik untuk Amazon EFS](#)
- [Menggunakan matematika metrik dengan CloudWatch metrik](#)
- [Memantau keberhasilan dan kegagalan upaya pemasangan](#)
- [Membuat CloudWatch alarm untuk memantau Amazon EFS](#)

CloudWatch metrik untuk Amazon EFS

Metrik Amazon EFS menggunakan EFS namespace. Namespace AWS/EFS mencakup metrik berikut. Semua metrik kecuali untuk `TimeSinceLastSync` satu dimensi. `FileSystemId` ID sistem file dapat ditemukan di konsol Amazon EFS, dan itu mengambil bentuk `fs-abcdef0123456789a`.

TimeSinceLastSync

Menunjukkan jumlah waktu yang telah berlalu sejak sinkronisasi terakhir yang berhasil ke sistem file tujuan dalam konfigurasi replikasi. Setiap perubahan data pada sistem file sumber yang terjadi sebelum TimeSinceLastSync nilai telah berhasil direplikasi. Setiap perubahan pada sumber yang terjadi setelahnya TimeSinceLastSync mungkin tidak sepenuhnya direplikasi.

Metrik ini menggunakan dua dimensi:

- `FileSystemId` — ID sistem file sumber dalam konfigurasi replikasi.
- `DestinationFileSystemId` — ID sistem file tujuan dalam konfigurasi replikasi.

Unit: detik

Statistik yang valid: Minimum, Maximum, Average

PercentIOLimit

Menunjukkan seberapa dekat sistem file untuk mencapai I/O batas mode kinerja Tujuan Umum.

Unit: Persen

Statistik yang valid: Minimum, Maximum, Average

BurstCreditBalance

Jumlah kredit burst yang dimiliki sistem file. Kredit burst memungkinkan sistem file meledak ke tingkat throughput di atas tingkat dasar sistem file untuk jangka waktu tertentu.

MinimumStatistik adalah saldo kredit burst terkecil untuk setiap menit selama periode tersebut.

MaximumStatistik adalah saldo kredit burst terbesar untuk setiap menit selama periode tersebut.

AverageStatistik adalah saldo kredit burst rata-rata selama periode tersebut.

Unit: Byte

Statistik yang valid: Minimum, Maximum, Average

PermittedThroughput

Jumlah maksimum throughput yang dapat dikendarai oleh sistem file.

- Untuk sistem file yang menggunakan throughput Elastic, nilai ini mencerminkan throughput penulisan maksimum dari sistem file.
- Untuk sistem file yang menggunakan throughput Provisioned, jika jumlah data yang disimpan di kelas penyimpanan EFS Standard memungkinkan sistem file Anda untuk mendorong

throughput yang lebih tinggi daripada yang Anda berikan, metrik ini mencerminkan throughput yang lebih tinggi daripada jumlah yang disediakan.

- Untuk sistem file dalam throughput Bursting, nilai ini adalah fungsi dari ukuran sistem file dan `BurstCreditBalance`

`MinimumStatistik` adalah throughput terkecil yang diizinkan untuk setiap menit selama periode tersebut. `MaximumStatistik` adalah throughput tertinggi yang diizinkan untuk setiap menit selama periode tersebut. `AverageStatistik` adalah throughput rata-rata yang diizinkan selama periode tersebut.

 Note

Operasi baca diukur pada sepertiga tingkat operasi lainnya.

Unit: Byte per detik

Statistik yang valid: `Minimum`, `Maximum`, `Average`

MeteredIOBytes

Jumlah byte terukur untuk setiap operasi sistem file, termasuk pembacaan data, penulisan data, dan operasi metadata, dengan operasi baca didiskon sesuai dengan batas throughput.

Anda dapat membuat [ekspresi matematika CloudWatch metrik](#) yang `MeteredIOBytes` dibandingkan `PermittedThroughput` dengan. Jika nilai-nilai ini sama, maka Anda mengonsumsi seluruh jumlah throughput yang dialokasikan ke sistem file Anda. Dalam situasi ini, Anda dapat mempertimbangkan untuk mengubah mode throughput sistem file untuk mendapatkan throughput yang lebih tinggi.

`SumStatistik` adalah jumlah total byte terukur yang terkait dengan semua operasi sistem file. `MinimumStatistik` adalah ukuran operasi terkecil selama periode tersebut. `MaximumStatistik` adalah ukuran operasi terbesar selama periode tersebut. `AverageStatistik` adalah ukuran rata-rata operasi selama periode tersebut. `SampleCountStatistik` memberikan hitungan semua operasi.

Unit:

- Byte untuk `Minimum`, `Maximum`, `Average`, dan `Sum` statistik.
- Jumlah untuk `SampleCount`.

Statistik yang valid: Minimum, Maximum, Average, Sum, SampleCount

Jumlah IOBytes

Jumlah byte aktual untuk setiap operasi sistem file yang diproses oleh Amazon EFS, tanpa diskon baca. Jumlah ini mungkin berbeda dari jumlah aktual yang diminta oleh aplikasi Anda karena termasuk minimum. Angka ini mungkin juga lebih tinggi dari angka yang ditunjukkan PermittedThroughput.

Operasi data diukur pada 32 KiB dan operasi lainnya diukur pada 4 KiB. Setelah minimum, semua operasi diukur per KiB.

SumStatistik adalah jumlah total byte yang terkait dengan semua operasi sistem file. MinimumStatistik adalah ukuran operasi terkecil selama periode tersebut. MaximumStatistik adalah ukuran operasi terbesar selama periode tersebut. AverageStatistik adalah ukuran rata-rata operasi selama periode tersebut. SampleCountStatistik memberikan hitungan semua operasi.

Note

Untuk menghitung operasi rata-rata per detik untuk suatu periode, bagilah SampleCount statistik dengan jumlah detik dalam periode tersebut. Untuk menghitung throughput rata-rata (byte per detik) untuk suatu periode, bagi statistik Sum dengan jumlah detik dalam periode tersebut.

Unit:

- Byte untuk Minimum, MaximumAverage, dan Sum statistik.
- Jumlah untuk SampleCount.

Statistik yang valid: Minimum, Maximum, Average, Sum, SampleCount

DataReadIOBytes

Jumlah byte aktual untuk setiap operasi membaca sistem file.

SumStatistik adalah jumlah total byte yang terkait dengan operasi baca. MinimumStatistik adalah ukuran operasi baca terkecil selama periode tersebut. MaximumStatistik adalah ukuran operasi baca terbesar selama periode tersebut. AverageStatistik adalah ukuran rata-rata operasi baca selama periode tersebut. SampleCountStatistik memberikan hitungan operasi baca.

Unit:

- Byte untuk Minimum, Maximum, Average, dan Sum.
- Jumlah untuk SampleCount.

Statistik yang valid: Minimum, Maximum, Average, Sum, SampleCount

DataWriteIOBytes

Jumlah byte aktual untuk setiap operasi penulisan sistem file.

Statistik Sum adalah jumlah total byte yang terkait dengan operasi tulis. MinimumStatistik adalah ukuran operasi tulis terkecil selama periode tersebut. MaximumStatistik adalah ukuran operasi penulisan terbesar selama periode tersebut. AverageStatistik adalah ukuran rata-rata operasi tulis selama periode tersebut. SampleCountStatistik menyediakan hitungan operasi tulis.

Unit:

- Byte adalah unit untuk Minimum, MaximumAverage, dan Sum statistik.
- Jumlah untuk SampleCount.

Statistik yang valid: Minimum, Maximum, Average, Sum, SampleCount

MetadataIOBytes

Jumlah byte aktual untuk setiap operasi metadata.

SumStatistik adalah jumlah total byte yang terkait dengan operasi metadata. MinimumStatistik adalah ukuran operasi metadata terkecil selama periode tersebut. MaximumStatistik adalah ukuran operasi metadata terbesar selama periode tersebut. AverageStatistik adalah ukuran operasi metadata rata-rata selama periode tersebut. SampleCountStatistik menyediakan hitungan operasi metadata.

Unit:

- Byte adalah unit untuk Minimum, MaximumAverage, dan Sum statistik.
- Jumlah untuk SampleCount.

Statistik yang valid: Minimum, Maximum, Average, Sum, SampleCount

MetadataReadIOBytes

Jumlah byte aktual untuk setiap operasi membaca metadata.

SumStatistik adalah jumlah total byte yang terkait dengan operasi pembacaan metadata.

MinimumStatistik adalah ukuran operasi pembacaan metadata terkecil selama periode tersebut.

MaximumStatistik adalah ukuran operasi pembacaan metadata terbesar selama periode tersebut.

AverageStatistik adalah ukuran rata-rata operasi pembacaan metadata selama periode tersebut.

SampleCountStatistik memberikan hitungan operasi pembacaan metadata.

Unit:

- Byte adalah unit untuk **Minimum**, **MaximumAverage**, dan **Sum** statistik.
- Jumlah untuk **SampleCount**.

Statistik yang valid: **Minimum**, **Maximum**, **Average**, **Sum**, **SampleCount**

MetadataWriteIOBytes

Jumlah byte aktual untuk setiap operasi penulisan metadata.

SumStatistik adalah jumlah total byte yang terkait dengan operasi penulisan metadata.

MinimumStatistik adalah ukuran operasi penulisan metadata terkecil selama periode tersebut.

MaximumStatistik adalah ukuran operasi penulisan metadata terbesar selama periode tersebut.

AverageStatistik adalah ukuran rata-rata operasi penulisan metadata selama periode tersebut.

SampleCountStatistik menyediakan hitungan operasi penulisan metadata.

Unit:

- Byte adalah unit untuk **Minimum**, **MaximumAverage**, dan **Sum** statistik.
- Jumlah untuk **SampleCount**.

Statistik yang valid: **Minimum**, **Maximum**, **Average**, **Sum**, **SampleCount**

ClientConnections

Jumlah koneksi klien ke sistem file. Saat menggunakan klien standar, ada satu koneksi per instans Amazon EC2 yang dipasang.

Note

Untuk menghitung rata-rata **ClientConnections** periode lebih dari satu menit, bagilah **Sum** statistik dengan jumlah menit dalam periode tersebut.

Unit: Hitungan koneksi klien

Statistik valid: Sum

StorageBytes

Ukuran sistem file dalam byte, termasuk jumlah data yang disimpan di kelas penyimpanan EFS. Metrik ini dipancarkan CloudWatch setiap 15 menit.

StorageBytesMetrik memiliki dimensi sebagai berikut:

- **Total** adalah ukuran terukur (dalam byte) data yang disimpan dalam sistem file, di semua kelas penyimpanan. Untuk kelas penyimpanan EFS Infrequent Access (IA) dan EFS Archive, file yang lebih kecil dari 128KiB dibulatkan ke 128KiB.
- **Standard** adalah ukuran terukur (dalam byte) data yang disimpan di kelas penyimpanan EFS Standard.
- **IA** adalah ukuran sebenarnya (dalam byte) data yang disimpan di kelas penyimpanan EFS Infrequent Access.
- **IASizeOverhead** adalah perbedaan (dalam byte) antara ukuran aktual data di kelas penyimpanan EFS Infrequent Access (ditunjukkan dalam IA dimensi) dan ukuran terukur dari kelas penyimpanan, setelah membulatkan file kecil ke 128KiB.
- **Archive** adalah ukuran sebenarnya (dalam byte) data yang disimpan di kelas penyimpanan EFS Archive.
- **ArchiveSizeOverhead** adalah perbedaan (dalam byte) antara ukuran aktual data di kelas penyimpanan EFS Archive (ditunjukkan dalam Archive dimensi) dan ukuran terukur dari kelas penyimpanan, setelah membulatkan file kecil ke 128KiB.

Unit: Byte

Statistik yang valid: Minimum, Maximum, Average

Note

StorageBytes ditampilkan di halaman metrik sistem File konsol Amazon EFS menggunakan basis 1024 unit (kibibytes, mebibytes, gibibytes, dan tebibytes).

Mengakses CloudWatch metrik untuk Amazon EFS

Anda dapat melihat metrik Amazon EFS CloudWatch dalam beberapa cara:

- Di konsol Amazon EFS

- Di CloudWatch konsol
- Menggunakan CloudWatch CLI
- Menggunakan CloudWatch API

Untuk melihat CloudWatch metrik dan alarm (konsol Amazon EFS)

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.
2. Pilih Sistem file.
3. Pilih sistem file yang ingin Anda lihat CloudWatch metrik.
4. Pilih Monitoring untuk menampilkan halaman metrik sistem File.

Halaman metrik sistem File menampilkan seperangkat CloudWatch metrik default untuk sistem file. CloudWatch Alarm apa pun yang telah Anda konfigurasi juga ditampilkan dengan metrik ini. Untuk sistem file yang menggunakan mode I/O kinerja Maks, set metrik default mencakup saldo Burst Credit sebagai pengganti batas Persen IO. Anda dapat mengganti pengaturan default menggunakan kotak dialog Pengaturan metrik, diakses dengan membuka pengaturan.

 Note

Metrik pemanfaatan Throughput (%) bukan CloudWatch metrik; itu diturunkan menggunakan matematika CloudWatch metrik.

5. Anda dapat menyesuaikan cara metrik dan alarm ditampilkan menggunakan kontrol pada halaman metrik sistem File, sebagai berikut.
 - Alihkan mode Tampilan antara Time series atau Single value.
 - Tampilkan atau sembunyikan CloudWatch alarm apa pun yang dikonfigurasi untuk sistem file.
 - Pilih Lihat selengkapnya CloudWatch untuk melihat metrik di CloudWatch.
 - Pilih Tambahkan ke dasbor untuk membuka CloudWatch dasbor Anda dan menambahkan metrik yang ditampilkan.
 - Sesuaikan jendela waktu metrik yang ditampilkan dari 1 jam hingga 1 minggu.

Untuk melihat CloudWatch metrik dan alarm (konsol) CloudWatch

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.

2. Pada panel navigasi, silakan pilih Metrik.
3. Pilih namespace EFS.
4. (Opsional) Untuk melihat metrik, masukkan nama metrik dalam kotak pencarian.
5. (Opsional) Untuk memfilter berdasarkan dimensi, pilih FileSystemId.

Untuk mengakses metrik dari AWS CLI

- Gunakan perintah [list-metrics](#) dengan perintah namespace `--namespace "AWS/EFS"`. Untuk informasi selengkapnya, lihat [Referensi Perintah AWS CLI](#).

Untuk mengakses metrik dari API CloudWatch

- Panggil [GetMetricStatistics](#). Untuk informasi selengkapnya, lihat [Referensi Amazon CloudWatch API](#).

Menggunakan CloudWatch metrik untuk Amazon EFS

Metrik yang dilaporkan oleh Amazon EFS memberikan informasi yang dapat Anda analisis dengan berbagai cara. Daftar berikut menunjukkan beberapa penggunaan umum untuk metrik. Berikut ini adalah saran untuk memulai, bukan daftar komprehensif.

Bagaimana cara saya?	Metrik terkait
Bagaimana saya bisa menentukan throughput saya?	Anda dapat memantau Sum statistik harian <code>TotalIOBytes</code> metrik untuk melihat throughput Anda.
Bagaimana cara menentukan jumlah total data yang ditransfer?	Anda dapat memantau Sum statistik harian <code>MeteredIOBytes</code> untuk melihat total data Anda ditransfer.
Bagaimana cara melacak jumlah instans Amazon EC2	Anda dapat memantau Sum statistik <code>ClientConnections</code> metrik. Untuk menghitung rata-rata <code>ClientConnections</code> periode lebih dari satu menit, bagilah jumlah dengan jumlah menit dalam periode tersebut.

Bagaimana cara saya?	Metrik terkait
yang terhubung ke sistem file?	
Bagaimana saya bisa melihat saldo kredit burst saya?	Anda dapat melihat saldo Anda dengan memantau <code>BurstCreditBalance</code> metrik untuk sistem file Anda. Untuk informasi lebih lanjut tentang kredit meledak dan meledak, lihat Throughput yang melonjak .

Memantau kinerja throughput

CloudWatch Metrik untuk pemantauan throughput—`TotalIOBytes`, `ReadIOBytesWriteIOBytes`, dan `MetadataIOBytes` — mewakili throughput aktual yang Anda kendarai di sistem file Anda. Metrik `MeteredIOBytes` mewakili perhitungan keseluruhan throughput terukur yang Anda kendarai. Anda dapat menggunakan grafik `Throughput utilization (%)` di bagian Amazon EFS console Monitoring untuk memantau pemanfaatan throughput Anda. Jika Anda menggunakan CloudWatch dasbor kustom atau alat pemantauan lain, Anda dapat membuat [ekspresi matematika CloudWatch metrik](#) yang sebanding. `MeteredIOBytes PermittedThroughput`

`PermittedThroughput` mengukur jumlah throughput yang diizinkan untuk sistem file. Nilai ini didasarkan pada salah satu metode berikut:

- Untuk sistem file dalam throughput Elastic, nilai ini mencerminkan throughput penulisan maksimum dari sistem file.
- Untuk sistem file yang menggunakan throughput Provisioned, jika jumlah data yang disimpan di kelas penyimpanan EFS Standard memungkinkan sistem file Anda untuk mendorong throughput yang lebih tinggi daripada yang Anda berikan, metrik ini mencerminkan throughput yang lebih tinggi daripada jumlah yang disediakan.
- Untuk sistem file yang menggunakan throughput Bursting, nilai ini adalah fungsi dari ukuran sistem file dan `BurstCreditBalance`. Pantau `BurstCreditBalance` untuk memastikan bahwa sistem file Anda beroperasi pada kecepatan burst daripada kecepatan dasarnya. Jika saldo secara konsisten pada atau mendekati nol, pertimbangkan untuk beralih ke throughput Elastis atau throughput yang disediakan untuk mendapatkan throughput tambahan.

Ketika nilai untuk `MeteredIOBytes` dan `PermittedThroughput` sama, sistem file Anda mengkonsumsi semua throughput yang tersedia. Untuk sistem file yang menggunakan throughput `Provisioned`, Anda dapat menyediakan throughput tambahan.

Menggunakan matematika metrik dengan CloudWatch metrik

Menggunakan matematika metrik, Anda dapat menanyakan beberapa CloudWatch metrik Amazon dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini. Anda dapat memvisualisasikan deret waktu yang dihasilkan di CloudWatch konsol dan menambahkannya ke dasbor. Misalnya, Anda dapat menggunakan metrik Amazon EFS untuk mengambil jumlah sampel `DataRead` operasi dibagi 60. Hasilnya adalah jumlah rata-rata pembacaan per detik pada sistem file Anda untuk periode 1 menit tertentu. Untuk informasi selengkapnya tentang matematika metrik, lihat [Menggunakan ekspresi matematika dengan CloudWatch metrik](#) di Panduan CloudWatch Pengguna Amazon.

Berikut ini, temukan beberapa ekspresi matematika metrik yang berguna untuk Amazon EFS.

Topik

- [Matematika metrik: Throughput di MiBps](#)
- [Matematika metrik: Persen throughput](#)
- [Matematika metrik: Persentase pemanfaatan throughput yang diizinkan](#)
- [Matematika metrik: IOPS Throughput](#)
- [Matematika metrik: Persentase IOPS](#)
- [Matematika metrik: I/O Ukuran rata-rata dalam KiB](#)
- [Menggunakan matematika metrik melalui CloudFormation template untuk Amazon EFS](#)

Matematika metrik: Throughput di MiBps

Untuk menghitung throughput rata-rata (in MiBps) untuk periode waktu, pertama-tama pilih statistik penjumlahan (`DataReadIOBytes`, `DataWriteIOBytes`, `MetadataIOBytes`, atau `TotalIOBytes`). Kemudian ubah nilainya menjadi MiB, dan bagi dengan jumlah detik dalam periode tersebut.

Misalkan logika contoh Anda adalah ini: $(\text{jumlah TotalIOBytes} \div 1.048.576 \text{ (untuk dikonversi ke MiB)}) \div \text{detik dalam periode}$

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik yang dapat digunakan	Statistik	Periode
m1	• DataReadIOBytes • DataWriteIOBytes • MetadataIOBytes • TotalIOBytes	sum	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	$(m1/1048576)/PERIOD(m1)$

Matematika metrik: Persen throughput

Ekspresi matematika metrik ini menghitung persen dari keseluruhan throughput yang digunakan untuk jenis yang I/O berbeda—misalnya, persentase total throughput yang didorong oleh permintaan baca. Untuk menghitung persentase throughput keseluruhan yang digunakan oleh salah satu tipe I/O (DataReadIOBytes, DataWriteIOBytes, atau MetadataIOBytes) untuk jangka waktu tertentu, pertama-tama kalikan statistik jumlah masing-masing dengan 100. Kemudian bagi hasilnya dengan jumlah statistik TotalIOBytes untuk periode yang sama.

Misalkan logika contoh Anda adalah ini: $(\text{jumlah DataReadIOBytes} \times 100 \text{ (untuk mengkonversi ke persentase)}) \div \text{jumlah TotalIOBytes}$

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik atau metrik yang dapat digunakan	Statistik	Periode
m1	• TotalIOBytes	sum	1 menit

ID	Metrik atau metrik yang dapat digunakan	Statistik	Periode
m2	• DataReadI OBytes	sum	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	$(m2 * 100) / m1$

Matematika metrik: Persentase pemanfaatan throughput yang diizinkan

Untuk menghitung persentase pemanfaatan throughput yang diizinkan (MeteredIOBytes) untuk jangka waktu tertentu, pertama kalikan throughput dengan 100. MiBps Kemudian bagi hasilnya dengan statistik rata-rata yang PermittedThroughput dikonversi ke MiB untuk periode yang sama.

Misalkan logika contoh Anda adalah ini: (ekspresi matematika metrik untuk throughput dalam MiBps x 100 (untuk mengonversi ke persentase)) ÷ (jumlah PermittedThroughput ÷ 1.048.576 (untuk mengonversi byte ke MiB))

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik atau metrik yang dapat digunakan	Statistik	Periode
m1	MeteredIOBytes	sum	1 menit
m2	Permitted Throughput	rata-rata	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	$(m1/1048576) / \text{PERIODE} (m1)$
e2	$m2/1048576$
e3	$((e1) * 100) / (e2)$

Matematika metrik: IOPS Throughput

Untuk menghitung rata-rata operasi per detik (IOPS) untuk jangka waktu tertentu, bagilah statistik jumlah sampel (DataReadIOBytes, DataWriteIOBytes, MetadataIOBytes, atau TotalIOBytes) dengan jumlah detik dalam periode tersebut.

Misalkan logika contoh Anda adalah ini: jumlah sampel DataWriteIOBytes ÷ detik dalam periode tersebut

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik yang dapat digunakan	Statistik	Periode
m1	- DataReadIOBytes - DataWriteIOBytes - MetadataIOBytes - TotalIOBytes	jumlah sampel	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	$m1 / \text{PERIOD}(m1)$

Matematika metrik: Persentase IOPS

Untuk menghitung persentase IOPS per detik dari berbagai I/O jenis (`DataReadIOBytes`, `DataWriteIOBytes`, atau `MetadataIOBytes`) untuk jangka waktu tertentu, pertama-tama kalikan statistik jumlah sampel masing-masing dengan 100. Kemudian bagi nilai tersebut dengan statistik jumlah sampel `TotalIOBytes` untuk periode yang sama.

Misalkan logika contoh Anda adalah ini: (jumlah sampel `MetadataIOBytes` x 100 (untuk dikonversi ke persentase)) ÷ jumlah sampel `TotalIOBytes`

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik yang dapat digunakan	Statistik	Periode
m1	• <code>TotalIOBytes</code>	jumlah sampel	1 menit
m2	• <code>DataReadIOBytes</code> • <code>DataWriteIOBytes</code> • <code>MetadataIOBytes</code>	jumlah sampel	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	$(m2 * 100) / m1$

Matematika metrik: I/O Ukuran rata-rata dalam KiB

Untuk menghitung I/O ukuran rata-rata (dalam KiB) untuk suatu periode, bagilah statistik jumlah masing-masing untuk `DataReadIOBytes`, `DataWriteIOBytes`, atau `MetadataIOBytes` metrik dengan statistik jumlah sampel yang sama dari metrik tersebut.

Misalkan logika contoh Anda adalah ini: $(\text{jumlah DataReadIOBytes} \div 1.024 \text{ (untuk dikonversi ke KiB)}) \div \text{jumlah sampel DataReadIOBytes}$

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik yang dapat digunakan	Statistik	Periode
m1	• DataReadIOBytes • DataWriteIOBytes • MetadataIOBytes	sum	1 menit
m2	• DataReadIOBytes • DataWriteIOBytes • MetadataIOBytes	jumlah sampel	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	$(m1/1024)/m2$

Menggunakan matematika metrik melalui CloudFormation template untuk Amazon EFS

Anda juga dapat membuat ekspresi matematika metrik melalui CloudFormation template. Salah satu template tersebut tersedia untuk Anda unduh dan sesuaikan untuk digunakan dari [tutorial Amazon EFS](#) di GitHub. Untuk informasi selengkapnya tentang penggunaan CloudFormation templat, lihat [Bekerja dengan CloudFormation templat](#) di Panduan AWS CloudFormation Pengguna.

Memantau keberhasilan dan kegagalan upaya pemasangan

Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau dan melaporkan keberhasilan atau kegagalan upaya pemasangan untuk sistem file EFS Anda dari jarak jauh tanpa harus masuk ke klien. Gunakan prosedur berikut untuk mengonfigurasi instans EC2 Anda agar menggunakan CloudWatch Log untuk memantau keberhasilan atau kegagalan upaya pemasangan sistem file-nya.

Untuk mengaktifkan pemberitahuan keberhasilan atau kegagalan upaya pemasangan di CloudWatch log

1. Instal `amazon-efs-utils` pada instans EC2 yang memasang sistem file. Untuk informasi selengkapnya, lihat [Secara otomatis menginstal atau memperbarui klien Amazon EFS menggunakan AWS Systems Manager](#) atau [Menginstal klien Amazon EFS secara manual](#).
2. Instal `botocore` pada instans EC2 yang akan me-mount sistem file. Untuk informasi selengkapnya, lihat [Instalasi dan upgrade botocore](#).
3. Aktifkan fitur CloudWatch Log di `amazon-efs-utils`. Ketika Anda menggunakan AWS Systems Manager untuk menginstal dan mengkonfigurasi `amazon-efs-utils`, CloudWatch logging secara otomatis dilakukan untuk Anda. Saat Anda menginstal `amazon-efs-utils` paket secara manual, Anda harus memperbarui file `/etc/amazon/efs/efs-utils.conf` konfigurasi secara manual dengan menghapus komentar `# enabled = true` baris di bagian tersebut `cloudwatch-log`. Gunakan salah satu perintah berikut untuk mengaktifkan CloudWatch Log secara manual.

Untuk instans Linux:

```
sudo sed -i -e '/\[cloudwatch-log\]/{N;s/# enabled = true/enabled = true/}' /etc/amazon/efs/efs-utils.conf
```

Untuk instance macOS:

```
EFS_UTILS_VERSION= efs-utils-version  
sudo sed -i -e '/\[cloudwatch-log\]/{N;s/# enabled = true/enabled = true/;}' /usr/local/Cellar/amazon-efs-utils/${EFS_UTILS_VERSION}/libexec/etc/amazon/efs/efs-utils.conf
```

Untuk instance Mac2:

```
EFS_UTILS_VERSION= efs-utils-version
```

```
sudo sed -i -e '/\[cloudwatch-log\]/{N;s/# enabled = true/enabled = true/;}' /opt/homebrew/Cellar/amazon-efs-utils/${EFS_UTILS_VERSION}/libexec/etc/amazon/efs/efs-utils.conf
```

4. Secara opsional, Anda dapat mengonfigurasi nama grup CloudWatch Log dan mengatur hari penyimpanan log dalam `efs-utils.conf` file. Jika Anda ingin memiliki grup log terpisah CloudWatch untuk setiap sistem file yang dipasang, tambahkan `{fs_id}` ke akhir `log_group_name` bidang dalam `efs-utils.conf` file, sebagai berikut:

```
[cloudwatch-log]
log_group_name = /aws/efs/utils/{fs_id}
```

5. Lampirkan kebijakan `AmazonElasticFileSystemsUtils` AWS terkelola ke peran IAM yang telah Anda lampirkan ke instans EC2, atau ke AWS kredensial yang dikonfigurasi pada instans Anda. Anda dapat menggunakan Systems Manager untuk melakukan ini, untuk informasi selengkapnya, lihat [Langkah 1: Konfigurasi profil instans IAM dengan izin yang diperlukan](#).

Berikut ini adalah contoh entri log status upaya pemasangan:

```
Successfully mounted fs-12345678.efs.us-east-1.amazonaws.com at /home/ec2-user/efs
Mount failed, Failed to resolve "fs-01234567.efs.us-east-1.amazonaws.com"
```

Untuk melihat status pemasangan di CloudWatch Log

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Pilih Grup log di bilah navigasi sebelah kiri.
3. Pilih grup `/aws/efs/utillslog`. Anda akan melihat aliran log untuk setiap instance Amazon EC2 dan kombinasi sistem file EFS.
4. Pilih aliran log untuk melihat peristiwa log tertentu termasuk status keberhasilan atau kegagalan upaya pemasangan.

Membuat CloudWatch alarm untuk memantau Amazon EFS

Anda dapat membuat CloudWatch alarm yang mengirimkan pesan Amazon SNS saat alarm berubah status. Alarm mengawasi metrik tunggal selama periode waktu yang Anda tentukan. Alarm kemudian melakukan satu atau lebih tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu

selama sejumlah periode waktu. Tindakan ini adalah notifikasi yang dikirim ke topik Amazon SNS atau kebijakan Penskalaan Otomatis.

Alarm memanggil tindakan untuk perubahan status berkelanjutan saja. CloudWatch alarm tidak memanggil tindakan hanya karena mereka berada dalam keadaan tertentu; negara harus telah berubah dan dipertahankan untuk sejumlah periode tertentu.

Salah satu penggunaan CloudWatch alarm penting untuk Amazon EFS adalah untuk menegakkan enkripsi saat istirahat untuk sistem file Anda. Anda dapat mengaktifkan enkripsi saat istirahat untuk sistem file Amazon EFS saat dibuat. Untuk menerapkan encryption-at-rest kebijakan data untuk sistem file Amazon EFS, Anda dapat menggunakan Amazon CloudWatch dan AWS CloudTrail mendeteksi pembuatan sistem file dan memverifikasi bahwa enkripsi saat istirahat diaktifkan.

 Note

Saat ini, Anda tidak dapat menerapkan enkripsi saat transit.

Prosedur berikut menguraikan cara membuat alarm untuk Amazon EFS.

Menggunakan konsol

Untuk mengatur alarm menggunakan konsol CloudWatch

1. Masuk ke Konsol Manajemen AWS dan buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Pilih Buat Alarm. Proses ini meluncurkan Wizard Buat Alarm.
3. Pilih EFS Metrics dan gulir metrik Amazon EFS untuk menemukan metrik yang ingin Anda gunakan alarm. Untuk hanya menampilkan metrik Amazon EFS di kotak dialog ini, cari ID sistem file sistem file Anda. Pilih metrik untuk mengaktifkan sebuah alarm lalu pilih Selanjutnya.
4. Isi Nama, Deskripsi, Setiap kali nilai untuk metrik.
5. Jika Anda CloudWatch ingin mengirimkan Anda email saat status alarm tercapai, di bidang Setiap kali alarm ini:, pilih Status adalah ALARM. Di bidang Send notification to: (Kirim notifikasi ke:), pilih topik SNS yang ada. Jika Anda memilih Create topic (Buat topik), Anda dapat mengatur nama dan alamat email untuk daftar langganan email baru. Daftar ini disimpan dan muncul dalam bidang isian untuk alarm selanjutnya.

Note

Jika Anda menggunakan Buat topik untuk membuat topik Amazon SNS baru, alamat email harus diverifikasi sebelum menerima pemberitahuan. Email hanya dikirimkan saat alarm berada dalam status alarm. Jika perubahan status alarm ini terjadi sebelum alamat email diverifikasi, alamat email tidak akan menerima pemberitahuan.

6. Pada titik ini, area Pratinjau Alarm memberi Anda kesempatan untuk melihat pratinjau alarm yang akan Anda buat. Pilih Buat Alarm.

Menggunakan AWS CLI

Untuk mengatur alarm menggunakan AWS CLI

- Panggil [put-metric-alarm](#). Untuk informasi selengkapnya, lihat [Referensi Perintah AWS CLI](#).

Menggunakan CloudWatch API

Untuk mengatur alarm menggunakan CloudWatch API

- Panggil [PutMetricAlarm](#). Untuk informasi selengkapnya, lihat [Referensi Amazon CloudWatch API](#).

Mencatat panggilan Amazon EFS API dengan AWS CloudTrail

Amazon EFS terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon EFS. CloudTrail menangkap semua panggilan API untuk Amazon EFS sebagai peristiwa, termasuk panggilan dari konsol Amazon EFS dan dari panggilan kode ke operasi Amazon EFS API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk Amazon EFS. 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 dibuat ke Amazon EFS, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk informasi lebih lanjut, lihat [Apa itu AWS CloudTrail?](#) dalam AWS CloudTrail User Guide.

Informasi Amazon EFS di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di Amazon EFS, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat [Bekerja dengan riwayat CloudTrail acara](#).

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk Amazon EFS, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak berlaku untuk semua AWS Region s. Trail mencatat peristiwa dari semua Wilayah AWS AWS partisi 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 topik berikut di Panduan Pengguna AWS CloudTrail :

- [Membuat jejak untuk AWS akun Anda](#)
- [AWS integrasi layanan dengan log CloudTrail](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa Wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua Amazon EFS [Amazon EFS API](#) dicatat oleh CloudTrail. Misalnya, panggilan ke `CreateFileSystem`, `CreateMountTarget` dan `CreateTags` operasi menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan dibuat dengan pengguna root atau kredensial pengguna AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [elemen CloudTrail UserIdentity](#) di AWS CloudTrail Panduan Pengguna.

Memahami entri file log Amazon EFS

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.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateTags operasi ketika tag untuk sistem file dibuat dari konsol.

```
{
  "eventVersion": "1.06",
  "userIdentity": {
    "type": "Root",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-03-01T18:02:37Z"
      }
    }
  },
  "eventTime": "2017-03-01T19:25:47Z",
  "eventSource": "elasticfilesystem.amazonaws.com",
  "eventName": "CreateTags",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "console.amazonaws.com",
  "requestParameters": {
    "fileSystemId": "fs-00112233",
    "tags": [{
      "key": "TagName",
      "value": "AnotherNewTag"
    }]
  }
}
```



```

]
},
"responseElements": null,
"requestID": "dEXAMPLE-feb4-11e6-85f0-736EXAMPLE75",
"eventID": "eEXAMPLE-2d32-4619-bd00-657EXAMPLEe4",
"eventType": "AwsApiCall",
"apiVersion": "2015-02-01",
"recipientAccountId": "111122223333"
}

```

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan DeleteTags tindakan ketika tag untuk sistem file dihapus dari konsol.

```

{
  "eventVersion": "1.06",
  "userIdentity": {
    "type": "Root",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-03-01T18:02:37Z"
      }
    }
  },
  "eventTime": "2017-03-01T19:25:47Z",
  "eventSource": "elasticfilesystem.amazonaws.com",
  "eventName": "DeleteTags",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "console.amazonaws.com",
  "requestParameters": {
    "fileSystemId": "fs-00112233",
    "tagKeys": []
  },
  "responseElements": null,
  "requestID": "dEXAMPLE-feb4-11e6-85f0-736EXAMPLE75",
  "eventID": "eEXAMPLE-2d32-4619-bd00-657EXAMPLEe4",
  "eventType": "AwsApiCall",

```

```
"apiVersion": "2015-02-01",
"recipientAccountId": "111122223333"
}
```

Entri log untuk peran terkait layanan EFS

Peran terkait layanan Amazon EFS membuat panggilan API ke sumber daya.

AWS Anda akan melihat entri CloudTrail log dengan panggilan username :

AWSServiceRoleForAmazonElasticFileSystem yang dilakukan oleh peran terkait layanan EFS. Untuk informasi selengkapnya tentang EFS dan peran terkait layanan, lihat. [Menggunakan peran terkait layanan untuk Amazon EFS](#)

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateServiceLinkedRole tindakan saat Amazon EFS membuat peran AWSService RoleForAmazonElasticFileSystem terkait layanan.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/user1",
    "accountId": "111122223333",
    "accessKeyId": "A111122223333",
    "userName": "user1",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-10-23T22:45:41Z"
      }
    }
  },
  "invokedBy": "elasticfilesystem.amazonaws.com",
  "eventTime": "2019-10-23T22:45:41Z",
  "eventSource": "iam.amazonaws.com",
  "eventName": "CreateServiceLinkedRole",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "user_agent",
  "requestParameters": {
    "awSServiceName": "elasticfilesystem.amazonaws.com"
  }
}
```

```

    },
    "responseElements": {
      "role": {
        "assumeRolePolicyDocument":
"111122223333-10-111122223333Statement111122223333Action111122223333AssumeRole111122223333Effect%22%3A%20%22Allow%22%2C%20%22Principal%22%3A%20%7B%22Service%22%3A%20%5B%22elasticfilesystem.amazonaws.com%22%5D%7D%7D%5D%7D",
        "arn": "arn:aws:iam::111122223333:role/aws-service-role/elasticfilesystem.amazonaws.com/AWSServiceRoleForAmazonElasticFileSystem",
        "roleId": "111122223333",
        "createDate": "Oct 23, 2019 10:45:41 PM",
        "roleName": "AWSServiceRoleForAmazonElasticFileSystem",
        "path": "/aws-service-role/elasticfilesystem.amazonaws.com/"
      }
    },
    "requestID": "11111111-2222-3333-4444-abcdef123456",
    "eventID": "11111111-2222-3333-4444-abcdef123456",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}

```

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateNetworkInterface tindakan yang dibuat oleh peran AWSServiceRoleForAmazonElasticFileSystem terkait layanan, yang dicatat dalam `sessionContext`

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:sts::0123456789ab:assumed-role/AWSServiceRoleForAmazonElasticFileSystem/0123456789ab",
    "accountId": "0123456789ab",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::0123456789ab:role/aws-service-role/elasticfilesystem.amazonaws.com/AWSServiceRoleForAmazonElasticFileSystem",
        "accountId": "0123456789ab",
        "userName": "AWSServiceRoleForAmazonElasticFileSystem"
      },
      "webIdFederationData": {},

```

```

        "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-10-23T22:50:05Z"
        }
    },
    "invokedBy": "AWS Internal"
},
"eventTime": "2019-10-23T22:50:05Z",
"eventSource": "ec2.amazonaws.com",
"eventName": "CreateNetworkInterface",
"awsRegion": "us-east-1",
"sourceIPAddress": "elasticfilesystem.amazonaws.com",
"userAgent": "elasticfilesystem.amazonaws.com",
"requestParameters": {
    "subnetId": "subnet-71e2f83a",
    "description": "EFS mount target for fs-1234567 (fsmt-1234567)",
    "groupSet": {},
    "privateIpAddressesSet": {}
},
"responseElements": {
    "requestId": "0708e4ad-03f6-4802-b4ce-4ba987d94b8d",
    "networkInterface": {
        "networkInterfaceId": "eni-0123456789abcdef0",
        "subnetId": "subnet-12345678",
        "vpcId": "vpc-01234567",
        "availabilityZone": "us-east-1b",
        "description": "EFS mount target for fs-1234567 (fsmt-1234567)",
        "ownerId": "666051418590",
        "requesterId": "0123456789ab",
        "requesterManaged": true,
        "status": "pending",
        "macAddress": "00:bb:ee:ff:aa:cc",
        "privateIpAddress": "192.0.2.0",
        "privateDnsName": "ip-192-0-2-0.ec2.internal",
        "sourceDestCheck": true,
        "groupSet": {
            "items": [
                {
                    "groupId": "sg-c16d65b6",
                    "groupName": "default"
                }
            ]
        }
    },
    "privateIpAddressesSet": {

```

```

        "item": [
            {
                "privateIpAddress": "192.0.2.0",
                "primary": true
            }
        ],
        "tagSet": {}
    },
    "requestID": "11112222-3333-4444-5555-666666777777",
    "eventID": "aaaabbbb-1111-2222-3333-444444555555",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}

```

Entri log untuk otentikasi EFS

Otorisasi Amazon EFS untuk pancaran dan acara `NewClientConnection` klien NFS.

`UpdateClientConnection` CloudTrail Sebuah `NewClientConnection` peristiwa dipancarkan ketika koneksi diotorisasi segera setelah koneksi awal, dan segera setelah koneksi ulang. An `UpdateClientConnection` dipancarkan ketika koneksi diotorisasi ulang dan daftar tindakan yang diizinkan telah berubah. Acara ini juga dipancarkan ketika daftar baru tindakan yang diizinkan tidak disertakan. `ClientMount` Untuk informasi selengkapnya tentang otorisasi EFS, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#).

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan suatu `NewClientConnection` peristiwa.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:sts::0123456789ab:assumed-role/abcdef0123456789",
    "accountId": "0123456789ab",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE ",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::0123456789ab:role/us-east-2",

```

```

        "accountId": "0123456789ab",
        "userName": "username"
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-12-23T17:50:16Z"
    },
    "ec2RoleDelivery": "1.0"
}
},
"eventTime": "2019-12-23T18:02:12Z",
"eventSource": "elasticfilesystem.amazonaws.com",
"eventName": "NewClientConnection",
"awsRegion": "us-east-2",
"sourceIPAddress": "AWS Internal",
"userAgent": "elasticfilesystem",
"requestParameters": null,
"responseElements": null,
"eventID": "27859ac9-053c-4112-ae3-f3429719d460",
"readOnly": true,
"resources": [
    {
        "accountId": "0123456789ab",
        "type": "AWS::EFS::FileSystem",
        "ARN": "arn:aws:elasticfilesystem:us-east-2:0123456789ab:file-system/
fs-01234567"
    },
    {
        "accountId": "0123456789ab",
        "type": "AWS::EFS::AccessPoint",
        "ARN": "arn:aws:elasticfilesystem:us-east-2:0123456789ab:access-point/
fsap-0123456789abcdef0"
    }
],
"eventType": "AwsServiceEvent",
"recipientAccountId": "0123456789ab",
"serviceEventDetails": {
    "permissions": {
        "ClientRootAccess": true,
        "ClientMount": true,
        "ClientWrite": true
    }
},
"sourceIpAddress": "10.7.3.72"

```

```
}  
}
```

Entri file log Amazon EFS untuk sistem encrypted-at-rest file

Amazon EFS memberi Anda opsi untuk menggunakan enkripsi saat istirahat, enkripsi dalam perjalanan, atau keduanya, untuk sistem file Anda. Untuk informasi selengkapnya, lihat [Enkripsi data di Amazon EFS](#).

Amazon EFS mengirimkan [konteks Enkripsi](#) saat membuat permintaan AWS KMS API untuk menghasilkan kunci data dan mendekripsi data Amazon EFS. ID sistem file adalah konteks enkripsi untuk semua sistem file yang dienkripsi saat istirahat. Di `requestParameters` bidang entri CloudTrail log, konteks enkripsi terlihat mirip dengan yang berikut ini.

```
"EncryptionContextEquals": {}  
"aws:elasticfilesystem:filesystem:id" : "fs-4EXAMPLE"
```

Menggunakan AWS laporan penagihan dan penggunaan

Saat menggunakan Amazon EFS, Anda hanya membayar untuk sumber daya yang Anda gunakan dan tidak ada biaya minimum atau biaya persiapan. Seperti AWS layanan lainnya, sementara harga dikutip setiap bulan, penggunaan Anda diprorata per jam, dan Anda ditagih untuk penggunaan rata-rata Anda selama sebulan.

AWS memberikan laporan berikut untuk Amazon EFS:

- Laporan penagihan — Beberapa laporan yang memberikan tampilan tingkat tinggi dari semua aktivitas untuk AWS layanan yang Anda gunakan, termasuk Amazon EFS. AWS selalu menagih pemilik sistem file EFS untuk biaya Amazon EFS. Untuk informasi selengkapnya tentang laporan penagihan, lihat [AWS Billing laporan untuk Amazon EFS](#).
- Laporan penggunaan — Ringkasan aktivitas untuk layanan tertentu, dikumpulkan berdasarkan jam, hari, atau bulan. Anda dapat memilih jenis penggunaan dan pengoperasian yang akan disertakan. Anda juga dapat memilih cara pengumpulan data. Untuk informasi selengkapnya, lihat [AWS laporan penggunaan untuk Amazon EFS](#).

Topik berikut memberikan informasi tentang pelaporan penagihan dan penggunaan untuk Amazon EFS.

Topik

- [Menggunakan tag sistem file EFS alokasi biaya](#)
- [AWS Billing laporan untuk Amazon EFS](#)
- [AWS laporan penggunaan untuk Amazon EFS](#)
- [Memahami laporan penagihan dan penggunaan untuk Amazon EFS](#)

Menggunakan tag sistem file EFS alokasi biaya

Untuk melacak biaya penyimpanan atau kriteria lain untuk masing-masing proyek atau grup proyek, beri label pada sistem file EFS Anda menggunakan tag alokasi biaya. Tag alokasi biaya adalah pasangan kunci-nilai yang Anda kaitkan dengan sistem file. Setelah Anda mengaktifkan tag alokasi biaya, AWS gunakan tag untuk mengatur biaya sumber daya Anda pada laporan alokasi biaya Anda.

Laporan alokasi biaya mencantumkan AWS penggunaan akun Anda berdasarkan kategori produk dan pengguna akun terkait. Laporan ini berisi item baris yang sama dengan laporan penagihan

terperinci (lihat [Memahami laporan penagihan dan penggunaan untuk Amazon EFS](#)) dan kolom tambahan untuk kunci tanda Anda.

AWS menyediakan dua jenis tag alokasi biaya, tag yang AWS dihasilkan dan tag yang ditentukan pengguna. AWS mendefinisikan, membuat, dan menerapkan kunci `aws:createdBy` tag AWS yang dihasilkan untuk Anda setelah peristiwa Amazon EFS `CreateFileSystem`. Anda menentukan, membuat, dan menerapkan tag yang ditentukan pengguna ke sistem file Anda.

Anda harus mengaktifkan kedua jenis tanda secara terpisah di konsol Manajemen Penagihan dan Biaya sebelum tanda tersebut muncul di laporan penagihan Anda. Untuk informasi selengkapnya tentang tag AWS yang dihasilkan, lihat [Mengatur dan melacak AWS biaya menggunakan tag alokasi biaya](#) di AWS Billing Panduan Pengguna.

- Untuk informasi selengkapnya tentang membuat tag yang ditentukan pengguna untuk sumber daya EFS, lihat [Menandai sumber daya EFS](#)
- Untuk informasi selengkapnya tentang mengaktifkan tag yang ditentukan pengguna di konsol Billing and Cost Management, lihat [Mengaktifkan tag alokasi biaya yang ditentukan pengguna](#) di Panduan Pengguna.AWS Billing
- Untuk informasi selengkapnya tentang mengaktifkan tag AWS yang dihasilkan di konsol Billing and Cost Management, [lihat AWS Mengaktifkan tag alokasi biaya yang dihasilkan](#) di Panduan Pengguna.AWS Billing

Tanda alokasi biaya yang ditentukan pengguna

Tanda alokasi biaya yang ditentukan pengguna memiliki komponen berikut:

- Kunci tanda. Kunci tanda adalah nama tanda. Misalnya, pada tanda `project/Trinity`, `project` adalah kuncinya. Kunci tanda adalah string yang peka huruf besar/kecil yang dapat berisi 1 hingga 128 karakter Unicode.
- Nilai tanda. Nilai tanda adalah string yang diperlukan. Misalnya, dalam tanda `project/Trinity`, `Trinity` adalah nilainya. Nilai tanda adalah string yang peka huruf besar/kecil yang dapat berisi 0 hingga 256 karakter Unicode.

Untuk informasi selengkapnya tentang tag yang ditentukan pengguna, lihat [Menggunakan tag alokasi biaya yang ditentukan pengguna](#) di Panduan Pengguna.AWS Billing

AWS Billing laporan untuk Amazon EFS

Tagihan bulanan Anda AWS memisahkan informasi penggunaan dan biaya Anda berdasarkan AWS layanan dan fungsi. Ada beberapa laporan AWS penagihan yang tersedia: laporan bulanan, laporan alokasi biaya, dan laporan penagihan terperinci. Untuk informasi selengkapnya, lihat [Memahami tagihan Anda](#) di Panduan AWS Billing Pengguna.

Untuk laporan lebih rinci tentang penggunaan penyimpanan Amazon EFS Anda, gunakan Ekspor Data AWS untuk membuat ekspor AWS Cost and Usage Report (AWS CUR). Dengan AWS CUR 2.0, Anda memublikasikan laporan AWS penagihan ke bucket Amazon Simple Storage Service (Amazon S3) yang Anda miliki. Laporan penggunaan Amazon EFS mencantumkan operasi berdasarkan jenis penggunaan dan AWS Region. Untuk informasi selengkapnya, lihat [Laporan Biaya dan Penggunaan \(CUR\) 2.0](#) di Panduan Ekspor Data AWS Pengguna.

Sistem file EFS ditagih berdasarkan kategori penggunaan berikut.

- Penyimpanan (per gigabyte [GB]) — Jumlah data yang disimpan dalam sistem file EFS Anda per bulan, dengan harga yang berbeda tergantung pada kelas penyimpanan (EFS Standard, EFS Infrequent Access (IA) dan EFS Archive).
- Throughput (per GB atau mebibytes per detik [MiBps]) - Jumlah data yang ditransfer (operasi baca/tulis) atau jumlah yang melebihi jumlah yang Anda berikan, tergantung pada throughput sistem file, per bulan.
- Akses dan transisi data (per MiBps) — Jumlah data yang dibaca dan ditransfer ke kelas penyimpanan EFS Infrequent Access (IA) dan EFS Archive per bulan.
- Backup storage (per GB) — Jumlah ruang penyimpanan yang dikonsumsi oleh backup per bulan. Meningkatkan periode retensi cadangan Anda atau mengambil cadangan volume tambahan yang diprakarsai pengguna meningkatkan jumlah penyimpanan cadangan yang dikonsumsi sistem file Anda. Untuk informasi selengkapnya, lihat [Mencadangkan sistem file EFS](#)
- Replikasi (per GB) — Tingkat aktivitas penyimpanan dan baca, tulis, dan tingkatan yang berlaku untuk sistem file EFS sumber dan tujuan Anda, serta biaya yang berlaku untuk aktivitas transfer data di antara mereka, per bulan.

Untuk informasi mendetail tentang biaya Amazon EFS, lihat [Harga Amazon EFS](#).

Untuk informasi tentang memahami kode dan singkatan yang digunakan dalam laporan penagihan dan penggunaan Amazon EFS, lihat. [Memahami laporan penagihan dan penggunaan untuk Amazon EFS](#)

AWS laporan penggunaan untuk Amazon EFS

Saat mengunduh laporan penggunaan, Anda dapat memilih untuk menggabungkan data penggunaan berdasarkan jam, hari, atau bulan. Laporan penggunaan Amazon EFS mencantumkan operasi berdasarkan jenis penggunaan dan AWS Region. Untuk laporan lebih rinci tentang penggunaan penyimpanan Amazon EFS Anda, unduh laporan AWS penggunaan yang dibuat secara dinamis. Anda dapat memilih jenis penggunaan, operasi, dan periode waktu yang akan disertakan. Anda juga dapat memilih cara pengumpulan data. Untuk informasi selengkapnya tentang mengunduh laporan penggunaan, lihat [Mengunduh Laporan AWS Penggunaan](#) di Panduan Pengguna Ekspor AWS Data.

Laporan penggunaan Amazon EFS mencakup informasi berikut:

- Layanan - Amazon EFS
- UsageType – Salah satu nilai berikut:
 - Kode yang mengidentifikasi jenis penyimpanan
 - Kode yang mengidentifikasi jenis permintaan
 - Kode yang mengidentifikasi jenis transfer data
 - Kode yang mengidentifikasi mode throughput
 - Kode yang mengidentifikasi penggunaan cadangan

Untuk penjelasan rinci tentang jenis penggunaan Amazon EFS, lihat [Memahami laporan penagihan dan penggunaan untuk Amazon EFS](#).

- Sumber Daya — Nama sumber daya yang terkait dengan penggunaan yang terdaftar.
- StartTime – Waktu mulai pada hari saat penggunaan berlaku, dalam Waktu Universal Terkoordinasi (UTC).
- EndTime— Waktu akhir hari penggunaan berlaku, di Coordinated Universal Time (UTC).
- UsageValue— Salah satu nilai volume berikut. Unit pengukuran yang umum untuk data adalah gigabyte (GB). Namun, tergantung pada layanan dan laporannya, terabyte (TB) mungkin juga akan digunakan.
 - Jumlah permintaan selama periode waktu yang ditentukan
 - Jumlah data yang ditransfer
 - Jumlah data yang disimpan dalam jam tertentu

Untuk informasi tentang memahami kode dan singkatan yang digunakan dalam laporan penagihan dan penggunaan Amazon EFS, lihat. [Memahami laporan penagihan dan penggunaan untuk Amazon EFS](#)

Memahami laporan penagihan dan penggunaan untuk Amazon EFS

Laporan penagihan dan penggunaan Amazon EFS menggunakan kode dan singkatan. Untuk jenis penggunaan dalam tabel berikut, ganti *region* dengan singkatan dari daftar ini:

- APE1: Asia Pasifik (Hong Kong)
- APN1: Asia Pasifik (Tokyo)
- APN2: Asia Pasifik (Seoul)
- APN3: Asia Pasifik (Osaka)
- APS1: Asia Pasifik (Singapura)
- APS2: Asia Pasifik (Sydney)
- APS3: Asia Pasifik (Mumbai)
- APS4: Asia Pasifik (Jakarta)
- APS5: Asia Pasifik (Hyderabad)
- APS6: Asia Pasifik (Melbourne)
- CAN1: Kanada (Tengah)
- CAN2: Kanada Barat (Calgary)
- CNN1: Tiongkok (Beijing)
- CNW1: Tiongkok (Ningxia)
- AFS1: Afrika (Cape Town)
- EUC2: Eropa (Zurich)
- EUN1: Eropa (Stockholm)
- EUS2: Eropa (Spanyol)
- EUC1: Eropa (Frankfurt)
- EU: Eropa (Irlandia)
- EUS1: Eropa (Milan)

- EUW2: Eropa (London)
- EUW3: Eropa (Paris)
- ILC1: Israel (Israel)
- MEC1: Timur Tengah (UEA)
- MES1: Timur Tengah (Bahrain)
- SAE1: South America (São Paulo)
- UGW1: AWS GovCloud (AS-Barat)
- UGE1: AWS GovCloud (AS-Timur)
- USE1 (atau tidak ada awalan): US East (Virginia N.)
- USE2: AS Timur (Ohio)
- USW1: AS Barat (California N.)
- USW2: AS Barat (Oregon)

Untuk informasi tentang harga menurut AWS Region, lihat [Harga Amazon EFS](#).

Kolom pertama dalam tabel berikut mencantumkan jenis penggunaan yang muncul dalam laporan penagihan dan penggunaan Anda. Unit pengukuran yang umum untuk data adalah gigabyte (GB). Namun, untuk jumlah throughput yang disediakan, mebibytes per detik (MiBps) digunakan sebagai gantinya.

Jenis penggunaan untuk Amazon EFS

Jenis penggunaan	CloudWatch metrik	Unit	Granularitas	Deskripsi
<i>region</i> -ArchiveDataAccess-Bytes	N/A	GB	N/A	Jumlah data yang ditransfer ke atau diakses di penyimpanan EFS Archive.
<i>region</i> -ArchiveEarlyDelete-ByteHrs	N/A	GB-Mo	N/A	Penggunaan penyimpanan prorata untuk file yang dihapus dari penyimpanan EFS Archive sebelum komitmen minimum 90 hari berakhir.

Jenis penggunaa n	CloudWa h metrik	Unit	Granular tas	Deskripsi
<i>region</i> - ArchiveE arlyDelet e-SmallFi les	N/A	GB-Mo	N/A	Penggunaan penyimpanan prorata untuk file kecil (lebih kecil dari 128 KB) yang dihapus dari penyimpanan EFS Archive sebelum komitmen minimum 90 hari berakhir.
<i>region</i> - ArchiveT imedStora ge-ByteHrs	StorageE tes.Arch ive	GB-Mo	Per Jam	Jumlah GB-bulan data yang disimpan dalam penyimpanan EFS Archive.
<i>region</i> - ArchiveT imedStora ge-SmallF iles	StorageE tes.Arch iveSizeO rhead	GB-Mo	Per Jam	Jumlah GB-bulan benda kecil (lebih kecil dari 128 KB) disimpan di penyimpanan EFS Archive.
<i>region</i> - ETDataAc cess-Bytes	TotalIOB tes	GB	Per Jam	Jumlah data yang ditulis atau dibaca untuk file menggunakan Elastic throughput.
<i>region</i> - IADDataAc cess-Bytes	N/A	GB	Per Jam	Jumlah data yang ditransfer ke atau diakses di penyimpanan EFS Infrequent Access (IA).
<i>region</i> - IATimedS torage-By teHrs	StorageE tes.IA	GB-Mo	Per Jam	Jumlah GB-bulan data yang disimpan dalam penyimpanan EFS IA.
<i>region</i> - IATimedS torage-ET- ByteHrs	StorageE tes.IA	GB-Mo	Per Jam	Jumlah GB-bulan data untuk file yang menggunakan throughput Elastic disimpan dalam penyimpanan EFS IA.

Jenis penggunaa n	CloudWa h metrik	Unit	Granular tas	Deskripsi
<i>region</i> - IATimedS torage-ET- SmallFiles	StorageE tes.IASiz eOverhe	GB-Mo	Per Jam	Jumlah GB-bulan file kecil (lebih kecil dari 128 KB) disimpan dalam penyimpanan EFS IA, untuk file yang menggunakan throughput Elastic.
<i>region</i> - IATimedS torage-Sm allFiles	StorageE tes.IASiz eOverhe	GB-Mo	Per Jam	Jumlah GB-bulan file kecil (lebih kecil dari 128 KB) disimpan dalam penyimpanan EFS IA.
<i>region</i> - IATimedS torage-Z- ByteHrs	StorageE tes.IA	GB-Mo	Per Jam	Ukuran sebenarnya dari data yang disimpan dalam penyimpanan EFS IA untuk sistem file One Zone.
<i>region</i> - IATimedS torage-Z- SmallFiles	StorageE tes.IASiz eOverhe	GB-Mo	Per Jam	Jumlah GB-bulan file kecil (lebih kecil dari 128 KB) disimpan dalam penyimpanan EFS IA untuk sistem file One Zone.
<i>region</i> - IncludedTP- MiBpsHrs	Permitter Through t	MiBps- Mo	Per Jam	Jumlah maksimum throughput yang dapat dikendarai oleh sistem file.
<i>region</i> - Provisio nedTP-MiB psHrs	Permitter Through t	MiBps- Mo	Per Jam	Untuk sistem file yang menggunakan throughpu t Provisioned, throughput disediakan atau throughput maksimum yang diizinkan.
<i>region</i> - TimedSto rage-Byte Hrs	StorageE tes.Stanc ard	GB-Mo	Per Jam	Jumlah total aktual data yang disimpan dalam penyimpanan EFS Standard untuk sistem file regional.

Jenis penggunaa n	CloudW h metrik	Unit	Granular tas	Deskripsi
<i>region-</i> TimedSto rage-Z-By teHrs	StorageE tes.Stanc ard	GB-Mo	Per Jam	Jumlah total aktual data yang disimpan dalam penyimpanan EFS Standard untuk sistem file One Zone.

Catatan

- Penyimpanan EFS Archive dikenakan biaya untuk durasi penyimpanan minimum 90 hari, bahkan jika data dihapus sebelum 90 hari.
- Biaya tiering berlaku untuk data berjenjang dari penyimpanan EFS Standard ke penyimpanan EFS Infrequent Access (IA), dan dari penyimpanan EFS IA hingga penyimpanan EFS Standard.
- Penyimpanan EFS IA dan EFS Archive memiliki ukuran file minimum yang dapat ditagih 128 KiB. File yang lebih kecil dari 128 KiB dapat berjenjang ke kelas penyimpanan ini tetapi dikenakan biaya untuk 128 KiB penyimpanan pada tingkat kelas penyimpanan yang sesuai.
- Untuk sistem file yang menggunakan throughput Elastis, operasi metadata diukur dalam peningkatan 1 KiB setelah 4 KiB pertama dan operasi data diukur dalam peningkatan 1 KiB setelah 32 KiB pertama.
- Penggunaan penyimpanan dihitung dalam gigabyte biner (GB), di mana 1 GB adalah 2^{30} byte. Lihat [Bagaimana Amazon EFS melaporkan sistem file dan ukuran objek](#) detail tambahan tentang cara penggunaan penyimpanan dan throughput diukur untuk menghitung tagihan Amazon EFS Anda.
- GB-Bulan diperoleh dengan mengambil jumlah total GB-jam, menggabungkannya selama sebulan, dan kemudian membaginya dengan jumlah jam di bulan itu.

Melacak operasi dalam laporan penggunaan Anda

Operasi menjelaskan tindakan yang diambil pada sistem file EFS Anda dengan jenis penggunaan yang ditentukan. Operasi ditunjukkan dengan kode yang cukup jelas, seperti Read atau Delete. Untuk melihat tindakan mana pada sistem file Anda yang menghasilkan jenis penggunaan tertentu, gunakan kode ini. Saat membuat laporan penggunaan, Anda dapat memilih untuk menyertakan Semua Operasi, atau operasi tertentu, misalnya, Write, untuk dilaporkan.

Spesifikasi kinerja Amazon EFS

Bagian berikut memberikan ikhtisar kinerja Amazon EFS, dan menjelaskan bagaimana konfigurasi sistem file Anda memengaruhi dimensi kinerja utama. Kami juga memberikan beberapa tips dan rekomendasi penting untuk mengoptimalkan kinerja sistem file Anda.

Topik

- [Ringkasan kinerja](#)
- [Kelas penyimpanan](#)
- [Mode kinerja](#)
- [Mode throughput](#)
- [Kiat kinerja Amazon EFS](#)
- [Memecahkan masalah kinerja Amazon EFS](#)
- [Memecahkan masalah AMI dan kernel](#)

Ringkasan kinerja

Kinerja sistem file biasanya diukur dengan menggunakan dimensi latensi, throughput, dan Input/Output operasi per detik (IOPS). Performa Amazon EFS di seluruh dimensi ini bergantung pada konfigurasi sistem file Anda. Konfigurasi berikut memengaruhi kinerja sistem file Amazon EFS:

- Jenis sistem file - Regional atau Satu Zona
- Mode kinerja - Tujuan Umum atau Max I/O

Important

Mode I/O kinerja maks memiliki latensi per operasi yang lebih tinggi daripada mode kinerja Tujuan Umum. Untuk kinerja yang lebih cepat, kami sarankan untuk selalu menggunakan mode kinerja Tujuan Umum. Untuk informasi selengkapnya, lihat [Mode kinerja](#).

- Mode Throughput - Elastis, Disediakan, atau Meledak

Tabel berikut menguraikan spesifikasi kinerja untuk sistem file menggunakan mode kinerja Tujuan Umum dan kemungkinan kombinasi yang berbeda dari jenis sistem file dan mode throughput.

Spesifikasi kinerja untuk sistem file menggunakan mode kinerja Tujuan Umum

Konfigurasi penyimpanan dan throughput		Latensi ¹		IOPS maksimum		Throughput maksimum		
Jenis sistem file	Mode throughput	Baca operasi	Tulis operasi	Baca operasi	Tulis operasi	Per-file-s ystem membaca ²	Per-file-s ystem tulis ²	Baca/tulis per klien
Regional	Elastis	~1 milidetik (ms)	~ 2,7 milidetik (ms)	900.000—2.500.000 ³	500.000 ³	20—60 gibibyte per detik () GiBps	1—5 GiBps	1.500 mebibytes per detik () 4 MiBps
Regional	Disediakan	~ 1 ms	~ 2,7 ms	55.000	25.000	3—10 GiBps	1—3,33 GiBps	500 MiBps
Regional	Meledak	~ 1 ms	~ 2,7 ms	35.000	7.000	3—5 GiBps	1—3 GiBps	500 MiBps
Satu Zona	Elastis, Disediakan, Meledak	~ 1 ms	~ 1,6 ms	35.000	7.000	3 GiBps ⁵	1 GiBps ⁵	500 MiBps

1. Nilai latensi yang ditampilkan mewakili kinerja kasus terbaik dalam kondisi optimal. Hasil aktual dapat bervariasi berdasarkan faktor jaringan, beban kerja, dan sistem.
2. Throughput baca dan tulis maksimum tergantung pada. AWS Region Throughput yang melebihi throughput maksimum AWS Region seseorang membutuhkan peningkatan kuota throughput. Setiap permintaan untuk throughput tambahan dipertimbangkan case-by-case berdasarkan tim layanan Amazon EFS. Persetujuan mungkin tergantung pada jenis beban kerja Anda. Untuk mempelajari lebih lanjut tentang meminta kenaikan kuota, lihat. [Kuota Amazon EFS](#)

3. Secara default, sistem file yang menggunakan Elastic throughput drive maksimum 90.000 IOPS baca untuk data yang jarang diakses, 250.000 IOPS baca untuk data yang sering diakses, dan 50.000 IOPS tulis. Jika beban kerja Anda membutuhkan lebih banyak IOPS, maka Anda dapat meminta peningkatan hingga 10 kali lipat angka-angka ini. Untuk informasi selengkapnya, lihat [Kuota Amazon EFS yang dapat Anda tingkatkan](#). Rekomendasi tambahan berlaku untuk mencapai IOPS maksimum. Untuk informasi selengkapnya, lihat [the section called “Mengoptimalkan beban kerja yang menuntut throughput tinggi dan IOPS”](#).
4. Throughput baca dan tulis gabungan maksimum adalah 1.500 MiBps untuk sistem file yang menggunakan throughput Elastic dan dipasang menggunakan versi 2.0 atau yang lebih baru dari klien Amazon EFS (amazon-efs-utils versi) atau Amazon EFS CSI Driver (). aws-efs-csi-driver Untuk semua sistem file lainnya, batas throughputnya adalah 500 MiBps. Untuk informasi selengkapnya tentang klien Amazon EFS, lihat [Menginstal klien Amazon EFS](#).
5. Sistem file One Zone yang menggunakan throughput Bursting dapat mendorong jumlah throughput per-file-system baca dan tulis yang sama dengan sistem file Regional menggunakan throughput Bursting (pembacaan maksimum 5 GiBps untuk dibaca dan 3 untuk tulis). GiBps

Kelas penyimpanan

Kelas penyimpanan Amazon EFS dirancang untuk penyimpanan paling efektif tergantung pada kasus penggunaan.

- Kelas penyimpanan EFS Standard menggunakan penyimpanan solid state drive (SSD) untuk memberikan tingkat latensi terendah untuk file yang sering diakses. Kelas penyimpanan ini menyediakan latensi byte pertama serendah 1 milidetik untuk pembacaan dan 2,7 milidetik untuk penulisan.
- Kelas penyimpanan EFS Infrequent Access (IA) dan EFS Archive menyimpan data yang jarang diakses yang tidak memerlukan kinerja latensi yang diperlukan oleh data yang sering diakses. Kelas penyimpanan ini menyediakan latensi byte pertama puluhan milidetik.

Untuk informasi selengkapnya tentang kelas penyimpanan EFS, lihat [the section called “Kelas penyimpanan EFS”](#).

Mode kinerja

Amazon EFS menawarkan dua mode kinerja, Tujuan Umum dan Max I/O.

- Mode Tujuan Umum memiliki latensi per operasi terendah dan merupakan mode kinerja default untuk sistem file. Sistem file One Zone selalu menggunakan mode kinerja Tujuan Umum. Untuk kinerja yang lebih cepat, kami sarankan untuk selalu menggunakan mode kinerja Tujuan Umum.
- I/O Mode maks adalah tipe kinerja generasi sebelumnya yang dirancang untuk beban kerja yang sangat paralel yang dapat mentolerir latensi yang lebih tinggi daripada mode Tujuan Umum. I/O Mode maks tidak didukung untuk sistem file One Zone atau sistem file yang menggunakan throughput Elastic.

Important

Karena latensi per operasi yang lebih tinggi dengan Max I/O, sebaiknya gunakan mode kinerja Tujuan Umum untuk semua sistem file.

Untuk membantu memastikan bahwa beban kerja Anda tetap dalam batas IOPS yang tersedia untuk sistem file menggunakan mode kinerja Tujuan Umum, Anda dapat memantau metrik.

PercentIOLimit CloudWatch Untuk informasi selengkapnya, lihat [CloudWatch metrik untuk Amazon EFS](#).

Aplikasi dapat menskalakan IOPS mereka secara elastis hingga batas yang terkait dengan mode kinerja. Anda tidak ditagih secara terpisah untuk IOPS; mereka termasuk dalam akuntansi throughput sistem file. Setiap permintaan Network File System (NFS) dihitung sebagai 4 kilobyte (KB) throughput, atau permintaan dan ukuran respons aktual, mana yang lebih besar.

Mode throughput

Mode throughput sistem file menentukan throughput yang tersedia untuk sistem file Anda. Amazon EFS menawarkan tiga mode throughput: Elastic, Provisioned, dan Bursting. Throughput baca didiskon untuk memungkinkan Anda mendorong throughput baca yang lebih tinggi daripada throughput tulis. Throughput maksimum yang tersedia dengan setiap mode throughput tergantung pada AWS Region Untuk informasi selengkapnya tentang throughput sistem file maksimum di berbagai wilayah, lihat [Kuota Amazon EFS](#).

Sistem file Anda dapat mencapai 100% gabungan throughput baca dan tulisnya. Misalnya, jika sistem file Anda menggunakan 33% dari batas throughput bacanya, sistem file dapat secara bersamaan mencapai hingga 67% dari batas throughput penulisannya. Anda dapat memantau penggunaan throughput sistem file Anda dalam grafik pemanfaatan Throughput (%) pada halaman Detail Sistem File konsol. Untuk informasi selengkapnya, lihat [Memantau kinerja throughput](#).

Memilih mode throughput yang benar untuk sistem file

Memilih mode throughput yang benar untuk sistem file Anda bergantung pada persyaratan kinerja beban kerja Anda.

- **Throughput elastis (Disarankan)** — Gunakan throughput Elastis default ketika Anda memiliki beban kerja yang runcing atau tidak dapat diprediksi serta persyaratan kinerja yang sulit diprediksi, atau saat aplikasi Anda mendorong throughput dengan rasio 5% atau kurang. average-to-peak Untuk informasi selengkapnya, lihat [Throughput elastis](#).
- **Throughput yang disediakan** — Gunakan throughput yang disediakan jika Anda mengetahui persyaratan kinerja beban kerja Anda, atau saat aplikasi Anda mendorong throughput dengan rasio 5% atau lebih. average-to-peak Untuk informasi selengkapnya, lihat [Throughput yang disediakan](#).
- **Throughput meledak** — Gunakan throughput Bursting saat Anda menginginkan throughput yang berskala dengan jumlah penyimpanan di sistem file Anda.

Jika, setelah menggunakan throughput Bursting, Anda menemukan bahwa aplikasi Anda dibatasi throughput (misalnya, menggunakan lebih dari 80% throughput yang diizinkan atau Anda telah menggunakan semua kredit burst Anda), maka Anda harus menggunakan throughput Elastis atau Provisioned. Untuk informasi selengkapnya, lihat [Throughput yang melonjak](#).

Untuk informasi selengkapnya tentang metrik Amazon EFS, lihat [CloudWatch metrik untuk Amazon EFS](#).

Throughput elastis

Untuk sistem file yang menggunakan throughput Elastic, Amazon EFS secara otomatis menskalakan kinerja throughput ke atas atau ke bawah untuk memenuhi kebutuhan aktivitas beban kerja Anda. Throughput elastis adalah mode throughput terbaik untuk beban kerja yang runcing atau tidak dapat diprediksi dengan persyaratan kinerja yang sulit diprediksi, atau untuk aplikasi yang mendorong throughput rata-rata 5% atau kurang dari throughput puncak (rasio). average-to-peak

Karena kinerja throughput untuk sistem file dengan skala throughput Elastic secara otomatis, Anda tidak perlu menentukan atau menyediakan kapasitas throughput untuk memenuhi kebutuhan aplikasi Anda. Anda hanya membayar untuk jumlah metadata dan data yang dibaca atau ditulis, dan Anda tidak memperoleh atau menggunakan kredit burst saat menggunakan throughput Elastic.

 Note

Meskipun throughput Elastic dirancang untuk menskalakan secara elastis dengan throughput Anda, kami merekomendasikan penerapan tata kelola yang tepat melalui metrik pemantauan dengan CloudWatch (TerukurIOBytes) dan peringatan penggunaan sebagai bagian dari praktik terbaik operasional Anda. Ini membantu Anda mempertahankan pemanfaatan sumber daya yang optimal dan tetap berada dalam parameter operasional yang direncanakan. Untuk informasi selengkapnya, lihat [Memantau metrik dengan Amazon CloudWatch](#).

Untuk informasi tentang batas throughput Elastic Per-wilayah, lihat. [Kuota Amazon EFS yang dapat Anda tingkatkan](#)

Throughput yang disediakan

Dengan Provisioned throughput, Anda menentukan tingkat throughput yang dapat didorong oleh sistem file secara independen dari ukuran sistem file atau saldo kredit burst. Gunakan throughput yang disediakan jika Anda mengetahui persyaratan kinerja beban kerja Anda, atau jika aplikasi Anda mendorong throughput sebesar 5% atau lebih dari rasio. average-to-peak

Untuk sistem file yang menggunakan throughput Provisioned, Anda dikenakan biaya untuk jumlah throughput yang diaktifkan untuk sistem file. Jumlah throughput yang ditagih dalam sebulan didasarkan pada throughput yang disediakan melebihi throughput dasar yang disertakan sistem file Anda dari penyimpanan Standar, hingga batas throughput dasar Bursting yang berlaku di. AWS Region

Jika throughput dasar sistem file melebihi jumlah throughput Provisioned, maka secara otomatis menggunakan throughput Bursting yang diizinkan untuk sistem file (hingga batas throughput dasar Bursting yang berlaku di dalamnya). AWS Region

Untuk informasi tentang batas throughput yang disediakan per wilayah, lihat. [Kuota Amazon EFS yang dapat Anda tingkatkan](#)

Throughput yang melonjak

Throughput yang meledak direkomendasikan untuk beban kerja yang memerlukan throughput yang diskalakan dengan jumlah penyimpanan di sistem file Anda. Dengan throughput Bursting, throughput dasar sebanding dengan ukuran sistem file di kelas penyimpanan Standar, dengan kecepatan 50 per

KiBps setiap GiB penyimpanan. Kredit burst bertambah ketika sistem file mengkonsumsi di bawah tingkat throughput dasarnya, dan dikurangkan ketika throughput melebihi tingkat dasar.

Ketika kredit burst tersedia, sistem file dapat mendorong throughput hingga 100 MiBps per TiB dalam penyimpanan Standar (50 KiBps per GiB), hingga AWS Region batas, dengan minimum 100. MiBps. Jika tidak ada kredit burst yang tersedia, sistem file dapat mendorong hingga 50 MiBps per TiB penyimpanan, dengan minimal 1. MiBps.

Untuk informasi tentang throughput Bursting Per-wilayah, lihat. [General resource quotas that cannot be changed](#)

Memahami kredit ledakan Amazon EFS

Dengan throughput Bursting, setiap sistem file menghasilkan kredit burst dari waktu ke waktu pada tingkat dasar yang ditentukan oleh ukuran sistem file yang disimpan di kelas penyimpanan EFS Standard. Tingkat dasar adalah 50 MiBps per tebibyte [TiB] penyimpanan (setara dengan 50 KiBps per GiB penyimpanan). Amazon EFS mengukur operasi baca hingga sepertiga tingkat operasi tulis, memungkinkan sistem file untuk mendorong tingkat dasar hingga 150 per KiBps GiB throughput baca, atau 50 per GiB throughput tulis. KiBps.

Sistem file dapat mendorong throughput pada tingkat meteran baseline secara terus menerus. Sistem file mengakumulasi kredit burst setiap kali tidak aktif atau mendorong throughput di bawah tingkat meteran baseline. Kredit burst yang terakumulasi memberi sistem file kemampuan untuk mendorong throughput di atas tingkat dasarnya.

Misalnya, sistem file dengan 100 GiB data terukur di kelas penyimpanan Standar memiliki throughput dasar 5. MiBps. Selama periode tidak aktif 24 jam, sistem file menghasilkan kredit senilai 432.000 MiB ($5 \text{ MiB} \times 86.400 \text{ detik} = 432.000 \text{ MiB}$), yang dapat digunakan untuk meledak pada MiBps 100 selama 72 menit ($432.000 \text{ MiB} \div 100 = 72 \text{ menit}$). MiBps.

Sistem file yang lebih besar dari 1 TiB selalu dapat meledak hingga 50 persen dari waktu jika mereka tidak aktif selama 50 persen sisanya.

Tabel berikut memberikan contoh perilaku meledak.

Ukuran sistem file	Throughput meledak	Throughput dasar
100 GiB data terukur dalam	Burst menjadi 300 (MiBps) read-only hingga 72 menit per hari, atau	Berkendara hingga 15 MiBps read-only terus menerus

Ukuran sistem file	Throughput meledak	Throughput dasar
penyimpanan Standar	Burst hingga 100 MiBps write-only hingga 72 menit per hari	Berkendara hingga 5 MiBps tulis saja terus menerus
1 TiB data terukur dalam penyimpanan Standar	- Burst menjadi 300 MiBps read-only selama 12 jam per hari, atau - Burst to 100 MiBps write-only selama 12 jam per hari	- Dorong 150 MiBps read-only terus menerus - Dorong 50 hanya MiBps tulis terus menerus
10 TiB data terukur dalam penyimpanan Standar	- Burst menjadi 3 GiBps read-only selama 12 jam per hari, atau - Burst to 1 GiBps write-only selama 12 jam per hari	- Drive 1.5 GiBps read-only terus menerus - Drive 500 hanya MiBps tulis terus menerus
Umumnya, sistem file yang lebih besar	- Burst menjadi 300 MiBps read-only per TiB penyimpanan selama 12 jam per hari, atau - Burst hingga 100 MiBps write-only per TiB penyimpanan selama 12 jam per hari	- Dorong 150 MiBps read-only per TiB penyimpanan terus menerus - Dorong 50 MiBps write-only per TiB penyimpanan terus menerus

Note

Amazon EFS menyediakan throughput terukur 1 MiBps untuk semua sistem file, bahkan jika tingkat dasar lebih rendah.

Ukuran sistem file yang digunakan untuk menentukan baseline dan burst rate adalah ukuran `ValueInStandard` terukur yang tersedia melalui operasi API. [DescribeFileSystems](#)

Sistem file dapat memperoleh kredit hingga saldo kredit maksimum 2,1 TiB untuk sistem file yang lebih kecil dari 1 TiB, atau 2,1 TiB per TiB yang disimpan untuk sistem file yang lebih besar dari 1 TiB. Perilaku ini berarti bahwa sistem file dapat mengumpulkan kredit yang cukup untuk meledak hingga 12 jam terus menerus.

Pembatasan pada pengalihan throughput dan perubahan jumlah yang disediakan

Anda dapat mengganti mode throughput sistem file yang ada dan mengubah jumlah throughput. Namun, setelah mengalihkan mode throughput ke throughput yang disediakan atau mengubah jumlah throughput yang disediakan, tindakan berikut dibatasi untuk periode 24 jam:

- Beralih dari mode throughput yang disediakan ke mode throughput Elastis atau Bursting.
- Mengurangi jumlah throughput yang disediakan.

Kiat kinerja Amazon EFS

Saat menggunakan Amazon EFS, ingatlah kiat kinerja berikut.

I/O Ukuran rata-rata

Sifat terdistribusi Amazon EFS memungkinkan tingkat ketersediaan, daya tahan, dan skalabilitas yang tinggi. Arsitektur terdistribusi ini menghasilkan overhead latensi kecil untuk setiap operasi file. Karena latensi per operasi ini, throughput keseluruhan umumnya meningkat seiring dengan meningkatnya I/O ukuran rata-rata, karena overhead diamortisasi pada jumlah data yang lebih besar.

Mengoptimalkan beban kerja yang menuntut throughput tinggi dan IOPS

Untuk beban kerja yang memerlukan throughput tinggi dan IOPS, gunakan sistem file Regional yang dikonfigurasi dengan mode kinerja Tujuan Umum dan throughput Elastis.

Note

Untuk mencapai IOPS baca maksimum untuk data yang sering diakses, sistem file harus menggunakan throughput Elastic.

Untuk mencapai tingkat kinerja tertinggi, Anda harus memanfaatkan paralelisasi dengan mengonfigurasi aplikasi atau beban kerja Anda sebagai berikut.

1. Mendistribusikan beban kerja secara merata di semua klien dan direktori, dengan setidaknya jumlah direktori yang sama dengan jumlah klien yang digunakan.

2. Minimalkan perdebatan dengan menyelaraskan thread individual ke kumpulan data atau file yang berbeda.
3. Distribusikan beban kerja di 10 klien NFS atau lebih, dengan setidaknya 64 utas per klien dalam satu target pemasangan.

Koneksi simultan

Anda dapat memasang sistem file Amazon EFS di hingga ribuan Amazon EC2 dan instans AWS komputasi lainnya secara bersamaan. Anda dapat mendorong tingkat throughput yang lebih tinggi pada sistem file secara agregat di seluruh instance komputasi jika Anda dapat memparalelkan aplikasi Anda di lebih banyak instance.

Model permintaan

Jika Anda mengaktifkan penulisan asinkron ke sistem file Anda, operasi penulisan yang tertunda akan di-buffer pada instans Amazon EC2 sebelum ditulis ke Amazon EFS secara asinkron. Penulisan asinkron biasanya memiliki latensi yang lebih rendah. Saat melakukan penulisan asinkron, kernel menggunakan memori tambahan untuk melakukan cache.

Sistem file yang telah mengaktifkan penulisan sinkron, atau yang membuka file menggunakan opsi yang melewati cache (misalnya, `O_DIRECT`), mengeluarkan permintaan sinkron ke Amazon EFS. Setiap operasi melewati perjalanan pulang pergi antara klien dan Amazon EFS.

Note

Model permintaan pilihan Anda telah mengorbankan konsistensi (jika Anda menggunakan beberapa instans Amazon EC2) dan kecepatan. Menggunakan penulisan sinkron memberikan peningkatan konsistensi data dengan menyelesaikan setiap transaksi permintaan tulis sebelum memproses permintaan berikutnya. Menggunakan penulisan asinkron memberikan peningkatan throughput dengan buffering operasi penulisan yang tertunda.

Pengaturan pemasangan klien NFS

Verifikasi bahwa Anda menggunakan opsi pemasangan yang disarankan seperti yang diuraikan di dalam [Memasang sistem file EFS](#) dan di [Pertimbangan pemasangan untuk Linux](#).

Saat memasang sistem file di instans Amazon EC2, Amazon EFS mendukung protokol Network File System versi 4.0 dan 4.1 (NFSv4). NFSv4.1 memberikan kinerja yang lebih baik untuk operasi baca file kecil paralel (lebih dari 10.000 file per detik) dibandingkan dengan NFSv4.0 (kurang dari 1.000 file per detik). Untuk instans macOS Amazon EC2 yang menjalankan macOS Big Sur, hanya 0,0 yang didukung. NFSv4

Jangan gunakan opsi pemasangan berikut:

- `noac,actimeo=0,acregmax=0,acdirmax=0` — Opsi ini menonaktifkan cache atribut, yang memiliki dampak kinerja yang sangat besar.
- `lookupcache=pos, lookupcache=none` — Opsi ini menonaktifkan cache pencarian nama file, yang memiliki dampak yang sangat besar pada kinerja.
- `fsc` — Opsi ini memungkinkan caching file lokal, tetapi tidak mengubah koherensi cache NFS, dan tidak mengurangi latensi.

Note

Saat Anda memasang sistem file Anda, pertimbangkan untuk meningkatkan ukuran buffer baca dan tulis untuk klien NFS Anda menjadi 1 MB.

Mengoptimalkan kinerja file kecil

Anda dapat meningkatkan kinerja file kecil dengan meminimalkan pembukaan kembali file, meningkatkan paralelisme, dan menggabungkan file referensi jika memungkinkan.

- Minimalkan jumlah perjalanan pulang pergi ke server.

Jangan menutup file secara tidak perlu jika Anda membutuhkannya nanti dalam alur kerja. Menjaga deskriptor file tetap terbuka memungkinkan akses langsung ke salinan lokal di cache. Operasi buka, tutup, dan metadata file umumnya tidak dapat dibuat secara asinkron atau melalui pipa.

Saat membaca atau menulis file kecil, dua perjalanan pulang pergi tambahan itu signifikan.

Setiap perjalanan pulang pergi (file terbuka, tutup file) dapat memakan waktu sebanyak membaca atau menulis megabyte data massal. Ini lebih efisien untuk membuka file input atau output sekali, di awal pekerjaan komputasi Anda, dan menahannya terbuka untuk seluruh panjang pekerjaan.

- Gunakan paralelisme untuk mengurangi dampak waktu pulang pergi.

- Bundel file referensi dalam .zip file. Beberapa aplikasi menggunakan satu set besar file referensi kecil, sebagian besar hanya-baca. Menggabungkan ini dalam .zip file memungkinkan Anda membaca banyak file dengan satu perjalanan pulang-pergi terbuka.

.zipFormat ini memungkinkan akses acak ke file individual.

Mengoptimalkan kinerja direktori

Saat melakukan listing (ls) pada direktori yang sangat besar (lebih dari 100k file) yang sedang dimodifikasi secara bersamaan, klien Linux NFS dapat hang, tidak mengembalikan respons. Masalah ini diperbaiki di kernel 5.11, yang telah di-porting ke kernel Amazon Linux 2 4.14, 5.4, dan 5.10.

Kami menyarankan agar jumlah direktori di sistem file Anda kurang dari 10.000, jika memungkinkan. Gunakan subdirektori bersarang sebanyak mungkin.

Saat mencantumkan direktori, hindari mendapatkan atribut file jika tidak diperlukan, karena tidak disimpan di direktori itu sendiri.

Mengoptimalkan ukuran read_ahead_kb NFS

read_ahead_kbAtribut NFS mendefinisikan jumlah kilobyte untuk kernel Linux untuk dibaca ke depan atau prefetch selama operasi baca berurutan.

Untuk versi kernel Linux sebelum 5.4.*, read_ahead_kb nilainya ditetapkan dengan mengalikan NFS_MAX_READAHEAD dengan nilai untuk rsize (klien mengonfigurasi ukuran buffer baca yang diatur dalam opsi pemasangan). Saat menggunakan [opsi pemasangan yang disarankan](#), rumus ini disetel read_ahead_kb ke 15 MB.

Note

Dimulai dengan kernel Linux versi 5.4.*, klien Linux NFS menggunakan read_ahead_kb nilai default 128 KB. Kami merekomendasikan untuk meningkatkan nilai ini menjadi 15 MB.

Amazon EFS mount helper yang tersedia dalam amazon-efs-utils versi 1.33.2 dan yang lebih baru secara otomatis memodifikasi read_ahead_kb nilainya menjadi sama dengan 15*rsize, atau 15 MB, setelah memasang sistem file.

Untuk kernel Linux 5.4 atau yang lebih baru, jika Anda tidak menggunakan mount helper untuk me-mount sistem file Anda, pertimbangkan pengaturan manual read_ahead_kb ke 15 MB untuk

meningkatkan kinerja. Setelah memasang sistem file, Anda dapat mengatur ulang `read_ahead_kb` nilainya dengan menggunakan perintah berikut. Sebelum menggunakan perintah ini, ganti nilai-nilai berikut:

- Ganti *read-ahead-value-kb* dengan ukuran yang diinginkan dalam kilobyte.
- Ganti *efs-mount-point* dengan titik pasang sistem file.

```
device_number=$(stat -c '%d' efs-mount-point)
((major = ($device_number & 0xFFF00) >> 8))
((minor = ($device_number & 0xFF) | (($device_number >> 12) & 0xFFF00)))
sudo bash -c "echo read-ahead-value-kb > /sys/class/bdi/$major:$minor/read_ahead_kb"
```

Contoh berikut menetapkan `read_ahead_kb` ukuran untuk 15 MB.

```
device_number=$(stat -c '%d' efs)
((major = ($device_number & 0xFFF00) >> 8))
((minor = ($device_number & 0xFF) | (($device_number >> 12) & 0xFFF00)))
sudo bash -c "echo 15000 > /sys/class/bdi/$major:$minor/read_ahead_kb"
```

Memecahkan masalah kinerja Amazon EFS

Secara umum, jika Anda mengalami masalah dengan Amazon EFS yang kesulitan Anda selesaikan, konfirmasi bahwa Anda menggunakan kernel Linux terbaru. Jika Anda menggunakan distribusi Linux perusahaan, kami merekomendasikan hal berikut:

- Amazon Linux 2 dengan kernel 4.3 atau yang lebih baru
- Amazon Linux 2015.09 atau yang lebih baru
- RHEL 7.3 atau yang lebih baru
- Semua versi Ubuntu 16.04
- Ubuntu 14.04 dengan kernel 3.13.0-83 atau yang lebih baru
- SLES 12 Sp2 atau yang lebih baru

Jika Anda menggunakan distribusi lain atau kernel khusus, kami merekomendasikan kernel versi 4.3 atau yang lebih baru.

 Note

RHEL 6.9 mungkin kurang optimal untuk beban kerja tertentu karena. [Kinerja buruk saat membuka banyak file secara paralel](#)

Topik

- [Tidak dapat membuat sistem berkas EFS](#)
- [Akses ditolak ke file yang diizinkan pada sistem file NFS](#)
- [Kesalahan saat mengakses konsol Amazon EFS](#)
- [Instans Amazon EC2 hang](#)
- [Aplikasi menulis sejumlah besar data hang](#)
- [Kinerja buruk saat membuka banyak file secara paralel](#)
- [Pengaturan NFS khusus yang menyebabkan penundaan penulisan](#)
- [Membuat backup dengan Oracle Recovery Manager lambat](#)

Tidak dapat membuat sistem berkas EFS

Permintaan untuk membuat sistem file EFS gagal dengan pesan berikut:

```
User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

Tindakan yang harus diambil

Periksa kebijakan AWS Identity and Access Management (IAM) Anda untuk mengonfirmasi bahwa Anda berwenang membuat sistem file EFS dengan kondisi sumber daya yang ditentukan. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses untuk Amazon EFS](#).

Akses ditolak ke file yang diizinkan pada sistem file NFS

Ketika pengguna yang ditugaskan lebih dari 16 grup akses IDs (GIDs) mencoba untuk melakukan operasi pada sistem file NFS, mereka dapat ditolak akses ke file yang diizinkan pada sistem file. [Masalah ini terjadi karena protokol NFS mendukung maksimum 16 GIDs per pengguna, dan tambahan apa pun GIDs dipotong dari permintaan klien NFS, seperti yang didefinisikan dalam RFC 5531.](#)

Tindakan yang harus diambil

Merestrukturisasi pemetaan pengguna dan grup NFS Anda sehingga setiap pengguna ditetapkan tidak lebih dari 16 grup akses (). GIDs

Kesalahan saat mengakses konsol Amazon EFS

Bagian ini menjelaskan kesalahan yang mungkin dialami pengguna saat mengakses konsol manajemen Amazon EFS.

Kesalahan mengautentikasi kredensial untuk **ec2:DescribeVPCs**

Pesan galat berikut ditampilkan saat mengakses konsol Amazon EFS:

```
AuthFailure: An error occurred authenticating your credentials for ec2:DescribeVPCs.
```

Kesalahan ini menunjukkan bahwa kredensial login Anda tidak berhasil mengautentikasi dengan layanan Amazon EC2. Konsol Amazon EFS memanggil layanan Amazon EC2 atas nama Anda saat membuat sistem file EFS di VPC yang Anda pilih.

Tindakan yang harus diambil

Pastikan waktu klien mengakses konsol Amazon EFS diatur dengan benar.

Instans Amazon EC2 hang

Instans Amazon EC2 dapat hang karena Anda menghapus target pemasangan sistem file tanpa terlebih dahulu melepas sistem file.

Tindakan yang harus diambil

Sebelum Anda menghapus target pemasangan sistem file, lepaskan sistem file. Untuk informasi selengkapnya tentang melepas sistem file Amazon EFS Anda, lihat [Melepaskan sistem file](#).

Aplikasi menulis sejumlah besar data hang

Aplikasi yang menulis sejumlah besar data ke Amazon EFS hang dan menyebabkan instance reboot.

Tindakan yang harus diambil

Jika aplikasi membutuhkan waktu terlalu lama untuk menulis semua datanya ke Amazon EFS, Linux mungkin reboot karena tampaknya prosesnya menjadi tidak responsif. Dua parameter konfigurasi kernel menentukan perilaku ini, `kernel.hung_task_panic` dan `kernel.hung_task_timeout_secs`.

Dalam contoh berikut, keadaan proses hang dilaporkan oleh `ps` perintah dengan `D` sebelum instance reboot, menunjukkan bahwa proses sedang menunggu I/O.

```
$ ps aux | grep large_io.py
root 33253 0.5 0.0 126652 5020 pts/3 D+ 18:22 0:00 python large_io.py
/efs/large_file
```

Untuk mencegah reboot, tingkatkan periode batas waktu atau nonaktifkan kepanikan kernel saat tugas yang macet terdeteksi. Perintah berikut menonaktifkan kepanikan kernel tugas yang digantung di sebagian besar sistem Linux.

```
$ sudo sysctl -w kernel.hung_task_panic=0
```

Kinerja buruk saat membuka banyak file secara paralel

Aplikasi yang membuka banyak file secara paralel tidak mengalami peningkatan kinerja I/O paralelisasi yang diharapkan.

Tindakan yang harus diambil

Masalah ini terjadi pada klien Network File System versi 4 (NFSv4) dan pada klien RHEL 6 yang menggunakan NFSv4 .1 karena klien NFS ini membuat serial operasi NFS OPEN dan CLOSE. Gunakan protokol NFS versi 4.1 dan salah satu [distribusi Linux](#) yang disarankan yang tidak memiliki masalah ini.

Jika Anda tidak dapat menggunakan NFSv4 .1, ketahuilah bahwa klien Linux NFSv4 .0 membuat serial permintaan buka dan tutup berdasarkan ID pengguna dan grup. IDs Serialisasi ini terjadi meskipun beberapa proses atau beberapa utas mengeluarkan permintaan secara bersamaan. Klien hanya mengirim satu operasi buka atau tutup ke server NFS pada satu waktu, ketika semua IDs pertandingan. Untuk mengatasi masalah ini, Anda dapat melakukan salah satu tindakan berikut:

- Anda dapat menjalankan setiap proses dari ID pengguna yang berbeda pada instans Amazon EC2 yang sama.

- Anda dapat membiarkan pengguna sama IDs di semua permintaan terbuka, dan memodifikasi kumpulan grup IDs sebagai gantinya.
- Anda dapat menjalankan setiap proses dari instans Amazon EC2 yang terpisah.

Pengaturan NFS khusus yang menyebabkan penundaan penulisan

Anda memiliki pengaturan klien NFS khusus, dan dibutuhkan hingga tiga detik untuk instans Amazon EC2 untuk melihat operasi penulisan yang dilakukan pada sistem file dari instans Amazon EC2 lainnya.

Tindakan yang harus diambil

Jika Anda mengalami masalah ini, Anda dapat menyelesaikannya dengan salah satu cara berikut:

- Jika klien NFS pada instans Amazon EC2 yang membaca data telah mengaktifkan caching atribut, lepaskan sistem file Anda. Kemudian pasang kembali dengan noac opsi untuk menonaktifkan caching atribut. Caching atribut di NFSv4 .1 diaktifkan secara default.

Note

Menonaktifkan caching sisi klien berpotensi mengurangi kinerja aplikasi Anda.

- Anda juga dapat menghapus cache atribut sesuai permintaan dengan menggunakan bahasa pemrograman yang kompatibel dengan prosedur NFS. Untuk melakukan ini, Anda dapat mengirim permintaan ACCESS prosedur segera sebelum permintaan baca.

Misalnya, menggunakan bahasa pemrograman Python, Anda dapat membuat panggilan berikut.

```
# Does an NFS ACCESS procedure request to clear the attribute cache, given a path to
the file
import os
os.access(path, os.W_OK)
```

Membuat backup dengan Oracle Recovery Manager lambat

Membuat backup dengan Oracle Recovery Manager bisa lambat jika Oracle Recovery Manager berhenti selama 120 detik sebelum memulai pekerjaan backup.

Tindakan yang harus diambil

Jika Anda mengalami masalah ini, nonaktifkan Oracle Direct NFS, seperti yang dijelaskan dalam [Mengaktifkan dan Menonaktifkan Kontrol Klien NFS Langsung NFS](#) di Pusat Bantuan Oracle.

 Note

Amazon EFS tidak mendukung Oracle Direct NFS.

Memecahkan masalah AMI dan kernel

Setelah itu, Anda dapat menemukan informasi tentang masalah pemecahan masalah yang terkait dengan Amazon Machine Image (AMI) atau versi kernel tertentu saat menggunakan Amazon EFS dari instans Amazon EC2.

Topik

- [Tidak dapat chown](#)
- [Sistem file terus melakukan operasi berulang kali karena bug klien](#)
- [Klien menemui jalan buntu](#)
- [Daftar file dalam direktori besar membutuhkan waktu lama](#)

Tidak dapat chown

Anda tidak dapat mengubah kepemilikan file/directory menggunakan chown perintah Linux.

Versi kernel dengan bug ini

2.6.32

Tindakan yang harus diambil

Anda dapat mengatasi kesalahan ini dengan melakukan hal berikut:

- Jika Anda melakukan chown langkah penyiapan satu kali yang diperlukan untuk mengubah kepemilikan direktori root EFS, Anda dapat menjalankan chown perintah dari instance yang menjalankan kernel yang lebih baru. Misalnya, gunakan versi terbaru Amazon Linux.
- Jika chown merupakan bagian dari alur kerja produksi Anda, Anda harus memperbarui versi kernel yang akan digunakanchown.

Sistem file terus melakukan operasi berulang kali karena bug klien

Sistem file macet melakukan operasi berulang karena bug klien.

Tindakan yang harus diambil

Perbarui perangkat lunak klien ke versi terbaru.

Klien menemui jalan buntu

Seorang klien menjadi menemui jalan buntu.

Versi kernel dengan bug ini

- CentOS-7 dengan kernel Linux 3.10.0-229.20.1.el7.x86_64
- Ubuntu 15.10 dengan kernel Linux 4.2.0-18-generic

Tindakan yang harus diambil

Lakukan salah satu tindakan berikut:

- Upgrade ke versi kernel yang lebih baru. Untuk CentOS-7, versi kernel Linux 3.10.0-327 atau yang lebih baru berisi perbaikan.
- Turunkan versi ke versi kernel yang lebih lama.

Daftar file dalam direktori besar membutuhkan waktu lama

Hal ini dapat terjadi jika direktori berubah sementara klien NFS Anda iterasi melalui direktori untuk menyelesaikan operasi daftar. Setiap kali klien NFS memperhatikan bahwa isi direktori berubah selama iterasi ini, klien NFS memulai ulang iterasi dari awal. Akibatnya, perintah ls bisa memakan waktu lama untuk menyelesaikan direktori besar dengan file yang sering berubah.

Versi kernel dengan bug ini

Versi kernel CentOS dan RHEL lebih rendah dari 2.6.32-696.el6

Tindakan yang harus diambil

Untuk mengatasi masalah ini, tingkatkan ke versi kernel yang lebih baru.

Mencadangkan dan mereplikasi data di Amazon EFS

Untuk melindungi data Anda, Amazon EFS secara otomatis mencadangkan sistem file EFS Anda. Untuk ketahanan dan perlindungan data lebih lanjut, Anda dapat mereplikasi sistem file EFS Anda dalam file. AWS Region Mencadangkan dan mereplikasi sistem file EFS Anda memastikan bahwa Anda dapat menyediakan operasi atau layanan lanjutan jika terjadi sesuatu pada data sistem file EFS. Misalnya, dalam kasus korupsi data atau kehilangan data.

Topik

- [Mencadangkan sistem file EFS](#)
- [Mereplikasi sistem file EFS](#)

Mencadangkan sistem file EFS

Amazon EFS terintegrasi secara native dengan AWS Backup, layanan berbasis kebijakan yang dikelola sepenuhnya yang dapat Anda gunakan untuk membuat dan mengelola kebijakan pencadangan guna melindungi data Anda di Amazon EFS. Sistem file yang Anda buat menggunakan konsol Amazon EFS secara otomatis dicadangkan secara AWS Backup default. Saat Anda menggunakan CLI atau API untuk membuat sistem file, pencadangan otomatis diaktifkan secara default hanya untuk sistem file One Zone.

Menggunakan AWS Backup untuk Amazon EFS, Anda dapat melakukan tindakan berikut:

- Kelola penjadwalan dan retensi pencadangan otomatis dengan mengonfigurasi paket cadangan. Anda menentukan frekuensi pencadangan, kapan harus mencadangkan, berapa lama untuk menyimpan cadangan, dan kebijakan siklus hidup untuk pencadangan.
- Pulihkan cadangan data Amazon EFS. Anda dapat mengembalikan data sistem file ke sistem file baru atau yang sudah ada. Anda juga dapat memilih apakah akan melakukan pemulihan penuh atau pemulihan tingkat item.

Untuk informasi selengkapnya tentang penggunaan AWS Backup, lihat [Memulai AWS Backup](#) di Panduan AWS Backup Pengembang.

Topik

- [Cara AWS Backup bekerja dengan Amazon EFS](#)
- [Izin IAM yang diperlukan](#)

- [Kinerja Backup](#)
- [Mengelola backup otomatis sistem file EFS](#)

Cara AWS Backup bekerja dengan Amazon EFS

Sistem file yang Anda buat menggunakan konsol Amazon EFS secara otomatis dicadangkan secara AWS Backup default. Saat Anda menggunakan CLI atau API untuk membuat sistem file, pencadangan otomatis diaktifkan secara default hanya untuk sistem file One Zone. Anda dapat mengaktifkan backup otomatis setelah membuat sistem file EFS Anda menggunakan API AWS CLI atau . Paket cadangan EFS default menggunakan pengaturan yang AWS Backup disarankan untuk pencadangan otomatis—pencadangan harian dengan periode retensi 35 hari. Pencadangan yang dibuat menggunakan paket cadangan EFS default disimpan dalam brankas cadangan EFS default, yang juga dibuat oleh Amazon EFS atas nama Anda. Paket cadangan default dan brankas cadangan tidak dapat dihapus.

Semua data dalam sistem file EFS dicadangkan, apa pun kelas penyimpanan data tersebut. Anda tidak dikenakan biaya akses data saat mencadangkan sistem file EFS yang mengaktifkan manajemen siklus hidup dan memiliki data di kelas penyimpanan Infrequent Access (IA) atau Archive. Saat Anda memulihkan titik pemulihan, semua file dikembalikan ke kelas penyimpanan Standar.

Cadangan inkremental

AWS Backup melakukan pencadangan tambahan dari sistem file EFS. Selama pencadangan awal, salinan seluruh sistem file dibuat. Selama pencadangan berikutnya dari sistem file itu, hanya file dan direktori yang telah diubah, ditambahkan, atau dihapus yang disalin. Dengan setiap cadangan tambahan, AWS Backup menyimpan data referensi yang diperlukan untuk memungkinkan pemulihan penuh. Pendekatan ini meminimalkan waktu yang diperlukan untuk menyelesaikan pencadangan dan menghemat biaya penyimpanan dengan tidak menduplikasi data.

Konsistensi Backup

Amazon EFS dirancang agar sangat tersedia. Anda dapat mengakses dan memodifikasi sistem file EFS Anda saat pencadangan Anda terjadi AWS Backup. Namun, ketidakkonsistenan, seperti data duplikat, miring, atau dikecualikan, dapat terjadi jika Anda membuat modifikasi pada sistem file Anda saat pencadangan terjadi. Modifikasi ini termasuk menulis, mengganti nama, memindahkan, atau menghapus operasi. Untuk memastikan pencadangan yang konsisten, kami sarankan Anda menjeda aplikasi atau proses yang memodifikasi sistem file selama proses pencadangan. Atau, jadwalkan pencadangan Anda terjadi selama periode ketika sistem file tidak dimodifikasi.

Jendela penyelesaian Backup

Anda dapat secara opsional menentukan jendela penyelesaian untuk cadangan. Jendela ini mendefinisikan periode waktu di mana cadangan perlu diselesaikan. Jika Anda menentukan jendela penyelesaian, pastikan Anda mempertimbangkan kinerja yang diharapkan dan ukuran serta susunan sistem file Anda. Melakukan hal ini membantu memastikan bahwa cadangan Anda dapat diselesaikan selama jendela.

Pencadangan yang tidak selesai selama jendela yang ditentukan ditandai dengan status yang tidak lengkap. Selama pencadangan terjadwal berikutnya, AWS Backup lanjutkan pada titik yang ditinggalkan. Anda dapat melihat status semua cadangan Anda di Konsol AWS Backup Manajemen.

Praktik terbaik untuk jendela penyelesaian cadangan

Saat mengonfigurasi jendela penyelesaian cadangan untuk sistem file EFS Anda, pertimbangkan praktik terbaik berikut untuk memastikan operasi pencadangan yang andal:

- **Pantau durasi pencadangan** - Jika Anda melihat bahwa pencadangan Anda membutuhkan waktu yang lama (misalnya, 20 hari atau lebih), pertimbangkan untuk memperpendek jendela penyelesaian Anda menjadi 3 hari. Pendekatan ini memungkinkan AWS Backup untuk melakukan beberapa pencadangan sebagian yang lebih pendek daripada mencoba menyelesaikan seluruh cadangan dalam satu jendela yang diperluas.
- **Pertimbangkan karakteristik sistem file** — Sistem file besar dengan jutaan file atau volume data yang signifikan dapat mengambil manfaat dari jendela penyelesaian yang lebih pendek. Rekomendasi jendela 3 hari membantu memastikan bahwa operasi pencadangan selesai dengan sukses dan dilanjutkan secara efisien jika diperlukan.
- **Rencanakan kemajuan tambahan** — Saat menggunakan jendela penyelesaian yang lebih pendek, AWS Backup secara otomatis melanjutkan dari tempat yang ditinggalkannya selama pencadangan terjadwal berikutnya. Pendekatan inkremental ini bisa lebih dapat diandalkan untuk sistem file besar daripada mencoba menyelesaikan semuanya dalam satu jendela panjang.
- **Uji strategi pencadangan Anda** — Sebelum mengandalkan konfigurasi cadangan dalam produksi, ujilah dengan ukuran dan karakteristik sistem file spesifik Anda untuk menentukan jendela penyelesaian optimal untuk kasus penggunaan Anda.

Note

Jendela penyelesaian yang lebih pendek tidak menunjukkan batasan dalam AWS Backup kemampuan untuk menangani sistem file besar. Sebaliknya, mereka menyediakan proses

pencadangan yang lebih dapat diprediksi dan dikelola yang dapat beradaptasi dengan berbagai ukuran sistem file dan kondisi jaringan.

Pencadangan sesuai permintaan

Dengan AWS Backup, Anda dapat menyimpan satu sumber daya ke brankas cadangan sesuai permintaan. Tidak seperti pencadangan terjadwal, Anda tidak perlu membuat rencana cadangan untuk memulai pencadangan sesuai permintaan. Anda masih dapat menetapkan siklus hidup ke cadangan Anda, yang secara otomatis memindahkan titik pemulihan ke tingkat penyimpanan dingin dan mencatat kapan harus menghapusnya.

Selain itu, AWS Backup secara otomatis mentransisikan data ke cold storage hanya untuk data yang tidak lagi ada dalam cadangan hangat terbaru. Misalnya, sistem file Anda memiliki 100 file saat Anda membuat cadangan dan Anda menghapus dua file sehari setelah Anda membuat cadangan ($100 \text{ file} - 2 \text{ file} = 98 \text{ file}$ pada hari kedua). Saat Anda mentransisikan data ke penyimpanan dingin, hanya dua file yang dihapus yang pindah ke penyimpanan dingin dan 98 file sisanya ditagih sebagai penyimpanan hangat.

Pencadangan bersamaan

AWS Backup membatasi pencadangan ke satu cadangan bersamaan per sumber daya. Oleh karena itu, pencadangan terjadwal atau sesuai permintaan mungkin gagal jika pekerjaan cadangan sudah berlangsung. Untuk informasi selengkapnya tentang AWS Backup batasan, lihat [AWS Backup kuota](#) di Panduan AWS Backup Pengembang.

Penghapusan Backup

Kebijakan EFS backup vault Access default disetel untuk menolak penghapusan titik pemulihan. Untuk menghapus cadangan sistem file EFS yang ada, Anda harus mengubah kebijakan akses vault. Jika Anda mencoba menghapus titik pemulihan EFS tanpa mengubah kebijakan akses vault, Anda menerima pesan galat berikut:

```
"Access Denied: Insufficient privileges to perform this action. Please consult with the account administrator for necessary permissions."
```

Untuk mengedit kebijakan akses vault cadangan default, Anda harus memiliki izin untuk mengedit kebijakan. Untuk informasi selengkapnya, lihat [Mengizinkan semua tindakan IAM \(akses admin\)](#) di Panduan Pengguna IAM.

Izin IAM yang diperlukan

AWS Backup membuat peran terkait layanan atas nama Anda di akun Anda. Peran ini memiliki izin yang diperlukan untuk melakukan backup Amazon EFS.

Anda dapat menggunakan `elasticfilesystem:restore` tindakan `elasticfilesystem:backup` dan untuk mengizinkan atau menolak entitas IAM (seperti pengguna, grup, atau peran) kemampuan untuk membuat atau memulihkan cadangan sistem file EFS. Anda dapat menggunakan tindakan ini dalam kebijakan sistem file atau dalam kebijakan IAM berbasis identitas. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses untuk Amazon EFS](#) dan [Menggunakan IAM untuk mengontrol akses ke sistem file](#).

Kinerja Backup

Secara umum, Anda dapat mengharapkan tingkat pencadangan dan pemulihan berikut dengan AWS Backup. Tarifnya mungkin lebih rendah untuk beberapa beban kerja, seperti yang berisi file atau direktori besar.

- Backup rate 2.000 file per detik atau 400 megabyte per detik (MBps), mana yang lebih lambat.
- Tingkat pemulihan 1.500 file per detik atau 200 MBps, mana yang lebih lambat.

Durasi maksimum untuk operasi cadangan AWS Backup adalah 30 hari.

Menggunakan AWS Backup tidak mengkonsumsi akumulasi kredit burst, dan itu tidak dihitung terhadap batas operasi file mode kinerja Tujuan Umum. Untuk informasi selengkapnya, lihat [Kuota untuk sistem file Amazon EFS](#).

Mengelola backup otomatis sistem file EFS

Saat Anda membuat sistem file menggunakan konsol Amazon EFS, pencadangan otomatis diaktifkan secara default. Anda dapat mengaktifkan backup otomatis setelah membuat sistem file Anda menggunakan API AWS CLI atau.

Anda dapat mengedit pengaturan paket cadangan default menggunakan AWS Backup konsol. Untuk informasi selengkapnya, lihat [Mengelola paket cadangan](#) di Panduan AWS Backup Pengembang. [Anda dapat melihat semua pencadangan otomatis Anda, dan mengedit pengaturan paket cadangan EFS default menggunakan konsol.AWS Backup](#)

Amazon EFS menerapkan kunci tag `aws:elasticfilesystem:default-backup` sistem dengan nilai `enabled` ke sistem file EFS saat pencadangan otomatis diaktifkan.

Setelah membuat sistem file, Anda dapat mengaktifkan atau menonaktifkan pencadangan otomatis menggunakan konsol, EFS API AWS CLI, atau EFS.

Menggunakan konsol

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di halaman Sistem file, pilih sistem file yang ingin Anda aktifkan atau nonaktifkan pencadangan otomatis dan tampilkan halaman detail sistem File.
3. Pilih Edit di panel Pengaturan umum.
4.
 - Untuk mengaktifkan pencadangan otomatis, pilih Aktifkan pencadangan otomatis.
 - Untuk mematikan pencadangan otomatis, hapus Aktifkan pencadangan otomatis.
5. Pilih Simpan perubahan.

Menggunakan AWS CLI

- Gunakan perintah `put-backup-policy` CLI (operasi API yang sesuai adalah [PutBackupPolicy](#)) aktifkan atau nonaktifkan pencadangan otomatis untuk sistem file yang ada.
- Gunakan perintah berikut untuk mengaktifkan backup otomatis.

```
$ aws efs put-backup-policy --file-system-id fs-01234567 \
--backup-policy Status="ENABLED"
```

Amazon EFS merespons dengan kebijakan cadangan baru.

```
{
  "BackupPolicy": {
    "Status": "ENABLING"
  }
}
```

- Gunakan perintah berikut untuk mematikan backup otomatis.

```
$ aws efs put-backup-policy --file-system-id fs-01234567 \
--backup-policy Status="DISABLED"
```

Amazon EFS merespons dengan kebijakan cadangan baru.

```
{
  "BackupPolicy": {
    "Status": "DISABLING"
  }
}
```

Mereplikasi sistem file EFS

Untuk ketahanan dan perlindungan data yang diperluas, Anda dapat mereplikasi sistem file EFS Anda dalam file. AWS Region Saat Anda mengaktifkan replikasi pada sistem file EFS, Amazon EFS secara otomatis dan transparan mereplikasi data dan metadata pada sistem file sumber ke sistem file tujuan. Jika terjadi bencana atau saat melakukan latihan hari permainan, Anda bisa gagal ke sistem file replika Anda. Untuk melanjutkan operasi, Anda kemudian dapat gagal kembali ke sistem file utama.

Untuk mengelola proses pembuatan sistem file tujuan dan menjaganya agar tetap disinkronkan dengan sistem file sumber, Amazon EFS menggunakan konfigurasi replikasi.

Setelah Anda membuat konfigurasi replikasi, Amazon EFS secara otomatis membuat sistem file sumber dan tujuan tetap disinkronkan. Perubahan yang dilakukan pada sistem file sumber tidak ditransfer ke sistem file tujuan secara point-in-time konsisten. Sebaliknya mereka ditransfer berdasarkan waktu terakhir yang disinkronkan untuk replikasi. Waktu sinkronisasi terakhir menunjukkan kapan sinkronisasi terakhir yang berhasil antara sumber dan tujuan selesai. Perubahan yang dilakukan pada sistem file sumber Anda pada waktu terakhir yang disinkronkan direplikasi ke sistem file tujuan, sementara perubahan yang dilakukan pada sistem file sumber setelah waktu terakhir yang disinkronkan mungkin tidak direplikasi. Untuk informasi selengkapnya, lihat [Melihat detail replikasi](#).

Replikasi tersedia Wilayah AWS di semua tempat Amazon EFS tersedia. Untuk mereplikasi sistem file EFS di Wilayah yang dinonaktifkan secara default, Anda harus memilih Region terlebih dahulu. Untuk informasi selengkapnya, lihat [Menentukan Wilayah AWS akun mana yang dapat digunakan](#) dalam Panduan Referensi AWS Umum. Jika Anda memilih keluar dari Wilayah nanti, Amazon EFS menghentikan sementara semua aktivitas replikasi untuk Wilayah tersebut. Untuk melanjutkan aktivitas replikasi untuk Wilayah, pilih AWS Region lagi.

 Note

Replikasi tidak mendukung penggunaan tag untuk kontrol akses berbasis atribut (ABAC).

Topik

- [Biaya](#)
- [Performa replikasi](#)
- [Izin IAM yang diperlukan](#)
- [Mengkonfigurasi replikasi ke sistem file EFS baru](#)
- [Mengkonfigurasi replikasi ke sistem file EFS yang ada](#)
- [Mereplikasi sistem file EFS di seluruh akun AWS](#)
- [Melihat detail replikasi](#)
- [Menghapus konfigurasi replikasi](#)
- [Menggunakan replika](#)

Biaya

Untuk memfasilitasi replikasi, Amazon EFS membuat direktori dan metadata tersembunyi pada sistem file tujuan. Ini setara dengan sekitar 12 mebibytes (MiB) data terukur yang Anda tagihan. Untuk informasi selengkapnya tentang pengukuran penyimpanan sistem file, lihat [Bagaimana Amazon EFS melaporkan sistem file dan ukuran objek](#).

Performa replikasi

Saat Anda membuat replikasi baru atau membalikkan arah replikasi yang ada selama proses failback, Amazon EFS melakukan sinkronisasi awal, yang mencakup serangkaian tindakan persiapan satu kali untuk mendukung replikasi. Data yang direplikasi dapat diakses di sistem file tujuan hanya setelah sinkronisasi awal selesai. Jumlah waktu yang diperlukan sinkronisasi awal untuk menyelesaikan tergantung pada faktor-faktor seperti ukuran sistem file sumber dan jumlah file di dalamnya.

Setelah replikasi awal selesai, Amazon EFS mempertahankan Recovery Point Objective (RPO) selama 15 menit untuk sebagian besar sistem file. Namun, jika sistem file sumber memiliki file yang sangat sering berubah dan memiliki lebih dari 100 juta file atau file yang lebih besar dari 100 GB,

replikasi mungkin memakan waktu lebih dari 15 menit. Untuk informasi tentang pemantauan kapan replikasi terakhir berhasil diselesaikan, lihat [Melihat detail replikasi](#).

Anda dapat memantau kapan sinkronisasi berhasil terakhir terjadi menggunakan konsol, AWS Command Line Interface (AWS CLI), API, dan Amazon CloudWatch. Di CloudWatch, gunakan metrik [TimeSinceLastSyncEFS](#). Untuk informasi selengkapnya, lihat [Melihat detail replikasi](#).

Izin IAM yang diperlukan

Amazon EFS menggunakan nama peran terkait layanan EFS `AWSServiceRoleForAmazonElasticFileSystem` atau peran IAM yang Anda tentukan untuk menyinkronkan replikasi antara sistem file sumber dan tujuan. Untuk memberikan peran IAM, pengguna IAM atau peran yang membuat konfigurasi replikasi harus memiliki izin `iam:PassRole`. Untuk informasi selengkapnya, lihat [Memberikan izin pengguna untuk meneruskan peran ke AWS layanan](#) di Panduan AWS Identity and Access Management Pengguna.

- Untuk informasi lebih lanjut tentang `iam:CreateServiceLinkedRole`, lihat contoh di [Menggunakan peran terkait layanan untuk Amazon EFS](#).
- Untuk informasi selengkapnya tentang peran IAM kustom — lihat [Buat peran IAM dengan kebijakan kepercayaan khusus](#).

Note

Jika Anda melakukan replikasi lintas akun, maka Anda harus memberikan peran IAM saat Anda membuat konfigurasi replikasi. Menggunakan peran terkait layanan tidak diizinkan. Untuk informasi selengkapnya, lihat [Mereplikasi sistem file EFS di seluruh akun AWS](#).

Peran terkait layanan atau peran IAM yang Anda berikan saat membuat konfigurasi replikasi harus memiliki izin replikasi berikut.

- `elasticfilesystem:DescribeFileSystems`
- `elasticfilesystem:CreateFileSystem`
- `elasticfilesystem:CreateReplicationConfiguration`
- `elasticfilesystem>DeleteReplicationConfiguration`
- `elasticfilesystem:DescribeReplicationConfigurations`

Anda dapat menggunakan kebijakan `AmazonElasticFileSystemFullAccess` terkelola untuk mendapatkan semua izin EFS yang diperlukan secara otomatis. Untuk informasi selengkapnya, lihat [AWS kebijakan terkelola: AmazonElasticFileSystemFullAccess](#).

Mengkonfigurasi replikasi ke sistem file EFS baru

Amazon EFS secara otomatis membuat sistem file baru dan menyalin data dan metadata pada sistem file sumber ke sistem file tujuan hanya-baca baru yang Anda pilih. AWS Region Saat Anda mereplikasi ke sistem file baru, Anda memilih jenis sistem file dan kunci AWS Key Management Service (AWS KMS) yang akan digunakan untuk enkripsi. Selain itu, Amazon EFS tidak membuat target pemasangan apa pun saat membuat sistem file tujuan. Setelah Anda membuat konfigurasi replikasi, Anda harus [membuat satu atau beberapa target mount](#) untuk [me-mount sistem file tujuan](#).

Note

Sistem file dapat menjadi bagian dari hanya satu konfigurasi replikasi. Anda tidak dapat menggunakan sistem file tujuan sebagai sistem file sumber dalam konfigurasi replikasi lain.

- Jenis sistem file - Jenis sistem file menentukan ketersediaan dan daya tahan sistem file Amazon EFS menyimpan data dalam file file AWS Region.
- Pilih Regional untuk membuat sistem file yang menyimpan data dan metadata secara berlebihan di semua Availability Zone di dalam. AWS Region
- Pilih One Zone untuk membuat sistem file yang menyimpan data dan metadata secara berlebihan dalam satu Availability Zone.

Untuk informasi selengkapnya tentang jenis sistem file, lihat [Jenis sistem file EFS](#).

- Enkripsi — Semua sistem file tujuan dibuat dengan enkripsi saat istirahat diaktifkan. Anda dapat menentukan AWS KMS kunci yang digunakan untuk mengenkripsi sistem file tujuan. Jika Anda tidak menentukan kunci KMS, kunci KMS yang dikelola layanan untuk Amazon EFS akan digunakan.

Important

Setelah sistem file tujuan dibuat, Anda tidak dapat mengubah kunci KMS.

Sistem file tujuan dibuat dengan pengaturan default berdasarkan sistem file sumber Anda. Pengaturan tambahan dapat diubah setelah pembuatan.

- Pencadangan otomatis - Untuk sistem file tujuan yang menggunakan penyimpanan One Zone, pencadangan otomatis diaktifkan secara default. Setelah sistem file dibuat, Anda dapat mengubah pengaturan cadangan otomatis. Untuk informasi selengkapnya, lihat [Mengelola backup otomatis sistem file EFS](#).
- Mode kinerja — Mode Kinerja sistem file tujuan cocok dengan sistem file sumber, kecuali sistem file tujuan menggunakan penyimpanan One Zone. Dalam hal ini, mode Tujuan Umum digunakan. Mode kinerja tidak dapat diubah.
- Mode Throughput — Mode Throughput sistem file tujuan cocok dengan sistem file sumber. Setelah sistem file dibuat, Anda dapat mengubah mode.

Jika mode throughput sistem file sumber Disediakan, maka jumlah throughput yang disediakan sistem file tujuan cocok dengan sistem file sumber, kecuali jumlah yang disediakan file sumber melebihi batas Wilayah sistem file tujuan. Jika jumlah yang disediakan sistem file sumber melebihi batas Wilayah untuk sistem file tujuan, maka jumlah throughput yang disediakan sistem file tujuan adalah batas Wilayah. Untuk informasi selengkapnya, lihat [Kuota Amazon EFS yang dapat Anda tingkatkan](#).

- Manajemen siklus hidup - Manajemen siklus hidup tidak diaktifkan pada sistem file tujuan. Setelah sistem file tujuan dibuat, Anda dapat mengaktifkannya. Untuk informasi selengkapnya, lihat [Mengelola siklus hidup penyimpanan](#).

Langkah 1: Buat konfigurasi replikasi

Langkah pertama dalam mereplikasi ke sistem file baru adalah membuat konfigurasi replikasi. Data direplikasi ke sistem file tujuan hanya dapat diakses setelah sinkronisasi awal selesai. Durasi sinkronisasi awal tergantung pada faktor-faktor seperti ukuran sistem file sumber dan jumlah file di dalamnya. Untuk informasi selengkapnya tentang kinerja replikasi, lihat [Performa replikasi](#).

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.
2. Buka sistem file yang ingin Anda tiru:

- a. Di panel navigasi kiri, pilih Sistem file.
- b. Dalam daftar Sistem file, pilih sistem file yang ingin Anda replikasi. Sistem file yang Anda pilih tidak dapat menjadi sumber atau sistem file tujuan dalam konfigurasi replikasi yang ada.
3. Pilih tab Replikasi.
4. Di bagian Replikasi, pilih Buat replikasi.
5. Di bagian Pengaturan replikasi, tentukan pengaturan replikasi:
 - a. Untuk konfigurasi Replikasi, pilih apakah akan mereplikasi ke sistem file baru atau yang sudah ada.
 - b. Untuk Tujuan AWS Region, pilih AWS Region di mana untuk mereplikasi sistem file.
6. Di bagian Pengaturan sistem file tujuan, tentukan pengaturan sistem file tujuan.
 - a. Untuk jenis sistem File, pilih opsi penyimpanan untuk sistem file:
 - Untuk membuat sistem file yang menyimpan data secara berlebihan di beberapa Availability Zone yang terpisah secara geografis di dalam sebuah AWS Region, pilih Regional.
 - Untuk membuat sistem file yang menyimpan data secara berlebihan dalam satu Availability Zone dalam satu AWS Region, pilih One Zone, lalu pilih Availability Zone.

Untuk informasi selengkapnya, lihat [Jenis sistem file EFS](#).

 Note

Sistem file One Zone tidak tersedia di semua Availability Zone di Wilayah AWS mana Amazon EFS tersedia.

- b. Untuk Enkripsi, enkripsi data saat istirahat diaktifkan secara otomatis pada sistem file tujuan. Secara default, Amazon EFS menggunakan AWS Key Management Service (AWS KMS) kunci layanan (aws/elasticfilesystem) Anda. Untuk menggunakan tombol KMS yang berbeda, pilih tombol KMS atau masukkan nama sumber daya Amazon (ARN) kunci.

 Important

Setelah sistem file dibuat, Anda tidak dapat mengubah kunci KMS.

Buat konfigurasi replikasi ()AWS CLI

Bagian ini memberikan contoh untuk membuat konfigurasi replikasi dalam AWS CLI menggunakan `create-replication-configuration` perintah. Perintah API yang setara adalah [CreateReplicationConfiguration](#).

Example: Buat konfigurasi replikasi untuk sistem file tujuan Regional

Contoh berikut membuat konfigurasi replikasi untuk sistem *fs-0123456789abcdef1* file. Contoh menggunakan Region parameter untuk membuat sistem file tujuan di file *eu-west-2* AWS Region. KmsKeyIdParameter menentukan ID kunci KMS untuk digunakan saat mengenkripsi sistem file tujuan:

```
aws efs create-replication-configuration \
--source-file-system-id fs-0123456789abcdef1 \
--destinations "[{\"Region\":\"eu-west-2\", \"KmsKeyId\":\"arn:aws:kms:us-east-2:111122223333:key/abcd1234-ef56-ab78-cd90-1111abcd2222\"}]"
```

AWS CLI Tanggapan sebagai berikut:

```
{
  "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1",
  "SourceFileSystemRegion": "us-east-1",
  "Destinations": [
    {
      "Status": "ENABLING",
      "FileSystemId": "fs-0123456789abcde22",
      "Region": "eu-west-2"
    }
  ],
  "SourceFileSystemId": "fs-0123456789abcdef1",
  "CreationTime": 1641491892.0,
  "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1"
}
```


Example: Buat konfigurasi replikasi untuk sistem file tujuan One Zone

Contoh berikut membuat konfigurasi replikasi untuk sistem `fs-0123456789abcdef1` file. Contoh menggunakan `AvailabilityZoneName` parameter untuk membuat sistem file tujuan One Zone di `us-west-2a` Availability Zone. Karena tidak ada kunci KMS yang ditentukan, sistem file tujuan dienkripsi menggunakan kunci AWS KMS layanan default akun (`aws/elasticfilesystem`

```
aws efs create-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1 \  
--destinations AvailabilityZoneName=us-west-2a
```

Langkah 2: Pasang sistem file tujuan

Amazon EFS tidak membuat target pemasangan apa pun saat membuat sistem file tujuan. Untuk me-mount sistem file tujuan, Anda harus membuat satu atau beberapa target mount. Untuk informasi selengkapnya, lihat [Memasang sistem file EFS](#).

Mengkonfigurasi replikasi ke sistem file EFS yang ada

Amazon EFS mereplikasi data dan metadata pada sistem file sumber ke sistem file tujuan dan AWS Region yang Anda pilih. Selama replikasi, Amazon EFS mengidentifikasi perbedaan data antara sistem file dan menerapkan perbedaan ke sistem file tujuan.

Untuk mereplikasi ke sistem file yang ada, lakukan langkah-langkah berikut.

Topik

- [Langkah 1: Nonaktifkan perlindungan penyimpanan replikasi sistem file](#)
- [Langkah 2: Buat konfigurasi replikasi](#)

Note

Sistem file dapat menjadi bagian dari hanya satu konfigurasi replikasi. Anda tidak dapat menggunakan sistem file tujuan sebagai sistem file sumber dalam konfigurasi replikasi lain.

Langkah 1: Nonaktifkan perlindungan penimpanan replikasi sistem file

Saat Anda membuat sistem file Amazon EFS, perlindungan penimpanan replikasi diaktifkan secara default. Perlindungan penimpanan replikasi mencegah sistem file digunakan sebagai tujuan dalam konfigurasi replikasi. Sebelum Anda dapat menggunakan sistem file sebagai tujuan dalam konfigurasi replikasi, Anda harus menonaktifkan perlindungan. Jika Anda menghapus konfigurasi replikasi, perlindungan penimpanan replikasi sistem file diaktifkan kembali dan sistem file menjadi dapat ditulis.

Status perlindungan penimpanan replikasi untuk sistem file Amazon EFS dapat memiliki salah satu nilai yang dijelaskan dalam tabel berikut.

Status sistem berkas	Deskripsi
ENABLED	Sistem file tidak dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file dapat ditulis. Perlindungan penimpanan replikasi secara default ENABLED.
DISABLED	Sistem file dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi.
MEREPLIKASI	Sistem file digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dimodifikasi oleh Amazon EFS selama replikasi.

Izin yang diperlukan

Menonaktifkan perlindungan penimpanan replikasi memerlukan izin untuk tindakan tersebut. `elasticfilesystem:UpdateFileSystemProtection` Untuk informasi selengkapnya, lihat [AWS kebijakan terkelola: AmazonElasticFileSystemFullAccess](#).

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file.
3. Dalam daftar Sistem file, pilih sistem file Amazon EFS yang ingin Anda gunakan sebagai sistem file tujuan dalam konfigurasi replikasi.

4. Di bagian Perlindungan sistem file, matikan Perlindungan Penimpaan Replikasi.

Untuk menonaktifkan perlindungan penimpaan replikasi (AWS CLI)

Dalam contoh berikut, perintah `update-file-system-protection` CLI menonaktifkan perlindungan penimpaan replikasi untuk sistem file yang ditentukan. Perintah API yang setara adalah [UpdateFileSystemProtection](#).

```
aws efs update-file-system-protection
--file-system-id fs-0a8b2be428114d97c
--replication-overwrite-protection DISABLED
```

AWS CLI Respons sebagai berikut.

```
{
  "ReplicationOverwriteProtection": "DISABLED"
}
```

Langkah 2: Buat konfigurasi replikasi

Setelah Anda menonaktifkan perlindungan penimpaan replikasi pada sistem file tujuan, Anda dapat membuat konfigurasi replikasi. Saat mereplikasi ke sistem file yang ada, sistem file tujuan dapat berada di akun yang sama atau di akun yang berbeda dari sistem file sumber.

Sebelum membuat konfigurasi replikasi untuk Amazon EFS, tinjau persyaratan dan pertimbangan penting berikut:

- Jika sistem file sumber dienkripsi, maka sistem file tujuan juga harus dienkripsi. Selain itu, jika file sumber tidak dienkripsi dan sistem file tujuan dienkripsi, maka Anda tidak dapat gagal kembali ke tujuan sumber setelah melakukan failover. Untuk informasi selengkapnya tentang enkripsi, lihat [Enkripsi data di Amazon EFS](#).
- Saat pertama kali mengonfigurasi replikasi ke sistem file yang ada, Amazon EFS menulis data ke atau menghapus data yang ada dari sistem file tujuan agar sesuai dengan data dalam sistem file sumber. Jika Anda tidak ingin mengubah data dalam sistem file tujuan, maka Anda harus mereplikasi ke sistem file baru sebagai gantinya. Untuk informasi selengkapnya, lihat [Mengkonfigurasi replikasi ke sistem file EFS baru](#).
- Data yang direplikasi ke sistem file tujuan hanya dapat diakses setelah sinkronisasi awal selesai. Durasi sinkronisasi tergantung pada faktor-faktor seperti ukuran sistem file sumber dan jumlah file di dalamnya. Untuk informasi selengkapnya tentang kinerja replikasi, lihat [Performa replikasi](#).

Prasyarat

Memiliki salinan ID sistem file tujuan (untuk replikasi akun yang sama) atau ARN sistem file tujuan (untuk replikasi lintas akun) yang ingin Anda gunakan.

Jika sistem file tujuan berbeda Akun AWS dari sistem file sumber, buat peran IAM yang memungkinkan Amazon EFS melakukan replikasi dan menetapkan kebijakan sumber daya ke sistem file. Untuk informasi selengkapnya, lihat [Mereplikasi sistem file EFS di seluruh akun AWS](#).

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.
2. Buka sistem file yang ingin Anda tiru:
 - a. Di panel navigasi kiri, pilih Sistem file.
 - b. Dalam daftar Sistem file, pilih sistem file Amazon EFS yang ingin Anda tiru. Sistem file yang Anda pilih tidak dapat menjadi sumber atau sistem file tujuan dalam konfigurasi replikasi yang ada.
3. Pilih tab Replikasi.
4. Di bagian Replikasi, pilih Buat replikasi.
5. Untuk konfigurasi Replikasi, pilih sistem file yang ada.
6. Pilih sistem file tujuan.
 - Untuk mereplikasi ke sistem file yang Akun AWS sama dengan sistem file sumber:
 1. Pilih Pilih sistem file di akun ini dan, untuk Tujuan AWS Region, pilih yang AWS Region akan mereplikasi sistem file.
 2. Pilih Browse EFS, lalu pilih sistem file. Jalur ke sistem file tujuan Anda muncul di kotak Tujuan.
 - Untuk mereplikasi ke sistem file yang berbeda Akun AWS dari sistem file sumber:
 1. Pilih Tentukan sistem file di akun lain.
 2. Untuk ARN sistem file Tujuan, masukkan Nama Sumber Daya Amazon (ARN) dari sistem file tujuan.

Note

Jika perlindungan penyimpanan replikasi diaktifkan pada sistem file, maka peringatan akan ditampilkan. Pilih Nonaktifkan perlindungan untuk membuka sistem file di tab baru dan matikan perlindungan tanpa replikasi. Setelah menonaktifkan perlindungan, kembali ke tab Buat replikasi dan klik tombol Refresh untuk menghapus pesan.

7. Untuk peran IAM, masukkan ARN dari peran IAM yang memungkinkan Amazon EFS mereplikasi ke sistem file tujuan. Ini opsional untuk replikasi akun yang sama, tetapi diperlukan untuk replikasi lintas akun. Untuk informasi selengkapnya, lihat [Mereplikasi sistem file EFS di seluruh akun AWS](#).
8. Pilih Buat replikasi, ketik konfirmasi di kotak masukan pesan konfirmasi, lalu pilih Buat replikasi. Bagian Replikasi menunjukkan detail replikasi.

Untuk membuat konfigurasi replikasi (AWS CLI)

Bagian ini memberikan contoh untuk membuat konfigurasi replikasi dalam AWS CLI menggunakan `create-replication-configuration` perintah. Perintah API yang setara adalah [CreateReplicationConfiguration](#).

Example: Buat konfigurasi replikasi ke sistem file tujuan yang ada di Wilayah lain

Contoh berikut membuat konfigurasi replikasi di mana ID sistem file *fs-0123456789abcdef1* direplikasi ke ID sistem file *fs-0a8b2be428114d97c* di *eu-west-2* AWS Region

```
aws efs create-replication-configuration \
--source-file-system-id fs-0123456789abcdef1 \
--destinations "[{"Region":"eu-west-2","FileSystemId":"fs-0a8b2be428114d97c"}]"
```

AWS CLI Tanggapan sebagai berikut:

```
{
  "SourceFileSystemId": "fs-0123456789abcdef1",
  "SourceFileSystemRegion": "us-east-1",
  "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1",
  "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1",
```

```

"CreationTime": "2024-10-20T20:40:13+00:00",
"Destinations": [
  {
    "Status": "ENABLING",
    "FileSystemId": "fs-0a8b2be428114d97c",
    "Region": "eu-west-2",
    "OwnerId": "123456789012",
  }
],
"SourceFileSystemOwnerId": "123456789012"
}

```

Example: Buat konfigurasi replikasi lintas akun

Contoh berikut membuat konfigurasi replikasi di mana sumber dan sistem file tujuan berbeda Akun AWS. ID sistem file sumber *fs-0123456789abcdef1* di akun *555666777888* direplikasi ke ID sistem file *fs-0a8b2be428114d97c* di akun *123456789012*. Contoh ini menentukan Nama Sumber Daya Amazon (ARN) dari sistem file tujuan dan ARN peran IAM di akun sumber yang memungkinkan Amazon EFS melakukan replikasi atas namanya. Karena tidak ada kunci KMS yang ditentukan, sistem file tujuan dienkripsi menggunakan kunci AWS KMS layanan default akun (*aws/elasticfilesystem*

```

aws efs
--region $REGION
--endpoint $ENDPOINT create-replication-configuration
--source-file-system-id fs-0123456789abcdef1
--destinations Region=eu-west-2,FileSystemId=arn:aws:elasticfilesystem:eu-
west-2:123456789012:file-system/
fs-0a8b2be428114d97c,RoleArn=arn:aws:iam::555666777888:role/cross-account-replication

```

AWS CLI Tanggapan sebagai berikut:

```

{
  "SourceFileSystemId": "fs-0123456789abcdef1",
  "SourceFileSystemRegion": "us-east-1",
  "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:555666777888:file-
system/fs-0123456789abcdef1",
  "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-
east-1:555666777888:file-system/fs-0123456789abcdef1",
  "CreationTime": "2024-10-20T20:40:13+00:00",
  "Destinations": [

```

```
{
  "Status": "ENABLING",
  "FileSystemId": "fs-0a8b2be428114d97c",
  "Region": "eu-west-2",
  "OwnerId": "123456789012",
  "RoleArn": "arn:aws:iam::555666777888:role/cross-account-replication"
},
"SourceFileSystemOwnerId": "555666777888"
}
```

Mereplikasi sistem file EFS di seluruh akun AWS

Anda dapat mereplikasi sistem file EFS di seluruh Akun AWS. Mereplikasi di seluruh akun meningkatkan ketahanan dan keandalan keseluruhan strategi pemulihan bencana (DR) Anda dan dapat membantu Anda memenuhi mandat kepatuhan perusahaan.

Misalnya, Anda mungkin diminta oleh kebijakan kepatuhan untuk menggunakan akun yang berbeda untuk lingkungan yang berbeda (seperti produksi, pementasan, dan pemulihan bencana (DR)). Atau Anda mungkin menemukan bahwa replikasi di berbagai tempat Akun AWS memberikan isolasi yang lebih kuat, kontrol yang lebih terperinci atas izin dan kebijakan akses, dan audit sumber daya yang lebih mudah. Jika akun produksi dikompromikan (seperti oleh pelanggaran keamanan, kesalahan konfigurasi, atau ancaman orang dalam), memiliki server DR di akun terpisah dapat mencegah penyerang mengaksesnya, mengurangi radius ledakan insiden keamanan, dan meminimalkan risiko perubahan yang tidak sah.

Mereplikasi di seluruh Akun AWS memerlukan pengaturan keamanan dan kebijakan tambahan. Alih-alih menggunakan peran terkait layanan untuk melakukan replikasi lintas akun, Anda harus membuat peran IAM yang memberi izin kepada Amazon EFS untuk melakukan replikasi di akun tujuan. Anda juga perlu membuat kebijakan pada sistem file yang ingin Anda bagikan di seluruh akun. Setelah peran IAM dan kebijakan sistem file dibuat, Anda membuat konfigurasi replikasi.

Topik

- [Buat peran IAM dengan kebijakan kepercayaan khusus](#)
- [Buat kebijakan pada sistem file sumber dan tujuan](#)
- [Buat konfigurasi replikasi](#)

Buat peran IAM dengan kebijakan kepercayaan khusus

Agar Amazon EFS dapat melakukan replikasi lintas akun atas nama akun sumber, peran IAM harus dibuat di akun sumber. Peran tersebut harus memiliki kebijakan `elasticfilesystem.amazonaws.com` kepercayaan untuk memungkinkan Amazon EFS mengambil peran dan bertindak sebagai kepala layanan. Peran harus berisi semua izin IAM yang diperlukan untuk melakukan replikasi (lihat [Izin IAM yang diperlukan](#)) dan memberikan izin eksplisit untuk mereplikasi ke sistem file di akun tujuan.

Prasyarat

Anda harus membuat sistem file sumber dan sistem file tujuan dalam konfigurasi replikasi sebelum Anda dapat membuat peran IAM untuk akun sumber. Amazon EFS tidak dapat membuat sistem file tujuan untuk Anda selama replikasi. Selain itu, Anda harus mengetahui dan memberikan Nama Sumber Daya Amazon (ARN) untuk setiap sistem file.

Untuk membuat peran IAM untuk replikasi lintas akun

Berikut ini adalah langkah-langkah umum untuk membuat peran IAM dengan kebijakan kepercayaan khusus untuk replikasi lintas akun dengan Amazon EFS. Untuk step-by-step petunjuk cara membuat peran IAM, lihat [Membuat peran menggunakan kebijakan kepercayaan khusus](#) di Panduan AWS Identity and Access Management Pengguna.

1. Di AWS Identity and Access Management konsol untuk akun sumber, buat peran IAM yang menggunakan kebijakan kepercayaan berikut. Untuk petunjuknya, lihat [Membuat peran menggunakan kebijakan kepercayaan khusus](#) di Panduan Pengguna AWS Identity and Access Management.

JSON

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



```
]
}
```

- Setelah Anda membuat peran, tetapkan izin berikut untuk peran tersebut. Ganti `arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1` dengan ARN dari sistem file tujuan dan ganti `arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1` dengan ARN dari sistem file sumber. Untuk petunjuk tentang menetapkan izin ke peran, lihat [Membuat kebijakan menggunakan editor JSON](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:CreateReplicationConfiguration",
        "elasticfilesystem:DescribeReplicationConfigurations",
        "elasticfilesystem>DeleteReplicationConfiguration",
        "elasticfilesystem:ReplicationWrite"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1"
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ReplicationRead",
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1"
    }
  ]
}
```

- Salin atau tulis ARN untuk peran IAM. Anda perlu menyediakan ARN saat Anda membuat konfigurasi replikasi.

Buat kebijakan pada sistem file sumber dan tujuan

Untuk berbagi sistem file lintas akun di Amazon EFS, Anda harus menetapkan kebijakan ke sistem file tujuan dan sumber. Kebijakan memberikan atau membatasi akses di seluruh akun ke sistem file tempat mereka diterapkan. Hanya pemilik akun dengan izin untuk mengedit sistem file yang dapat menetapkan kebijakan ke sistem file di akun mereka.

Selain memberikan atau membatasi akses di seluruh akun, kebijakan perlu memberikan izin lain yang diperlukan bagi klien untuk bekerja dengan sistem file, seperti. `elasticfilesystem:ClientMount` Jika tidak, sistem file mungkin tidak dapat diakses oleh klien.

Important

Anda tidak dapat membatasi akses ke sumber daya melalui koneksi TLS. Jika Anda menyertakan `"aws:SecureTransport": "false"` kondisi dalam pernyataan Anda, koneksi klien NFS akan gagal.

Kebijakan untuk sistem file tujuan

Untuk mengizinkan izin akun sumber mereplikasi ke sistem file tujuan dan menghapus konfigurasi replikasi dari akun tujuan, kebijakan berikut harus dibuat pada sistem file tujuan. Ganti `arn:aws:iam::444455556666:root` dengan ID akun yang memiliki sistem file sumber. Ganti `arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1` dengan ARN dari sistem file tujuan

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSourceAccountReplicationActions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
      },
      "Action": [
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:CreateReplicationConfiguration",
```

```

        "elasticfilesystem:DescribeReplicationConfigurations",
        "elasticfilesystem>DeleteReplicationConfiguration",
        "elasticfilesystem:ReplicationWrite"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/
fs-0123456789abcdef1"
  },
  {
    "Sid": "AllowReadOnlyClientAccess",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
    },
    "Action": [
      "elasticfilesystem:ClientMount"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/fs-0123456789abcdef1"
  }
]
}

```

Kebijakan untuk sistem file sumber

Untuk mengizinkan izin akun tujuan menghapus konfigurasi replikasi dari akun sumber, Anda harus menetapkan kebijakan berikut ke sistem file sumber. Ganti `arn:aws:iam::111122223333:root` dengan ID akun yang memiliki sistem file tujuan. Ganti `arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1` dengan ARN dari sistem file sumber.

JSON

```

{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
  "Statement": [
    {
      "Sid": "AllowDestinationAccountToDeleteReplication",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      }
    }
  ]
}

```

```

    },
    "Action": "elasticfilesystem:DeleteReplicationConfiguration",
    "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-
system/fs-5678910112hijkqr1"
  },
  {
    "Sid": "AllowClientAccess",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
    },
    "Action": [
      "elasticfilesystem:ClientMount",
      "elasticfilesystem:ClientWrite",
      "elasticfilesystem:ClientRootAccess"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-
system/fs-5678910112hijkqr1",
    "Condition": {
      "Bool": {
        "elasticfilesystem:AccessedViaMountTarget": "true"
      }
    }
  }
]
}

```

Untuk membuat kebijakan sistem file

Lakukan langkah-langkah berikut untuk sistem file tujuan dan sumber, menggunakan kebijakan di bagian sebelumnya.

1. Masuk ke Konsol Manajemen AWS dengan akun yang memiliki sistem file, lalu buka konsol Amazon EFS di [Amazon EFS Console](#).
2. Buka sistem file:
 - a. Di panel navigasi kiri, pilih Sistem file.
 - b. Dalam daftar Sistem file, pilih sistem file.
3. Pada tab Kebijakan sistem berkas, pilih Edit.
4. Tempelkan kebijakan di Editor kebijakan {Json} lalu pilih Simpan.

Buat konfigurasi replikasi

Setelah Anda membuat peran IAM dan menambahkan kebijakan sistem file ke sistem file sumber dan tujuan, ikuti petunjuk [Mengkonfigurasi replikasi ke sistem file EFS yang ada](#) untuk membuat konfigurasi replikasi.

Melihat detail replikasi

Anda dapat memantau waktu ketika sinkronisasi berhasil terakhir diselesaikan dalam konfigurasi replikasi. Setiap perubahan data pada sistem file sumber yang terjadi sebelum waktu ini telah berhasil direplikasi ke sistem file tujuan. Setiap perubahan yang terjadi setelah waktu ini mungkin tidak sepenuhnya direplikasi. Untuk memantau kapan replikasi terakhir berhasil diselesaikan, Anda dapat menggunakan konsol Amazon EFS, API AWS CLI, atau Amazon CloudWatch.

- Di konsol EFS — Properti terakhir yang disinkronkan di bagian Detail sistem File > Replikasi menunjukkan waktu ketika sinkronisasi terakhir yang berhasil antara sumber dan tujuan selesai.
- Dalam AWS CLI atau API — `LastReplicatedTimestamp` Properti dalam `Destination` objek menunjukkan waktu bahwa sinkronisasi berhasil terakhir selesai. Untuk mengakses properti ini, gunakan perintah `describe-replication-configurations` CLI. [DescribeReplicationConfigurations](#) adalah operasi API yang setara.
- Di CloudWatch — `TimeSinceLastSync` CloudWatch Metrik untuk Amazon EFS menunjukkan waktu yang telah berlalu sejak sinkronisasi terakhir yang berhasil diselesaikan. Untuk informasi selengkapnya, lihat [CloudWatch metrik untuk Amazon EFS](#).

Konfigurasi replikasi dapat memiliki salah satu nilai status yang dijelaskan dalam tabel berikut.

Status replikasi	Deskripsi
ENABLED	Konfigurasi replikasi dalam keadaan sehat dan tersedia untuk digunakan.
ENABLING	Amazon EFS sedang dalam proses membuat konfigurasi replikasi.
DELETING	Amazon EFS menghapus konfigurasi replikasi sebagai respons terhadap permintaan penghapusan yang dimulai pengguna.
PAUSING	Amazon EFS sedang dalam proses menjeda replikasi.

Status replikasi	Deskripsi
PAUSED	Replikasi dijeda karena masalah dengan konfigurasi. Informasi tambahan tentang masalah ini disediakan. Beberapa masalah yang menyebabkan replikasi dihentikan sementara meliputi: • Kesalahan otorisasi. Masalah izin mencegah Amazon EFS melakukan replikasi. Pastikan bahwa peran IAM yang digunakan untuk membuat konfigurasi replikasi memiliki izin untuk melakukan replikasi. Pastikan juga bahwa kebijakan sistem file sudah benar. • AWS Akun tidak tersedia. Pastikan akun sumber dan tujuan dipilih sesuai dengan yang sesuai Wilayah AWS dan tidak ditanggihkan. Untuk informasi selengkapnya, lihat Menentukan Wilayah AWS akun mana yang dapat digunakan dalam Panduan Referensi AWS Umum. • Kunci KMS untuk sistem file sumber atau tujuan tidak dapat diakses. Pastikan bahwa kunci KMS yang ditetapkan untuk setiap sistem file dapat diakses. Untuk informasi selengkapnya, lihat Menggunakan AWS KMS kunci untuk Amazon EFS.
ERROR	Konfigurasi replikasi dalam keadaan gagal dan tidak dapat dipulihkan. Anda harus menghapus konfigurasi replikasi dan membuat yang baru. Informasi tambahan tentang masalah ini disediakan. Untuk replikasi lintas akun atau lintas wilayah, kesalahan dapat disebabkan karena konfigurasi replikasi dihapus dari yang lain atau. Akun AWS AWS Region

Menggunakan konsol

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file.
3. Pilih sistem file dari daftar.
4. Pilih tab Replikasi untuk menampilkan bagian Replikasi.

Di bagian Replikasi, Anda dapat melihat informasi berikut untuk konfigurasi replikasi:

- Status replikasi mungkin Mengaktifkan, Mengaktifkan, Menghapus, Menjeda, Dijeda, atau Kesalahan. Amazon EFS menampilkan detail tentang penyebab status Dijeda atau Kesalahan.
- Arah replikasi menunjukkan arah di mana data sedang direplikasi. Sistem file pertama yang terdaftar adalah sumbernya, dan datanya direplikasi ke sistem file kedua yang terdaftar, yang merupakan tujuan.
- Terakhir disinkronkan menunjukkan kapan sinkronisasi berhasil terakhir terjadi pada sistem file tujuan. Setiap perubahan data pada sistem file sumber yang terjadi sebelum waktu ini berhasil direplikasi ke sistem file tujuan. Setiap perubahan yang terjadi setelah waktu ini mungkin tidak sepenuhnya direplikasi.
- Sistem file replikasi mencantumkan setiap sistem file dalam konfigurasi replikasi dengan ID sistem file-nya, peran yang dimilikinya dalam konfigurasi replikasi (baik sumber atau tujuan), AWS Region di mana ia berada, dan Izinnya. Sistem file sumber memiliki izin Writable, dan sistem file tujuan memiliki izin Read-only.

Menggunakan AWS CLI

Untuk melihat konfigurasi replikasi, gunakan `describe-replication-configurations` perintah. Anda dapat melihat konfigurasi replikasi baik untuk sistem file tertentu, atau semua konfigurasi replikasi untuk tertentu Akun AWS dalam file. AWS Region Perintah API yang setara adalah [DescribeReplicationConfigurations](#).

Jika status konfigurasi replikasi adalah PAUSED atau ERROR, informasi tentang penyebab masalah dan cara memperbaikinya dikembalikan dalam `StatusMessage` parameter.

Example: Lihat konfigurasi replikasi untuk sistem file tertentu

Contoh berikut menjelaskan konfigurasi replikasi untuk sistem `fs-0123456789abcdef1` file.

```
aws efs describe-replication-configurations --file-system-id fs-0123456789abcdef1
```

AWS CLI Respons sebagai berikut.

```
{
  "Replications": [
    {
      "SourceFileSystemArn": "arn:aws:elasticfilesystem:eu-west-1:111122223333:file-system/fs-abcdef0123456789a",
```

```

    "CreationTime": 1641491892.0,
    "SourceFileSystemRegion": "eu-west-1",
    "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:111122223333:file-system/fs-abcdef0123456789a",
    "SourceFileSystemId": "fs-abcdef0123456789a",
    "Destinations": [
      {
        "Status": "ENABLED",
        "FileSystemId": "fs-0123456789abcdef1",
        "Region": "us-east-1"
      }
    ]
  }
]
}

```

Example: Lihat konfigurasi replikasi untuk semua konfigurasi replikasi di akun

Contoh berikut menjelaskan konfigurasi replikasi untuk semua konfigurasi replikasi untuk akun dalam sistem file AWS Region .

```
aws efs describe-replication-configurations
```

AWS CLI Respons sebagai berikut.

```

{
  "Replications": [
    {
      "SourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-0123456789abcdef1",
      "CreationTime": 1641491892.0,
      "SourceFileSystemRegion": "eu-west-1",
      "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-0123456789abcdef1",
      "SourceFileSystemId": "fs-0123456789abcdef1",
      "Destinations": [
        {
          "Status": "ENABLED",

```



```

        "FileSystemId": "fs-abcdef0123456789a",
        "Region": "us-east-1",
        "LastReplicatedTimestamp": 1641491802.375
    }
]
},
{
    "SourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-021345abcdef6789a",
    "CreationTime": 1641491822.0,
    "SourceFileSystemRegion": "eu-west-1",
    "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-021345abcdef6789a",
    "SourceFileSystemId": "fs-021345abcdef6789a",
    "Destinations": [
        {
            "Status": "ENABLED",
            "FileSystemId": "fs-012abc3456789def1",
            "Region": "us-east-1",
            "LastReplicatedTimestamp": 1641491823.575
        }
    ]
}
]
}
}

```

Menghapus konfigurasi replikasi

Jika Anda perlu gagal ke sistem file tujuan, hapus konfigurasi replikasi yang menjadi anggotanya. Setelah Anda menghapus konfigurasi replikasi, sistem file tujuan menjadi dapat ditulis dan perlindungan penyimpanan replikasi diaktifkan kembali. Untuk informasi selengkapnya, lihat [Menggunakan replika](#).

Menghapus konfigurasi replikasi dan mengubah sistem file tujuan agar dapat ditulis dapat memakan waktu beberapa menit untuk menyelesaikannya. Setelah konfigurasi dihapus, Amazon EFS mungkin menulis beberapa data ke `lost+found` direktori di direktori root sistem file tujuan, menggunakan konvensi penamaan berikut:

```
efs-replication-lost+found-source-file-system-id-TIMESTAMP
```

 Note

Anda tidak dapat menghapus sistem file yang merupakan bagian dari konfigurasi replikasi. Anda harus menghapus konfigurasi replikasi sebelum menghapus sistem file.

Anda dapat menghapus konfigurasi replikasi yang ada baik dari sumber atau sistem file tujuan dengan menggunakan konsol Amazon EFS, the AWS CLI, atau API.

Untuk replikasi lintas akun atau lintas wilayah, Amazon EFS menghapus konfigurasi replikasi dari akun sumber dan tujuan atau Wilayah. Jika ada masalah konfigurasi atau izin yang mencegah Amazon EFS menghapus konfigurasi replikasi dari kedua sisi, Anda dapat menghapus konfigurasi hanya dari sisi lokal (akun atau Wilayah tempat penghapusan dilakukan). Menghapus konfigurasi lokal membuat konfigurasi di akun lain atau Wilayah tidak dapat dipulihkan.

Menggunakan konsol

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file.
3. Pilih sumber atau sistem file tujuan yang ada dalam konfigurasi replikasi yang ingin Anda hapus.
4. Pilih tab Replikasi untuk menampilkan bagian Replikasi.
5. Pilih Hapus replikasi untuk menghapus konfigurasi replikasi. Saat diminta, konfirmasi pilihan Anda.

Jika Anda menghapus konfigurasi replikasi lintas akun, dan ada masalah yang mencegah Anda menghapus konfigurasi dari sisi sumber dan tujuan, maka Anda dapat memilih opsi untuk menghapus konfigurasi sistem file ini saja.

 Note

Hapus konfigurasi sistem file hanya jika Amazon EFS tidak dapat menghapus konfigurasi replikasi di akun sumber dan tujuan atau Wilayah. Menghapus konfigurasi lokal membuat konfigurasi di akun lain atau Wilayah tidak dapat dipulihkan.

Menggunakan AWS CLI

Untuk menghapus konfigurasi replikasi, gunakan `delete-replication-configuration` CLI. Perintah API yang setara adalah [DeleteReplicationConfiguration](#).

Contoh berikut menghapus konfigurasi replikasi untuk sistem file sumber. `fs-0123456789abcdef1`

```
aws efs --region us-west-2 delete-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1
```

Jika masalah konfigurasi atau izin mencegah Amazon EFS menghapus konfigurasi replikasi dari kedua sisi, Anda dapat menghapus konfigurasi hanya dari sisi lokal (akun atau Wilayah tempat penghapusan dilakukan). Menghapus konfigurasi lokal membuat konfigurasi di akun lain atau Wilayah tidak dapat dipulihkan. Parameter API yang setara adalah `DeletionMode` dan nilainya adalah `LOCAL_CONFIGURATION_ONLY`.

Contoh berikut menghapus konfigurasi replikasi untuk sistem file sumber `fs-0123456789abcdef1` dari sisi lokal saja.

```
aws efs --region us-west-2 delete-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1 \  
--deletion-mode LOCAL_CONFIGURATION_ONLY
```

Menggunakan replika

Jika terjadi bencana atau saat melakukan latihan hari permainan, Anda dapat gagal ke sistem file replika Anda dengan menghapus konfigurasi replikasinya. Setelah konfigurasi replikasi dihapus, replika menjadi dapat ditulis dan Anda dapat mulai menggunakannya dalam alur kerja aplikasi Anda. Ketika bencana dikurangi atau latihan hari permainan selesai, Anda dapat terus menggunakan replika sebagai sistem file utama atau Anda dapat melakukan failback untuk melanjutkan operasi pada sistem file utama asli Anda.

Selama proses failback, Anda dapat memilih untuk membuang perubahan yang dibuat pada sistem file replika Anda atau melestarikannya dengan menyalinnya kembali ke primer Anda.

- Untuk membuang perubahan yang dibuat pada replika Anda selama failover, buat ulang konfigurasi replikasi asli pada sistem file utama Anda, di mana sistem file replika adalah tujuan replikasi. Selama replikasi, Amazon EFS menyinkronkan sistem file dengan memperbarui data sistem file replika Anda agar sesuai dengan data utama Anda.

- Untuk mereplikasi perubahan yang dibuat pada replika Anda selama failover, buat konfigurasi replikasi pada sistem file replika, di mana sistem file utama adalah tujuan replikasi. Selama replikasi, Amazon EFS mengidentifikasi dan mentransfer perbedaan dari sistem file replika Anda kembali ke sistem file utama. Setelah replikasi selesai, Anda dapat melanjutkan replikasi sistem file utama dengan membuat ulang konfigurasi replikasi asli atau membuat konfigurasi baru.

Jumlah waktu yang dibutuhkan Amazon EFS untuk menyelesaikan proses replikasi bervariasi dan tergantung pada faktor-faktor seperti ukuran sistem file dan jumlah file di dalamnya. Lihat informasi yang lebih lengkap di [Performa replikasi](#).

Mengamankan data Anda di Amazon EFS

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 dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program AWS Kepatuhan Program AWS Kepatuhan](#). Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Elastic File System, lihat [AWS Layanan dalam Lingkup menurut AWS Layanan Program Kepatuhan](#).
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon EFS. Topik berikut menunjukkan cara mengonfigurasi Amazon EFS untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon EFS Anda.

Topik

- [Perlindungan data di Amazon EFS](#)
- [Manajemen identitas dan akses untuk Amazon EFS](#)
- [Menggunakan IAM untuk mengontrol akses ke sistem file](#)
- [Validasi kepatuhan untuk Amazon EFS](#)
- [Ketahanan di Amazon EFS](#)
- [Mengontrol akses jaringan ke sistem file EFS untuk klien NFS](#)
- [Network File System \(NFS\) tingkat pengguna, grup, dan izin](#)
- [Bekerja dengan titik akses](#)
- [Memblokir akses publik ke sistem file EFS](#)

- [Isolasi jaringan untuk Amazon EFS](#)

Perlindungan data di Amazon EFS

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di Amazon EFS.

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. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti

bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon EFS atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Topik

- [Enkripsi data di Amazon EFS](#)
- [Privasi antar jaringan](#)

Enkripsi data di Amazon EFS

Amazon EFS menyediakan kemampuan enkripsi komprehensif untuk melindungi data Anda baik saat istirahat maupun saat transit.

- Enkripsi saat istirahat — Mengenkripsi data yang disimpan di sistem file Anda.
- Enkripsi dalam perjalanan — Mengenkripsi data saat bergerak antara klien Anda dan sistem file.

Jika organisasi Anda tunduk pada kebijakan perusahaan atau peraturan yang memerlukan enkripsi data dan metadata, sebaiknya buat sistem file yang dienkripsi saat istirahat dan memasang sistem file Anda menggunakan enkripsi data dalam perjalanan.

Topik

- [Mengenkripsi data diam](#)
- [Mengenkripsi data dalam perjalanan](#)
- [Menggunakan AWS KMS kunci untuk Amazon EFS](#)
- [Enkripsi pemecahan masalah](#)

Mengkripsi data diam

Enkripsi saat istirahat mengenkripsi data yang disimpan di sistem file EFS Anda. Ini membantu Anda memenuhi persyaratan kepatuhan dan melindungi data sensitif dari akses yang tidak sah. Organisasi Anda mungkin memerlukan enkripsi semua data yang memenuhi klasifikasi tertentu atau dikaitkan dengan aplikasi, beban kerja, atau lingkungan tertentu.

 Note

Infrastruktur manajemen AWS kunci menggunakan Federal Information Processing Standards (FIPS) 140-3 algoritma kriptografi yang disetujui. Infrastruktur ini konsisten dengan rekomendasi National Institute of Standard and Technology (NIST) 800-57.

Saat Anda membuat sistem file menggunakan konsol Amazon EFS, enkripsi saat istirahat diaktifkan secara default. Saat menggunakan AWS CLI, API, atau SDKs untuk membuat sistem file, Anda harus mengaktifkan enkripsi secara eksplisit.

Setelah Anda membuat sistem file EFS, Anda tidak dapat mengubah pengaturan enkripsi. Ini berarti Anda tidak dapat memodifikasi sistem file yang tidak terenkripsi untuk membuatnya terenkripsi. Sebagai gantinya, [replikasi sistem file](#) untuk menyalin data dari sistem file yang tidak terenkripsi ke sistem file terenkripsi baru. Untuk informasi selengkapnya, lihat [Bagaimana cara mengaktifkan enkripsi saat istirahat untuk sistem file EFS yang ada?](#)

Cara kerja enkripsi saat istirahat

Dalam sistem file terenkripsi, data dan metadata dienkripsi secara default sebelum ditulis ke penyimpanan dan secara otomatis didekripsi saat dibaca. Proses ini ditangani secara transparan oleh Amazon EFS, jadi Anda tidak perlu memodifikasi aplikasi Anda.

Amazon EFS menggunakan AWS KMS manajemen kunci sebagai berikut:

- Enkripsi data file — Isi file Anda dienkripsi menggunakan kunci KMS yang Anda tentukan. Ini bisa berupa:
 - The Kunci milik AWS for Amazon EFS (aws/elasticfilesystem) — Opsi default, tanpa biaya tambahan.
 - Kunci terkelola pelanggan yang Anda buat dan kelola — Memberikan kemampuan kontrol dan audit tambahan.
- Enkripsi metadata - Nama file, nama direktori, dan konten direktori dienkripsi menggunakan kunci yang dikelola Amazon EFS secara internal.

Proses enkripsi

Saat sistem file dibuat atau direplikasi ke sistem file di akun yang sama, Amazon EFS menggunakan [Forward Access Session \(FAS\)](#) untuk melakukan panggilan KMS menggunakan kredensial

pemanggil. Dalam CloudTrail log, `kms:CreateGrant` panggilan tampaknya dibuat oleh identitas pengguna yang sama yang menciptakan sistem file atau replikasi. Anda dapat mengidentifikasi panggilan layanan Amazon EFS CloudTrail dengan mencari `invokedBy` bidang dengan nilainya `elasticfilesystem.amazonaws.com`. Kebijakan sumber daya pada kunci KMS harus mengizinkan `CreateGrant` tindakan agar FAS melakukan panggilan.

 Important

Anda mengelola kendali atas hibah, dan dapat mencabutkannya kapan saja. Mencabut hibah mencegah Amazon EFS mengakses kunci KMS untuk operasi masa depan. Untuk informasi selengkapnya, lihat [Pensiun dan pencabutan hibah di Panduan](#) Pengembang AWS Key Management Service .

Saat menggunakan kunci KMS yang dikelola pelanggan, kebijakan sumber daya juga harus mengizinkan prinsipal layanan Amazon EFS dan menyertakan `kms:ViaService` kondisi untuk membatasi akses ke titik akhir layanan tertentu. Contoh:

```
"kms:ViaService":  
  "elasticfilesystem.us-east-2.amazonaws.com"
```

Amazon EFS menggunakan algoritma enkripsi AES-256 standar industri untuk mengenkripsi data dan metadata saat istirahat.

Untuk informasi selengkapnya tentang kebijakan utama KMS untuk Amazon EFS, lihat [Menggunakan AWS KMS kunci untuk Amazon EFS](#).

Menegakkan enkripsi saat istirahat untuk sistem file baru

Anda dapat menggunakan kunci kondisi `elasticfilesystem:Encrypted IAM` dalam kebijakan berbasis identitas AWS Identity and Access Management (IAM) untuk menerapkan pembuatan saat istirahat saat pengguna membuat sistem file EFS. Untuk informasi selengkapnya tentang menggunakan kunci kondisi, lihat [Contoh: Menegakkan pembuatan sistem file terenkripsi](#).

Anda juga dapat menentukan kebijakan kontrol layanan (SCPs) di dalam AWS Organizations untuk menerapkan enkripsi Amazon EFS untuk semua yang ada Akun AWS di organisasi Anda. Untuk informasi selengkapnya tentang kebijakan kontrol [layanan AWS Organizations](#), lihat [Kebijakan kontrol layanan](#) di Panduan AWS Organizations Pengguna.

Menkripsi data dalam perjalanan

Amazon EFS mendukung enkripsi data dalam perjalanan dengan Transport Layer Security (TLS). Saat enkripsi data dalam perjalanan dinyatakan sebagai opsi pemasangan untuk sistem file EFS Anda, Amazon EFS membuat koneksi TLS yang aman dengan sistem file EFS Anda saat memasang sistem file Anda. Semua lalu lintas NFS dirutekan melalui koneksi terenkripsi ini.

Cara kerja enkripsi dalam perjalanan

Sebaiknya gunakan EFS mount helper untuk memasang sistem file Anda karena menyederhanakan proses pemasangan dibandingkan dengan pemasangan dengan NFS. mount EFS mount helper mengelola proses dengan menggunakan `efs-proxy` (untuk `efs-utils` versi 2.0.0 dan yang lebih baru) atau `stunnel` (untuk `efs-utils` versi sebelumnya) untuk membuat koneksi TLS aman dengan sistem file EFS Anda.

Jika Anda tidak menggunakan mount helper, Anda masih dapat mengaktifkan enkripsi data dalam perjalanan. Berikut ini adalah langkah-langkah untuk melakukannya.

Untuk mengaktifkan enkripsi data dalam perjalanan tanpa menggunakan mount helper

1. Unduh dan instal `stunnel`, dan catat port yang sedang didengarkan aplikasi. Untuk informasi selengkapnya, lihat [Upgrade stunnel](#).
2. Jalankan `stunnel` untuk terhubung ke sistem file EFS Anda di port 2049 menggunakan TLS.
3. Menggunakan klien NFS, `mountlocalhost:port`, di `port` mana port yang Anda catat pada langkah pertama.

Karena enkripsi data dalam transit dikonfigurasi berdasarkan per-koneksi, setiap mount yang dikonfigurasi memiliki `stunnel` proses khusus yang berjalan pada instance. Secara default, proses `stunnel` yang digunakan oleh mount helper mendengarkan pada port lokal antara 20049 dan 20449, dan terhubung ke Amazon EFS pada port 2049.

Note

Secara default, saat menggunakan EFS mount helper dengan TLS, ini memberlakukan penggunaan Online Certificate Status Protocol (OCSP) dan pemeriksaan nama host sertifikat. EFS mount helper menggunakan program `stunnel` untuk fungsionalitas TLS-nya. Beberapa versi Linux tidak menyertakan versi `stunnel` yang mendukung fitur TLS ini secara default. Saat menggunakan salah satu versi Linux tersebut, pemasangan sistem file EFS menggunakan TLS gagal.

Setelah Anda menginstal `amazon-efs-utils` paket, untuk meningkatkan versi stunnel sistem Anda, lihat [Upgrade stunnel](#).

Untuk masalah dengan enkripsi, lihat [Enkripsi pemecahan masalah](#).

Saat menggunakan enkripsi data dalam perjalanan, pengaturan klien NFS Anda diubah. Saat Anda memeriksa sistem file yang dipasang secara aktif, Anda melihat satu dipasang ke `127.0.0.1`, atau `localhost`, seperti pada contoh berikut.

```
$ mount | column -t
127.0.0.1:/ on /home/ec2-user/efs          type nfs4
(rw,relatime,vers=4.1,rsize=1048576,wsiz=1048576,namlen=255,hard,proto=tcp,port=20127,timeo=600,siz=1048576)
```

Saat memasang dengan TLS dan EFS mount helper, Anda mengonfigurasi ulang klien NFS Anda untuk dipasang ke port lokal. EFS mount helper memulai `stunnel` proses klien yang mendengarkan di port lokal ini, dan `stunnel` membuka koneksi terenkripsi ke sistem file EFS menggunakan TLS. EFS mount helper bertanggung jawab untuk menyiapkan dan memelihara koneksi terenkripsi ini dan konfigurasi terkait.

Untuk menentukan ID sistem file Amazon EFS mana yang sesuai dengan titik pemasangan lokal mana, Anda dapat menggunakan perintah berikut. Ingatlah untuk mengganti *efs-mount-point* dengan jalur lokal tempat Anda memasang sistem file Anda.

```
grep -E "Successfully mounted.*efs-mount-point" /var/log/amazon/efs/mount.log | tail -1
```

Saat Anda menggunakan EFS mount helper untuk enkripsi data dalam perjalanan, itu juga membuat proses yang disebut `amazon-efs-mount-watchdog`. Proses ini memastikan bahwa setiap proses `stunnel` mount berjalan, dan menghentikan `stunnel` saat sistem file EFS dilepas. Jika karena alasan tertentu proses `stunnel` dihentikan secara tak terduga, proses pengawas memulai ulang.

Menggunakan AWS KMS kunci untuk Amazon EFS

Amazon EFS terintegrasi dengan AWS Key Management Service (AWS KMS) untuk manajemen kunci. Amazon EFS menggunakan kunci yang dikelola pelanggan untuk mengenkripsi sistem file Anda dengan cara berikut:

- Mengenkripsi metadata saat istirahat — Amazon EFS menggunakan kunci yang dikelola AWS Key Management Service untuk mengenkripsi dan mendekripsi metadata sistem file (yaitu, nama file, nama direktori, dan konten direktori).

- Mengenkripsi data file saat istirahat — Anda memilih kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi dan mendekripsi data file (yaitu, isi file Anda). Anda dapat mengaktifkan, menonaktifkan, atau mencabut izin pada kunci yang dikelola pelanggan ini. Kunci yang dikelola pelanggan ini dapat menjadi salah satu dari dua jenis berikut:
 - Kunci yang dikelola AWS untuk Amazon EFS — Ini adalah kunci terkelola pelanggan default, `aws/elasticfilesystem`. Anda tidak dikenakan biaya untuk membuat dan menyimpan kunci yang dikelola pelanggan, tetapi ada biaya penggunaan. Untuk mempelajari lebih lanjut, lihat [AWS Key Management Service harga](#).
 - Kunci yang dikelola pelanggan — Ini adalah kunci KMS yang paling fleksibel untuk digunakan, karena Anda dapat mengonfigurasi kebijakan dan izin kuncinya untuk beberapa pengguna atau layanan. Untuk informasi selengkapnya tentang membuat kunci terkelola pelanggan, lihat [Membuat kunci](#) di Panduan AWS Key Management Service Pengembang.

Jika Anda menggunakan kunci yang dikelola pelanggan untuk enkripsi dan dekripsi data file, Anda dapat mengaktifkan rotasi kunci. Ketika Anda mengaktifkan rotasi kunci, AWS KMS secara otomatis memutar kunci Anda sekali per tahun. Selain itu, dengan kunci yang dikelola pelanggan, Anda dapat memilih kapan harus menonaktifkan, mengaktifkan kembali, menghapus, atau mencabut akses ke kunci yang dikelola pelanggan kapan saja. Untuk informasi selengkapnya, lihat [Menggunakan AWS KMS kunci untuk Amazon EFS](#).

Important

Amazon EFS hanya menerima kunci yang dikelola pelanggan simetris. Anda tidak dapat menggunakan kunci terkelola pelanggan asimetris dengan Amazon EFS.

Enkripsi data dan dekripsi saat istirahat ditangani secara transparan. Namun, AWS akun IDs khusus untuk Amazon EFS muncul di AWS CloudTrail log Anda yang terkait dengan AWS KMS tindakan. Untuk informasi selengkapnya, lihat [Entri file log Amazon EFS untuk sistem encrypted-at-rest file](#).

Kebijakan utama Amazon EFS untuk AWS KMS

Kebijakan utama adalah cara utama untuk mengontrol akses ke kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang kebijakan [utama](#), lihat [Kebijakan utama AWS KMS di Panduan AWS Key Management Service Pengembang](#). Daftar berikut menjelaskan semua izin AWS KMS terkait —yang diperlukan atau didukung oleh Amazon EFS untuk sistem file terenkripsi saat istirahat:

- kms:Encrypt – (Opsional) Mengenkripsi plaintext ke ciphertext. Izin ini termasuk dalam kebijakan kunci default.
- kms:Decrypt – (Wajib) Mendekripsi ciphertext. Ciphertext adalah plaintext yang telah dienkripsi sebelumnya. Izin ini termasuk dalam kebijakan kunci default.
- kms: ReEncrypt — (Opsional) Mengenkripsi data di sisi server dengan kunci yang dikelola pelanggan baru, tanpa mengekspos plaintext data di sisi klien. Data pertama kali didekripsi dan kemudian dienkripsi ulang. Izin ini termasuk dalam kebijakan kunci default.
- kms: GenerateDataKeyWithoutPlaintext — (Diperlukan) Mengembalikan kunci enkripsi data yang dienkripsi di bawah kunci yang dikelola pelanggan. Izin ini disertakan dalam kebijakan kunci default di bawah kms: GenerateDataKey *.
- kms: CreateGrant — (Diperlukan) Menambahkan hibah ke kunci untuk menentukan siapa yang dapat menggunakan kunci dan dalam kondisi apa. Hibah adalah mekanisme izin lainnya untuk kebijakan kunci. Untuk informasi selengkapnya tentang hibah, lihat [Hibah AWS KMS di Panduan AWS Key Management Service Pengembang](#). Izin ini termasuk dalam kebijakan kunci default.
- kms: DescribeKey — (Diperlukan) Memberikan informasi rinci tentang kunci terkelola pelanggan yang ditentukan. Izin ini termasuk dalam kebijakan kunci default.
- kms: ListAliases — (Opsional) Daftar semua alias kunci di akun. Saat Anda menggunakan konsol untuk membuat sistem file terenkripsi, izin ini mengisi daftar kunci Select KMS. Kami merekomendasikan untuk menggunakan izin ini untuk memberikan pengalaman pengguna yang terbaik. Izin ini termasuk dalam kebijakan kunci default.

Kunci yang dikelola AWS untuk kebijakan Amazon EFS KMS

Kebijakan KMS JSON untuk Kunci yang dikelola AWS Amazon EFS, `aws/elasticfilesystem` adalah sebagai berikut:

```
{
  "Version": "2012-10-17",
  "Id": "auto-elasticfilesystem-1",
  "Statement": [
    {
      "Sid": "Allow access to EFS for all principals in the account that are
authorized to use EFS",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
    },
  ],
}
```

```

    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": "elasticfilesystem.us-east-2.amazonaws.com",
            "kms:CallerAccount": "111122223333"
        }
    }
},
{
    "Sid": "Allow direct access to key metadata to the account",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "kms:RevokeGrant"
    ],
    "Resource": "*"
}
]
}

```

Negara-negara kunci dan efeknya

Status kunci KMS Anda secara langsung memengaruhi akses ke sistem file terenkripsi Anda:

Diaktifkan

Operasi normal - akses baca dan tulis penuh ke sistem file

Nonaktif

Sistem file menjadi tidak dapat diakses setelah periode yang singkat. Dapat diaktifkan kembali.

Menunggu penghapusan

Sistem file menjadi tidak dapat diakses. Penghapusan dapat dibatalkan selama masa tunggu.

Dihapus

Sistem file tidak dapat diakses secara permanen. Tindakan ini tidak dapat dibalik.

Warning

Jika Anda menonaktifkan atau menghapus kunci KMS yang digunakan untuk sistem file Anda, atau mencabut akses Amazon EFS ke kunci, sistem file Anda akan menjadi tidak dapat diakses. Hal ini dapat mengakibatkan hilangnya data jika Anda tidak memiliki backup. Selalu pastikan Anda memiliki prosedur pencadangan yang tepat sebelum membuat perubahan pada kunci enkripsi.

Enkripsi pemecahan masalah

Berikut ini, Anda dapat menemukan informasi tentang pemecahan masalah enkripsi untuk Amazon EFS.

- [Pemasangan dengan enkripsi data dalam perjalanan gagal](#)
- [Pemasangan dengan enkripsi data dalam perjalanan terganggu](#)
- [Encrypted-at-rest sistem berkas tidak dapat dibuat](#)
- [Sistem file terenkripsi yang tidak dapat digunakan](#)

Pemasangan dengan enkripsi data dalam perjalanan gagal

Secara default, saat Anda menggunakan helper mount Amazon EFS dengan Transport Layer Security (TLS), ini akan memberlakukan pemeriksaan nama host. Beberapa sistem tidak mendukung fitur ini, seperti ketika Anda menggunakan Red Hat Enterprise Linux atau CentOS. Dalam kasus ini, pemasangan sistem file EFS menggunakan TLS gagal.

Tindakan yang harus diambil

Kami menyarankan Anda meningkatkan versi stunnel pada klien Anda untuk mendukung pemeriksaan nama host. Untuk informasi selengkapnya, lihat [Upgrade stunnel](#).

Pemasangan dengan enkripsi data dalam perjalanan terganggu

Mungkin saja, bagaimanapun tidak mungkin, koneksi terenkripsi Anda ke sistem file Amazon EFS Anda dapat macet atau terganggu oleh peristiwa sisi klien.

Tindakan yang harus diambil

Jika koneksi Anda ke sistem file Amazon EFS Anda dengan enkripsi data dalam perjalanan terputus, lakukan langkah-langkah berikut:

1. Pastikan bahwa layanan stunnel berjalan pada klien.
2. Konfirmasikan bahwa aplikasi `amazon-efs-mount-watchdog` pengawas berjalan pada klien. Anda dapat mengetahui apakah aplikasi ini berjalan dengan perintah berikut:

```
ps aux | grep [a]mazon-efs-mount-watchdog
```

3. Periksa log dukungan Anda. Untuk informasi selengkapnya, lihat [Mendapatkan log dukungan](#).
4. Secara opsional, Anda dapat mengaktifkan log stunnel Anda dan memeriksa informasi di dalamnya juga. Anda dapat mengubah konfigurasi log Anda `/etc/amazon/efs/efs-utils.conf` untuk mengaktifkan log stunnel. Namun, melakukan hal itu memerlukan pelepasan dan kemudian memasang kembali sistem file dengan mount helper agar perubahan diterapkan.

Important

Mengaktifkan log stunnel dapat menggunakan jumlah ruang yang tidak sepele pada sistem file Anda.

Jika interupsi berlanjut, hubungi Support AWS .

Encrypted-at-rest sistem berkas tidak dapat dibuat

Anda telah mencoba membuat sistem encrypted-at-rest file baru. Namun, Anda mendapatkan pesan kesalahan yang mengatakan bahwa AWS KMS itu tidak tersedia.

Tindakan yang harus diambil

Kesalahan ini dapat terjadi dalam kasus langka yang AWS KMS menjadi sementara tidak tersedia di Anda AWS Region. Jika ini terjadi, tunggu sampai AWS KMS kembali ke ketersediaan penuh, dan kemudian coba lagi untuk membuat sistem file.

Sistem file terenkripsi yang tidak dapat digunakan

Sistem file terenkripsi secara konsisten mengembalikan kesalahan server NFS. Kesalahan ini dapat terjadi ketika EFS tidak dapat mengambil kunci master Anda karena salah satu AWS KMS alasan berikut:

- Kuncinya dinonaktifkan.
- Kuncinya telah dihapus.
- Izin Amazon EFS untuk menggunakan kunci telah dicabut.
- AWS KMS sementara tidak tersedia.

Tindakan yang harus diambil

Pertama, konfirmasi bahwa AWS KMS kunci diaktifkan. Anda dapat melakukannya dengan melihat tombol di konsol. Untuk informasi selengkapnya, lihat [Melihat Kunci](#) di Panduan AWS Key Management Service Pengembang.

Jika kunci tidak diaktifkan, aktifkan. Untuk informasi selengkapnya, lihat [Mengaktifkan dan Menonaktifkan Kunci di Panduan](#) Pengembang.AWS Key Management Service

Jika kunci tertunda penghapusan, maka status ini menonaktifkan kunci. Anda dapat membatalkan penghapusan, dan mengaktifkan kembali kunci. Untuk informasi selengkapnya, lihat [Menjadwalkan dan Membatalkan Penghapusan Kunci](#) di Panduan Pengembang.AWS Key Management Service

Jika kunci diaktifkan, dan Anda masih mengalami masalah, atau jika Anda mengalami masalah saat mengaktifkan kembali kunci Anda, hubungi Support AWS .

Privasi antar jaringan

Topik ini menjelaskan bagaimana Amazon EFS mengamankan koneksi dari layanan ke lokasi lain.

Lalu lintas antara layanan dan aplikasi serta klien on-premise

Anda memiliki dua opsi konektivitas antara jaringan pribadi Anda dan AWS:

- AWS Site-to-Site VPN Koneksi. Untuk informasi lebih lanjut, lihat [Apa itu AWS Site-to-Site VPN?](#)
- Direct Connect Koneksi. Untuk informasi lebih lanjut, lihat [Apa itu Direct Connect?](#)

Akses ke Amazon EFS melalui jaringan melalui AWS dipublikasikan APIs. Klien harus mendukung Transport Layer 1.2 atau lebih tinggi. Kami merekomendasikan TLS 1.3 atau lebih tinggi. Klien juga harus mendukung suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern seperti Java 7 dan versi yang lebih baru support mode ini. Selain itu, Anda harus menandatangani permintaan menggunakan access key ID dan secret access key yang terkait dengan IAM utama, atau Anda dapat menggunakan [AWS Security Token Service \(AWS STS\)](#) guna membuat kredensial keamanan sementara untuk menandatangani permintaan.

Lalu lintas antara VPC dan Amazon EFS API

Untuk membuat koneksi pribadi antara virtual private cloud (VPC) dan Amazon EFS API, Anda dapat membuat antarmuka VPC endpoint. Anda dapat menggunakan koneksi ini untuk memanggil Amazon EFS API dari VPC Anda tanpa mengirim lalu lintas melalui internet. Endpoint menyediakan konektivitas aman ke Amazon EFS API tanpa memerlukan gateway internet, instans NAT, atau koneksi jaringan pribadi virtual (VPN). Untuk informasi selengkapnya, lihat [Bekerja dengan titik akhir VPC antarmuka di Amazon EFS](#).

Lalu lintas antar AWS sumber daya di Wilayah yang sama

Titik akhir Amazon Virtual Private Cloud (Amazon VPC) untuk Amazon EFS adalah entitas logis dalam VPC yang memungkinkan konektivitas hanya ke Amazon EFS. Amazon VPC merutekan permintaan ke Amazon EFS dan merutekan respons kembali ke VPC. Untuk informasi selengkapnya, lihat [VPC Endpoints](#) dalam Panduan Pengguna Amazon VPC.

Manajemen identitas dan akses untuk Amazon EFS

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 diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon EFS. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)

- [Bagaimana Amazon Elastic File System bekerja dengan IAM](#)
- [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)
- [Contoh kebijakan berbasis sumber daya untuk Amazon EFS](#)
- [AWS kebijakan terkelola untuk Amazon EFS](#)
- [Menggunakan tag dengan Amazon EFS](#)
- [Menggunakan peran terkait layanan untuk Amazon EFS](#)
- [Memecahkan masalah identitas dan akses Amazon Elastic File System](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat [Memecahkan masalah identitas dan akses Amazon Elastic File System](#))
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat [Bagaimana Amazon Elastic File System bekerja dengan IAM](#))
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#))

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensial dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensial. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat [Cara masuk ke Akun AWS Anda](#) dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat [AWS Signature Version 4 untuk permintaan API](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensial sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensial dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat [Apa itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat [Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensial sementara](#) di Panduan Pengguna IAM.

[Grup IAM](#) menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk pengguna IAM](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan [beralih dari pengguna ke peran IAM \(konsol\)](#) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat [Pilih 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. Contohnya termasuk 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. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya.

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.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Kebijakan kontrol sumber daya \(RCPs\)](#) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu 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 Amazon Elastic File System bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon EFS, pelajari fitur IAM yang tersedia untuk digunakan dengan Amazon EFS.

Fitur IAM yang dapat Anda gunakan dengan Amazon Elastic File System

Fitur IAM	Dukungan Amazon EFS
Kebijakan berbasis identitas	Ya

Fitur IAM	Dukungan Amazon EFS
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Ya
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon EFS dan AWS layanan lainnya dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk Amazon EFS

Mendukung kebijakan berbasis identitas: Ya

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 oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya

tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amazon EFS

Untuk melihat contoh kebijakan berbasis identitas Amazon EFS, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

Kebijakan berbasis sumber daya dalam Amazon EFS

Mendukung kebijakan berbasis sumber daya: Ya

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 tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Untuk mempelajari tentang menggunakan kebijakan sumber daya untuk mengontrol akses data sistem file, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#). Untuk mempelajari cara melampirkan kebijakan berbasis sumber daya ke sistem file, lihat. [Membuat kebijakan sistem file](#)

Contoh kebijakan berbasis sumber daya dalam Amazon EFS

Untuk melihat contoh kebijakan berbasis sumber daya Amazon EFS, lihat. [Contoh kebijakan berbasis sumber daya untuk Amazon EFS](#)

Tindakan kebijakan untuk Amazon EFS

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Amazon EFS, lihat [Tindakan yang ditentukan oleh Amazon Elastic File System](#) di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amazon EFS menggunakan awalan berikut sebelum tindakan:

```
elasticfilesystem
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [  
    "elasticfilesystem:action1",  
    "elasticfilesystem:action2"  
]
```

Untuk melihat contoh kebijakan berbasis identitas Amazon EFS, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

Sumber daya kebijakan untuk Amazon EFS

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya Amazon EFS dan jenisnya ARNs, lihat Sumber [daya yang ditentukan oleh Amazon Elastic File System](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh Amazon Elastic File System](#).

Untuk melihat contoh kebijakan berbasis identitas Amazon EFS, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

Kunci kondisi kebijakan untuk Amazon EFS

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi Amazon EFS, lihat [Kunci kondisi untuk Amazon Elastic File System](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh Amazon Elastic File System](#).

Untuk melihat contoh kebijakan berbasis identitas Amazon EFS, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

ACLs di Amazon EFS

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Amazon EFS

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat [Tentukan izin dengan otorisasi ABAC](#) dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) dalam Panduan Pengguna IAM.

Menggunakan kredensial sementara dengan Amazon EFS

Mendukung kredensial sementara: Ya

Kredensial sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensial sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensial keamanan sementara di IAM](#) dan [Layanan AWS yang berfungsi dengan IAM](#) dalam Panduan Pengguna IAM.

Izin utama lintas layanan untuk Amazon EFS

Mendukung sesi akses terusan (FAS): Ya

Sesi akses terusan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses terusan](#).

Peran layanan untuk Amazon EFS

Mendukung peran layanan: Ya

Peran layanan adalah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

 Warning

Mengubah izin untuk peran layanan dapat merusak fungsionalitas Amazon EFS. Edit peran layanan hanya jika Amazon EFS memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon EFS

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan 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.

Untuk detail tentang membuat atau mengelola peran terkait layanan Amazon EFS, lihat.

[Menggunakan peran terkait layanan untuk Amazon EFS](#)

Contoh kebijakan berbasis identitas untuk Amazon Elastic File System

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon EFS. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM \(konsol\) di Panduan Pengguna IAM](#).

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon EFS, termasuk format ARNs untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Amazon Elastic File System](#) dalam Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol Amazon EFS](#)
- [Contoh: Izinkan pengguna untuk melihat izin mereka sendiri](#)
- [Contoh: Menegakkan pembuatan sistem file terenkripsi](#)
- [Contoh: Menegakkan pembuatan sistem file yang tidak terenkripsi](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon EFS di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. 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 tugas](#) dalam 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 melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat [Kebijakan dan izin dalam IAM](#) dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu 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 CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Kondisi](#) dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – 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 dengan IAM Access Analyzer](#) dalam 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 meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Amankan akses API dengan MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.

Menggunakan konsol Amazon EFS

Untuk mengakses konsol Amazon Elastic File System, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon EFS di situs Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Amazon EFS, lampirkan juga kebijakan `AmazonElasticFileSystemReadOnlyAccess` AWS terkelola Amazon EFS ke entitas. Untuk informasi selengkapnya, lihat [Menambah izin untuk pengguna](#) dalam Panduan Pengguna IAM.

Anda dapat melihat `AmazonElasticFileSystemReadOnlyAccess` dan kebijakan layanan terkelola Amazon EFS lainnya di [AWS kebijakan terkelola untuk Amazon EFS](#).

Contoh: Izinkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan 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": "*"
    }
]
}

```

Contoh: Menegakkan pembuatan sistem file terenkripsi

Contoh berikut menggambarkan kebijakan berbasis identitas yang mengotorisasi prinsipal untuk membuat hanya sistem file terenkripsi.

```

{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticfilesystem:CreateFileSystem",
            "Condition": {
                "Bool": {
                    "elasticfilesystem:Encrypted": "true"
                }
            },
            "Resource": "*"
        }
    ]
}

```

```

    }
  ]
}
```

Jika kebijakan ini ditetapkan ke pengguna yang mencoba membuat sistem file yang tidak terenkripsi, permintaan akan gagal. Pengguna melihat pesan yang mirip dengan yang berikut ini, apakah mereka menggunakan Konsol Manajemen AWS, AWS API AWS CLI, atau SDK:

```
User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

Contoh: Menegakkan pembuatan sistem file yang tidak terenkripsi

Contoh berikut mengilustrasikan kebijakan berbasis identitas yang mengizinkan prinsipal untuk membuat hanya sistem file yang tidak terenkripsi.

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticfilesystem:CreateFileSystem",
      "Condition": {
        "Bool": {
          "elasticfilesystem:Encrypted": "false"
        }
      },
      "Resource": "*"
    }
  ]
}
```

Jika kebijakan ini ditetapkan ke pengguna yang mencoba membuat sistem file terenkripsi, permintaan akan gagal. Pengguna melihat pesan yang mirip dengan yang berikut ini, apakah mereka menggunakan Konsol Manajemen AWS, AWS API AWS CLI, atau SDK:

```
User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

Anda juga dapat menerapkan pembuatan sistem file EFS terenkripsi atau tidak terenkripsi dengan membuat kebijakan kontrol AWS Organizations layanan (SCP). Untuk informasi selengkapnya

tentang kebijakan kontrol [layanan AWS Organizations](#), lihat [Kebijakan kontrol layanan](#) di Panduan AWS Organizations Pengguna.

Contoh kebijakan berbasis sumber daya untuk Amazon EFS

Di bagian ini, Anda dapat menemukan contoh kebijakan sistem file yang memberikan atau menolak izin untuk berbagai tindakan Amazon EFS. Kebijakan sistem file EFS memiliki batas 20.000 karakter. Untuk informasi tentang elemen kebijakan berbasis sumber daya, lihat. [Kebijakan berbasis sumber daya dalam Amazon EFS](#)

Important

Jika Anda memberikan izin kepada pengguna IAM individu atau peran dalam kebijakan sistem file, jangan menghapus atau membuat ulang pengguna atau peran tersebut saat kebijakan berlaku pada sistem file. Jika ini terjadi, pengguna atau peran itu secara efektif dikunci dari sistem file dan tidak akan dapat mengaksesnya. Untuk informasi selengkapnya, lihat [Menentukan Principal](#) di Panduan Pengguna IAM.

Untuk informasi tentang cara membuat kebijakan sistem file, lihat [Membuat kebijakan sistem file](#).

Topik

- [Contoh: Berikan akses baca dan tulis ke AWS peran tertentu](#)
- [Contoh: Berikan akses hanya-baca](#)
- [Contoh: Berikan akses ke titik akses EFS](#)

Contoh: Berikan akses baca dan tulis ke AWS peran tertentu

Dalam contoh ini, kebijakan sistem file EFS memiliki karakteristik sebagai berikut:

- Efeknya adalah `Allow`.
- Prinsipal diatur ke `Testing_Role` di Akun AWS
- Tindakan diatur ke `ClientMount` (baca), dan `ClientWrite`.
- Kondisi untuk memberikan izin diatur ke `AccessedViaMountTarget`

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/Testing_Role"
      },
      "Action": [
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-system/
fs-1234abcd",
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    }
  ]
}
```

Contoh: Berikan akses hanya-baca

Kebijakan sistem file berikut hanya memberikan `ClientMount`, atau hanya-baca, izin untuk peran IAM. `EfsReadOnly`

```
{
  "Id": "read-only-example-policy02",
  "Statement": [
    {
      "Sid": "efs-statement-example02",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
      },
      "Action": [
        "elasticfilesystem:ClientMount"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-system/
fs-12345678"
    }
  ]
}
```

}

Untuk mempelajari cara menyetel kebijakan sistem file tambahan, termasuk menolak akses root ke semua prinsipal IAM, kecuali untuk workstation manajemen tertentu, lihat [Aktifkan root squashing menggunakan otorisasi IAM untuk klien NFS](#)

Contoh: Berikan akses ke titik akses EFS

Anda menggunakan kebijakan akses EFS untuk menyediakan klien NFS dengan tampilan khusus aplikasi ke dalam kumpulan data berbasis file bersama pada sistem file EFS. Anda memberikan izin titik akses pada sistem file menggunakan kebijakan sistem file.

Contoh kebijakan file ini menggunakan elemen kondisi untuk memberikan titik akses tertentu yang diidentifikasi oleh ARN akses penuh ke sistem file.

Untuk informasi selengkapnya tentang penggunaan titik akses EFS, lihat [Bekerja dengan titik akses](#).

```
{
  "Id": "access-point-example03",
  "Statement": [
    {
      "Sid": "access-point-statement-example03",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::555555555555:role/EfsAccessPointFullAccess"},
      "Action": "elasticfilesystem:Client*",
      "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-system/fs-12345678",
      "Condition": {
        "StringEquals": {
          "elasticfilesystem:AccessPointArn": "arn:aws:elasticfilesystem:us-east-2:555555555555:access-point/fsap-12345678" }
        }
      }
    ]
  }
```

AWS kebijakan terkelola untuk Amazon EFS

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan [kebijakan yang dikelola pelanggan](#) yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat [Kebijakan terkelola AWS](#) dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AWSService RoleForAmazonElasticFileSystem

Amazon EFS menggunakan peran terkait layanan yang diberi nama `AWSServiceRoleForAmazonElasticFileSystem` untuk memungkinkan Amazon EFS mengelola AWS sumber daya atas nama Anda. Peran ini mempercayai layanan `elasticfilesystem.amazonaws.com` untuk mengasumsikan peran tersebut. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk Amazon EFS](#).

AWS kebijakan terkelola: AmazonElasticFileSystemFullAccess

Anda dapat melampirkan kebijakan `AmazonElasticFileSystemFullAccess` ke identitas IAM Anda.

Kebijakan ini memberikan izin administratif yang memungkinkan akses penuh ke Amazon EFS dan akses ke AWS layanan terkait melalui. Konsol Manajemen AWS

Detail izin

Kebijakan ini mencakup izin berikut.

- `elasticfilesystem`— Memungkinkan prinsipal untuk melakukan semua tindakan di konsol Amazon EFS. Hal ini juga memungkinkan prinsipal untuk membuat (`elasticfilesystem:Backup`) dan restore (`elasticfilesystem:Restore`) backup menggunakan. AWS Backup
- `cloudwatch`— Memungkinkan prinsipal untuk menggambarkan metrik sistem CloudWatch file Amazon dan alarm untuk metrik di konsol Amazon EFS.

- `ec2`— Memungkinkan prinsipal untuk membuat, menghapus, dan mendeskripsikan antarmuka jaringan, mendeskripsikan dan memodifikasi atribut antarmuka jaringan, menjelaskan Availability Zone, grup keamanan, subnet, virtual private cloud (VPCs) dan atribut VPC yang terkait dengan sistem file EFS di konsol Amazon EFS.
- `kms`— Memungkinkan prinsipal untuk membuat daftar alias untuk AWS Key Management Service (AWS KMS) kunci dan mendeskripsikan kunci KMS di konsol Amazon EFS.
- `iam`— Memberikan izin untuk membuat peran terkait layanan yang memungkinkan Amazon EFS mengelola AWS sumber daya atas nama pengguna.
- `iam:PassRole`— Memberikan izin untuk meneruskan peran IAM ke Amazon EFS.

Untuk melihat izin kebijakan ini, lihat [AmazonElasticFileSystemFullAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: `AmazonElasticFileSystemReadOnlyAccess`

Anda dapat melampirkan kebijakan `AmazonElasticFileSystemReadOnlyAccess` ke identitas IAM Anda.

Kebijakan ini memberikan akses baca saja ke Amazon EFS melalui Konsol Manajemen AWS

Detail izin

Kebijakan ini mencakup izin berikut.

- `elasticfilesystem`— Memungkinkan prinsipal untuk mendeskripsikan atribut sistem file Amazon EFS, termasuk preferensi akun, kebijakan pencadangan dan sistem file, konfigurasi siklus hidup, target pemasangan dan grup keamanan, tag, dan titik aksesnya di konsol Amazon EFS.
- `cloudwatch`— Memungkinkan kepala sekolah untuk mengambil CloudWatch metrik dan menjelaskan alarm untuk metrik di konsol Amazon EFS.
- `ec2`— Memungkinkan prinsipal untuk melihat Availability Zone, antarmuka jaringan dan atributnya, grup keamanan, subnet, VPCs dan atributnya di konsol Amazon EFS.
- `kms`— Memungkinkan prinsipal untuk membuat daftar alias untuk kunci AWS KMS di konsol Amazon EFS.

Untuk melihat izin kebijakan ini, lihat [AmazonElasticFileSystemReadOnlyAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonElasticFileSystemClientFullAccess

Anda dapat melampirkan AmazonElasticFileSystemClientFullAccess kebijakan ke entitas IAM.

Kebijakan ini memberikan akses baca dan tulis klien ke sistem file EFS. Kebijakan ini memungkinkan klien NFS untuk me-mount, membaca, dan menulis ke sistem file EFS.

Untuk melihat izin kebijakan ini, lihat [AmazonElasticFileSystemClientFullAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonElasticFileSystemClientReadWriteAccess

Anda dapat melampirkan AmazonElasticFileSystemClientReadWriteAccess kebijakan ke entitas IAM.

Kebijakan ini memberikan akses baca dan tulis klien ke sistem file EFS. Kebijakan ini memungkinkan klien NFS untuk me-mount, membaca, dan menulis ke sistem file EFS.

Untuk melihat izin kebijakan ini, lihat [AmazonElasticFileSystemClientReadWriteAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

Amazon EFS memperbarui kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon EFS sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Amazon EFS [Riwayat dokumen](#).

Ubah	Deskripsi	Date
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemFullAccess Amazon EFS menambahkan yang berikut ini: • <code>ReplicationRead</code> dan <code>ReplicationWrite</code> untuk memberikan izin untuk membaca dan menulis data sistem file untuk replikasi. • <code>iam:PassRole</code> untuk memberikan izin kepada Amazon EFS untuk membuat konfigurasi replikasi	November 7, 2024

Ubah	Deskripsi	Date
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemServiceRolePolicy Amazon EFS menambahkan <code>ReplicationRead</code> dan <code>ReplicationWrite</code> memberikan izin untuk membaca dan menulis data sistem file untuk replikasi.	November 7, 2024
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemReadOnlyAccess Amazon EFS menambahkan <code>ReplicationRead</code> tindakan untuk memberikan izin membaca data sistem file untuk replikasi.	November 7, 2024
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemReadOnlyAccess Amazon EFS menambahkan izin baru yang memberikan akses akun sumber dan tujuan ke sistem file untuk replikasi lintas akun.	Agustus 7, 2024
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemFullAccess Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal menonaktifkan dan mengaktifkan perlindungan pada sistem file. Izin diperlukan untuk memungkinkan Amazon EFS mereplikasi ke sistem file yang ada.	27 November 2023

Ubah	Deskripsi	Date
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemServiceRolePolicy Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal membuat, mendeskripsikan, dan menghapus replikasi Amazon EFS, dan membuat sistem file Amazon EFS. Izin diperlukan untuk memungkinkan Amazon EFS mengelola konfigurasi replikasi sistem file atas nama pengguna.	Januari 25, 2022
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemReadOnlyAccess Amazon EFS menambahkan izin baru untuk mengizinkan kepala sekolah mendeskripsikan replikasi Amazon EFS. Izin diperlukan untuk memungkinkan pengguna melihat konfigurasi replikasi sistem file.	Januari 25, 2022
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemFullAccess Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal membuat, mendeskripsikan, dan menghapus replikasi Amazon EFS. Izin diperlukan untuk memungkinkan pengguna mengelola konfigurasi replikasi sistem file.	Januari 25, 2022
Memulai kebijakan pelacakan	Kebijakan: AmazonElasticFileSystemClientReadWriteAccess Memberikan hak baca dan tulis di sistem file Amazon EFS kepada klien NFS.	Januari 3, 2022

Ubah	Deskripsi	Date
Memulai kebijakan pelacakan	Kebijakan: AmazonElasticFileSystemServiceRolePolicy Izin peran terkait layanan untuk Amazon EFS.	Oktober 8, 2021
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemFullAccess Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal memodifikasi dan menjelaskan preferensi akun Amazon EFS. Izin diperlukan untuk memungkinkan pengguna melihat dan mengatur pengaturan preferensi akun di konsol Amazon EFS.	7 Mei 2021
Memperbarui ke kebijakan yang ada	Kebijakan: AmazonElasticFileSystemReadOnlyAccess Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal menjelaskan preferensi i akun Amazon EFS. Izin diperlukan untuk memungkinkan pengguna melihat pengaturan preferensi akun di konsol Amazon EFS.	7 Mei 2021
Amazon EFS mulai melacak perubahan	Amazon EFS mulai melacak perubahan untuk kebijakan AWS terkelolanya.	7 Mei 2021

Menggunakan tag dengan Amazon EFS

Anda dapat menggunakan tag untuk mengontrol akses ke sumber daya Amazon EFS dan menerapkan kontrol akses berbasis atribut (ABAC). Untuk informasi lebih lanjut, lihat:

- [Menandai sumber daya EFS](#)
- [Mengontrol akses berdasarkan tag pada sumber daya](#)
- [Untuk apa ABAC? AWS](#) di Panduan Pengguna IAM

 Note

Replikasi Amazon EFS tidak mendukung penggunaan tag untuk kontrol akses berbasis atribut (ABAC).

Untuk menerapkan tag ke resource Amazon EFS selama pembuatan, pengguna harus memiliki izin AWS Identity and Access Management (IAM) tertentu.

Memberikan izin untuk menandai sumber daya selama pembuatan

Tindakan tag-on create Amazon EFS API berikut memungkinkan Anda menentukan tag saat membuat resource.

- `CreateAccessPoint`
- `CreateFileSystem`

Untuk memungkinkan pengguna menandai sumber daya pada pembuatan, mereka harus memiliki izin untuk menggunakan tindakan yang membuat sumber daya, seperti `elasticfilesystem:CreateAccessPoint` atau `elasticfilesystem:CreateFileSystem`. Jika tag ditentukan dalam tindakan pembuatan sumber daya, AWS lakukan otorisasi tambahan pada `elasticfilesystem:TagResource` tindakan untuk memverifikasi apakah pengguna memiliki izin untuk membuat tag. Oleh karena itu, para pengguna juga harus memiliki izin eksplisit untuk menggunakan tindakan `elasticfilesystem:TagResource`.

Di dalam definisi kebijakan IAM untuk tindakan `elasticfilesystem:TagResource`, gunakan elemen `Condition` dengan kunci syarat `elasticfilesystem:CreateAction` untuk memberikan izin pemberian tanda pada tindakan yang membuat sumber daya.

Example policy: Izinkan menambahkan tag ke sistem file hanya pada saat pembuatan

Contoh kebijakan berikut memungkinkan pengguna untuk membuat sistem file dan menerapkan tag kepada mereka hanya selama pembuatan. Pengguna tidak diizinkan untuk menandai sumber daya yang sudah ada (mereka tidak dapat memanggil tindakan `elasticfilesystem:TagResource` secara langsung).

```
{  
  "Statement": [  

```

```
{
  "Effect": "Allow",
  "Action": [
    "elasticfilesystem:CreateFileSystem"
  ],
  "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*"
},
{
  "Effect": "Allow",
  "Action": [
    "elasticfilesystem:TagResource"
  ],
  "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*",
  "Condition": {
    "StringEquals": {
      "elasticfilesystem:CreateAction": "CreateFileSystem"
    }
  }
}
]
```

Menggunakan tag untuk mengontrol akses ke sumber daya Amazon EFS Anda

Untuk mengontrol akses ke sumber daya dan tindakan Amazon EFS, Anda dapat menggunakan kebijakan IAM berdasarkan tag. Anda dapat memberikan kontrol ini dengan dua cara:

- Anda dapat mengontrol akses ke sumber daya Amazon EFS berdasarkan tag pada sumber daya tersebut.
- Anda dapat mengontrol tag mana yang dapat diteruskan dalam kondisi permintaan IAM.

Untuk informasi tentang cara menggunakan tag untuk mengontrol akses ke AWS sumber daya, lihat [Mengontrol akses menggunakan tag](#) di Panduan Pengguna IAM.

Mengontrol akses berdasarkan tag pada sumber daya

Untuk mengontrol tindakan yang dapat dilakukan pengguna atau peran pada sumber daya Amazon EFS, Anda dapat menggunakan tag pada sumber daya. Misalnya, Anda mungkin ingin mengizinkan atau menolak operasi API tertentu pada sumber daya sistem file berdasarkan pasangan nilai kunci tag pada sumber daya.

Example kebijakan: Buat sistem file hanya ketika tag tertentu digunakan

Contoh kebijakan berikut memungkinkan pengguna untuk membuat sistem file hanya ketika mereka menandainya dengan pasangan nilai kunci tag tertentu, dalam contoh ini, `key=Department.value=Finance`

```
{
  "Effect": "Allow",
  "Action": [
    "elasticfilesystem:CreateFileSystem",
    "elasticfilesystem:TagResource"
  ],
  "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*",
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/Department": "Finance"
    }
  }
}
```

Example kebijakan: Hapus sistem file dengan tag tertentu

Contoh kebijakan berikut memungkinkan pengguna untuk menghapus hanya sistem file yang ditandai dengan `Department=Finance`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem>DeleteFileSystem"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "Finance"
        }
      }
    }
  ]
}
```

```
}  
]  
}
```

Menggunakan peran terkait layanan untuk Amazon EFS

Amazon EFS menggunakan peran AWS Identity and Access Management [terkait layanan](#) (IAM). Peran terkait layanan Amazon EFS adalah jenis peran IAM unik yang ditautkan langsung ke Amazon EFS. Peran terkait layanan Amazon EFS yang telah ditentukan sebelumnya mencakup izin yang diperlukan layanan untuk memanggil orang lain Layanan AWS atas nama Anda.

Peran terkait layanan membuat pengaturan Amazon EFS lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Amazon EFS mendefinisikan izin peran terkait layanannya, dan hanya Amazon EFS yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran terkait layanan Amazon EFS hanya setelah pertama kali menghapus sistem file Amazon EFS Anda. Ini melindungi sumber daya Amazon EFS Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Peran terkait layanan memungkinkan semua panggilan API terlihat. AWS CloudTrail Ini membantu memantau dan mengaudit persyaratan karena Anda dapat melacak semua tindakan yang dilakukan Amazon EFS atas nama Anda. Untuk informasi selengkapnya, lihat [Entri log untuk peran terkait layanan EFS](#).

Untuk informasi selengkapnya, lihat [izin peran terkait layanan](#) dalam Panduan Pengguna IAM.

Izin peran terkait layanan untuk Amazon EFS

Amazon EFS menggunakan peran terkait layanan yang diberi nama `AWSServiceRoleForAmazonElasticFileSystem` untuk memungkinkan Amazon EFS memanggil dan mengelola AWS sumber daya atas nama sistem file EFS Anda.

Peran `AWSServiceRoleForAmazonElasticFileSystem` terkait layanan mempercayai `elasticfilesystem.amazonaws.com` untuk mengambil peran.

Kebijakan izin peran memungkinkan Amazon EFS menyelesaikan tindakan yang disertakan dalam definisi kebijakan JSON:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "backup-storage:MountCapsule",
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:ModifyNetworkInterfaceAttribute",
        "tag:GetResources"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:*:*:key/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "backup:CreateBackupVault",
        "backup:PutBackupVaultAccessPolicy"
      ],
      "Resource": [
        "arn:aws:backup:*:*:backup-vault:aws/efs/automatic-backup-vault"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "backup:CreateBackupPlan",
        "backup:CreateBackupSelection"
      ],
      "Resource": [
```

```

        "arn:aws:backup:*:*:backup-plan:*"
    ],
    },
    {
        "Effect": "Allow",
        "Action": [
            "iam:CreateServiceLinkedRole"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:AWSServiceName": [
                    "backup.amazonaws.com"
                ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "iam:PassRole"
        ],
        "Resource": [
            "arn:aws:iam:*:*:role/aws-service-role/backup.amazonaws.com/AWSServiceRoleForBackup"
        ],
        "Condition": {
            "StringLike": {
                "iam:PassedToService": "backup.amazonaws.com"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticfilesystem:DescribeFileSystems",
            "elasticfilesystem:CreateReplicationConfiguration",
            "elasticfilesystem:DescribeReplicationConfigurations",
            "elasticfilesystem>DeleteReplicationConfiguration",
            "elasticfilesystem:ReplicationRead",
            "elasticfilesystem:ReplicationWrite"
        ],
        "Resource": "*"
    }
}

```

```
}  
]
```

Note

Anda harus mengonfigurasi izin IAM secara manual AWS KMS saat membuat sistem file EFS baru yang dienkripsi saat istirahat. Untuk mempelajari selengkapnya, lihat [Mengenkripsi data diam](#).

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Izin peran yang terkait dengan layanan](#) dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Amazon EFS

Dalam kebanyakan kasus, Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat target mount atau konfigurasi replikasi untuk sistem file EFS Anda di Konsol Manajemen AWS, the AWS CLI, atau AWS API, Amazon EFS membuat peran terkait layanan untuk Anda.

Selain itu, jika Anda menghapus ini secara manual service-linked-role, dan kemudian perlu membuatnya lagi, Anda dapat menggunakan proses yang sama untuk membuat ulang peran di akun Anda. Saat Anda membuat target pemasangan atau konfigurasi replikasi untuk sistem file EFS Anda, Amazon EFS membuat peran terkait layanan untuk Anda.

Namun, jika Amazon EFS tidak membuat service-linked-role atau jika Anda mulai menggunakan Amazon EFS sebelum mendukung peran terkait layanan, Anda dapat membuat peran terkait layanan secara manual. Untuk petunjuknya, lihat [Membuat peran terkait layanan di Panduan](#) Pengguna IAM.

Mengedit peran terkait layanan untuk Amazon EFS

Amazon EFS tidak mengizinkan Anda mengedit peran `AWSServiceRoleForAmazonElasticFileSystem` terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Memperbarui peran terkait layanan](#) di Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Amazon EFS

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual. Untuk informasi selengkapnya, lihat [Bersihkan sumber daya dan lindungi AWS akun Anda](#).

Note

Jika layanan Amazon EFS menggunakan peran saat Anda mencoba menghapus sumber daya, penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran AWSService RoleForAmazonElasticFileSystem terkait layanan. Untuk informasi selengkapnya, lihat [Menghapus peran terkait layanan](#) dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk peran terkait layanan Amazon EFS

Amazon EFS mendukung penggunaan peran terkait layanan di semua Wilayah AWS tempat layanan tersedia. Untuk informasi selengkapnya, lihat [titik akhir AWS layanan](#) di Panduan Referensi Umum AWS Pengguna.

Memecahkan masalah identitas dan akses Amazon Elastic File System

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon EFS dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di Amazon EFS](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS mengakses sumber daya Amazon EFS saya](#)

Saya tidak berwenang untuk melakukan tindakan di Amazon EFS

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `elasticfilesystem:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticfilesystem:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `elasticfilesystem:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon EFS.

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 untuk melakukan tindakan di Amazon EFS. 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.

Saya ingin mengizinkan orang di luar saya Akun AWS mengakses sumber daya Amazon EFS saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang 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 (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah Amazon EFS mendukung fitur-fitur ini, lihat [Bagaimana Amazon Elastic File System 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 [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM](#).

Menggunakan IAM untuk mengontrol akses ke sistem file

Anda dapat menggunakan kebijakan identitas IAM dan kebijakan sumber daya untuk mengontrol akses klien ke sumber daya Amazon EFS dengan cara yang dapat diskalakan dan dioptimalkan untuk lingkungan cloud. Menggunakan IAM, Anda dapat mengizinkan klien untuk melakukan tindakan tertentu pada sistem file, termasuk akses read-only, write, dan root. Izin “izinkan” pada suatu tindakan baik dalam kebijakan identitas IAM atau kebijakan sumber daya sistem file memungkinkan akses untuk tindakan tersebut. Izin tidak perlu diberikan baik dalam identitas maupun kebijakan sumber daya.

Klien NFS dapat mengidentifikasi diri mereka menggunakan peran IAM saat menghubungkan ke sistem file EFS. Saat klien terhubung ke sistem file, Amazon EFS mengevaluasi kebijakan sumber

daya IAM sistem file, yang disebut kebijakan sistem file, bersama dengan kebijakan IAM berbasis identitas apa pun untuk menentukan izin akses sistem file yang sesuai untuk diberikan.

Saat Anda menggunakan otorisasi IAM untuk klien NFS, koneksi klien dan keputusan otorisasi IAM dicatat. AWS CloudTrail Untuk informasi selengkapnya tentang cara mencatat panggilan Amazon EFS API CloudTrail, lihat [Mencatat panggilan Amazon EFS API dengan AWS CloudTrail](#).

Important

Anda harus menggunakan EFS mount helper untuk memasang sistem file EFS Anda agar dapat menggunakan otorisasi IAM untuk mengontrol akses klien. Untuk informasi selengkapnya, lihat [Pemasangan dengan otorisasi IAM](#).

Kebijakan sistem file EFS default

Kebijakan sistem file EFS default tidak menggunakan IAM untuk mengautentikasi, dan memberikan akses penuh ke klien anonim mana pun yang dapat terhubung ke sistem file menggunakan target pemasangan. Kebijakan default berlaku setiap kali kebijakan sistem file yang dikonfigurasi pengguna tidak berlaku, termasuk pada pembuatan sistem file. Setiap kali kebijakan sistem file default berlaku, operasi [DescribeFileSystemPolicy](#) API mengembalikan PolicyNotFound respons.

Tindakan EFS untuk klien

Anda dapat menentukan tindakan berikut untuk klien yang mengakses sistem file menggunakan kebijakan sistem file.

Tindakan	Deskripsi
<code>elasticfilesystem:ClientMount</code>	Menyediakan akses read-only ke sistem file.
<code>elasticfilesystem:ClientWrite</code>	Memberikan izin menulis pada sistem file.
<code>elasticfilesystem:ClientRootAccess</code>	Menyediakan penggunaan pengguna root saat mengakses sistem file.

Kunci kondisi EFS untuk klien

Untuk menyatakan kondisi, Anda menggunakan kunci kondisi standar. Amazon EFS memiliki kunci kondisi standar berikut untuk klien NFS. Kunci kondisi lainnya tidak diberlakukan saat menggunakan kontrol IAM untuk mengamankan akses ke sistem file EFS.

Kunci Kondisi EFS	Deskripsi	Operator
<code>aws:SecureTransport</code>	Gunakan kunci ini untuk meminta klien menggunakan TLS saat menghubungkan ke sistem file EFS.	Boolean
<code>aws:SourceIp</code>	Gunakan kunci ini untuk membandingkan alamat IP pemohon dengan alamat IP yang Anda tentukan dalam kebijakan. Kunci kondisi <code>aws:SourceIp</code> hanya dapat digunakan untuk rentang alamat IP publik.	String
<code>elasticfilesystem:AccessPointArn</code>	ARN dari titik akses EFS yang terhubung dengan klien.	String
<code>elasticfilesystem:AccessedViaMountTarget</code>	Gunakan kunci ini untuk mencegah akses ke sistem file EFS oleh klien yang tidak menggunakan target pemasangan sistem file.	Boolean

Contoh kebijakan sistem file

Untuk melihat contoh kebijakan sistem file Amazon EFS, lihat [Contoh kebijakan berbasis sumber daya untuk Amazon EFS](#).

Validasi kepatuhan untuk Amazon EFS

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. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat [Dokumentasi AWS Keamanan](#).

Ketahanan di Amazon EFS

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones (AZs). Wilayah AWS menyediakan beberapa jaringan yang terpisah dan terisolasi secara fisik AZs, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan AZs, Anda dapat merancang dan mengoperasikan aplikasi dan database yang secara otomatis gagal di antara zona tanpa gangguan. AZs lebih tersedia, toleran terhadap kesalahan, dan skalabel daripada infrastruktur pusat data tunggal atau ganda tradisional.

Sistem file Amazon EFS tahan terhadap satu atau beberapa kegagalan Availability Zone dalam file. AWS Region Target mount sendiri dirancang agar sangat tersedia. Saat Anda merancang ketersediaan tinggi dan failover ke yang lain AZs, perlu diingat bahwa meskipun alamat IP dan DNS untuk target pemasangan Anda di setiap AZ bersifat statis, mereka adalah komponen redundan yang didukung oleh beberapa sumber daya. Untuk informasi selengkapnya, lihat [Bagaimana Amazon EFS bekerja dengan Amazon EC2](#).

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Mengontrol akses jaringan ke sistem file EFS untuk klien NFS

Anda dapat mengontrol akses oleh klien NFS ke sistem file Amazon EFS menggunakan keamanan lapisan jaringan dan kebijakan sistem file EFS. Anda dapat menggunakan mekanisme keamanan lapisan jaringan yang tersedia dengan Amazon EC2, seperti aturan grup keamanan VPC dan

jaringan. ACLs Anda juga dapat menggunakan AWS IAM untuk mengontrol akses NFS dengan kebijakan sistem file EFS dan kebijakan berbasis identitas.

Topik

- [Menggunakan grup keamanan VPC](#)
- [Bekerja dengan titik akhir VPC antarmuka di Amazon EFS](#)

Menggunakan grup keamanan VPC

Saat menggunakan Amazon EFS, Anda menentukan grup keamanan VPC untuk instans EC2 dan grup keamanan untuk target pemasangan EFS yang terkait dengan sistem file. Grup keamanan bertindak sebagai firewall, dan aturan yang Anda tambahkan menentukan arus lalu lintas. Dalam [latihan Memulai](#), Anda membuat satu grup keamanan saat meluncurkan instans EFS. Anda kemudian mengaitkan yang lain dengan target pemasangan EFS (yaitu, grup keamanan default untuk VPC default Anda). Pendekatan itu bekerja untuk latihan Memulai. Namun, untuk sistem produksi, Anda harus menyiapkan grup keamanan dengan izin minimal untuk digunakan dengan Amazon EFS.

Anda dapat mengotorisasi akses masuk dan keluar ke sistem file EFS Anda. Untuk melakukannya, Anda menambahkan aturan yang memungkinkan instance EFS terhubung ke sistem file EFS Anda melalui target pemasangan menggunakan port Network File System (NFS).

- Setiap instans EC2 yang memasang sistem file harus memiliki grup keamanan dengan aturan yang memungkinkan akses keluar ke target pemasangan pada port NFS 2049.
- Target pemasangan EFS harus memiliki grup keamanan dengan aturan yang memungkinkan akses masuk pada port NFS 2049 dari setiap instans EC2 tempat Anda ingin memasang sistem file.

Tabel berikut menunjukkan aturan grup keamanan khusus yang diperlukan:

Grup Keamanan	Jenis Aturan	Protokol	Port	Sumber/Tujuan
Instans EC2	Ke luar	TCP	2049	Pasang kelompok keamanan target
Target Gunung	Ke dalam	TCP	2049	Grup keamanan instans EC2

Port sumber untuk bekerja dengan Amazon EFS

Untuk mendukung serangkaian klien NFS yang luas, Amazon EFS memungkinkan koneksi dari port sumber apa pun. Jika Anda mengharuskan hanya pengguna istimewa yang dapat mengakses Amazon EFS, sebaiknya gunakan aturan firewall klien berikut. Connect ke sistem file Anda menggunakan SSH dan jalankan perintah berikut:

```
iptables -I OUTPUT 1 -m owner --uid-owner 1-4294967294 -m tcp -p tcp --dport 2049 -j DROP
```

Perintah ini menyisipkan aturan baru di awal rantai OUTPUT (-I OUTPUT 1). Aturan ini mencegah proses nonkernel (-m owner --uid-owner 1-4294967294) yang tidak memiliki hak istimewa membuka koneksi ke port NFS 2049 (). -m tcp -p tcp -dport 2049

Pertimbangan keamanan untuk akses jaringan

Klien NFS versi 4.1 (NFSv4.1) hanya dapat me-mount sistem file jika dapat membuat koneksi jaringan ke port NFS (port TCP 2049) dari salah satu target mount sistem file. Demikian pula, klien NFSv4 .1 hanya dapat menegaskan ID pengguna dan grup saat mengakses sistem file jika dapat membuat koneksi jaringan ini.

Apakah Anda dapat membuat koneksi jaringan ini diatur oleh kombinasi berikut ini:

- Isolasi jaringan yang disediakan oleh VPC target mount — Target pemasangan sistem file tidak dapat memiliki alamat IP publik yang terkait dengannya. Satu-satunya target yang dapat me-mount sistem file adalah sebagai berikut:
 - Instans Amazon EC2 di VPC Amazon lokal
 - Instans EC2 terhubung VPCs
 - Server lokal yang terhubung ke VPC Amazon dengan AWS Direct Connect menggunakan dan (VPN AWS Virtual Private Network)
- Daftar kontrol akses jaringan (ACLs) untuk subnet VPC klien dan target mount, untuk akses dari luar subnet target mount — Untuk memasang sistem file, klien harus dapat membuat koneksi TCP ke port NFS 2049 dari target mount dan menerima lalu lintas kembali.
- Aturan grup keamanan VPC klien dan target mount, untuk semua akses — Agar instans EC2 memasang sistem file, aturan grup keamanan berikut harus berlaku:
 - Sistem file harus memiliki target mount yang antarmuka jaringannya memiliki grup keamanan dengan aturan yang memungkinkan koneksi masuk pada port NFS 2049 dari instance. Anda

dapat mengaktifkan koneksi masuk baik dengan alamat IP (rentang CIDR) atau grup keamanan. Sumber aturan grup keamanan untuk port NFS masuk pada antarmuka jaringan target mount adalah elemen kunci dari kontrol akses sistem file. Aturan masuk selain yang untuk port NFS 2049, dan aturan keluar apa pun, tidak digunakan oleh antarmuka jaringan untuk target pemasangan sistem file.

- Instans pemasangan harus memiliki antarmuka jaringan dengan aturan grup keamanan yang memungkinkan koneksi keluar ke port NFS 2049 pada salah satu target pemasangan sistem file. Anda dapat mengaktifkan koneksi outbound menurut alamat IP (kisaran CIDR) atau grup keamanan.

Untuk informasi selengkapnya, lihat [Mengelola target mount](#).

Membuat grup keamanan

Membuat grup keamanan untuk instans EC2 dan target pemasangan EFS

Berikut ini adalah langkah-langkah umum yang akan Anda lakukan saat membuat grup keamanan untuk Amazon EFS. Untuk petunjuk cara membuat grup keamanan, lihat [Membuat grup keamanan](#) di Panduan Pengguna Amazon VPC.

1. Untuk instans EC2 Anda, buat grup keamanan dengan aturan berikut:
 - Aturan masuk yang memungkinkan akses masuk menggunakan Secure Shell (SSH) pada port 22 dari alamat IP atau jaringan Anda. Secara opsional, batasi alamat Sumber untuk keamanan.
 - Aturan keluar yang memungkinkan akses keluar pada port NFS 2049 ke grup keamanan target mount. Identifikasi grup keamanan target mount sebagai tujuan.
2. Untuk target pemasangan EFS Anda, buat grup keamanan dengan aturan berikut:
 - Aturan masuk yang memungkinkan akses pada port NFS 2049 dari grup keamanan EC2. Identifikasi grup keamanan EC2 sebagai sumbernya.

Note

Anda tidak perlu menambahkan aturan keluar karena aturan keluar default memungkinkan semua lalu lintas keluar.

Bekerja dengan titik akhir VPC antarmuka di Amazon EFS

Untuk membuat koneksi pribadi antara virtual private cloud (VPC) dan Amazon EFS API, Anda dapat membuat antarmuka VPC endpoint. Endpoint menyediakan konektivitas aman ke Amazon EFS API tanpa memerlukan gateway internet, instans NAT, atau koneksi jaringan pribadi virtual (VPN). Untuk informasi selengkapnya, lihat [Mengakses AWS layanan menggunakan titik akhir VPC antarmuka di Panduan Pengguna Amazon VPC](#).

Endpoint VPC antarmuka didukung oleh AWS PrivateLink, fitur yang memungkinkan komunikasi pribadi antar AWS layanan menggunakan alamat IP pribadi. Untuk menggunakannya AWS PrivateLink, buat titik akhir VPC antarmuka untuk Amazon EFS di VPC Anda menggunakan konsol Amazon VPC, API, atau CLI. Melakukan hal ini akan membuat elastic network interface di subnet Anda dengan alamat IP pribadi yang melayani permintaan Amazon EFS API. Anda juga dapat mengakses titik akhir VPC dari lingkungan lokal atau dari VPCs penggunaan lain Site-to-Site VPN,, Direct Connect atau peering VPC. Untuk mempelajari selengkapnya, lihat [Menyambungkan VPC ke layanan yang digunakan AWS PrivateLink](#) di Panduan Pengguna Amazon VPC.

Membuat titik akhir antarmuka untuk Amazon EFS

Untuk membuat titik akhir VPC antarmuka untuk Amazon EFS, gunakan salah satu dari berikut ini:

- **com.amazonaws.*region*.elasticfilesystem**— Membuat titik akhir untuk operasi Amazon EFS API.
- **com.amazonaws.*region*.elasticfilesystem-fips**— Membuat titik akhir untuk Amazon EFS API yang sesuai dengan [Federal Information Processing Standard \(FIPS\)](#) 140-2.

Untuk daftar lengkap titik akhir Amazon EFS, lihat titik akhir [dan kuota Amazon Elastic File System](#) di Referensi Umum Amazon Web Services

Untuk informasi selengkapnya tentang cara membuat titik akhir antarmuka, lihat [Mengakses AWS layanan menggunakan titik akhir VPC antarmuka](#) di Panduan Pengguna Amazon VPC.

Membuat kebijakan titik akhir VPC untuk Amazon EFS

Untuk mengontrol akses ke Amazon EFS API, Anda dapat melampirkan kebijakan AWS Identity and Access Management (IAM) ke titik akhir VPC Anda. Kebijakan menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.

- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir di Panduan Pengguna](#) Amazon VPC.

Contoh berikut menunjukkan kebijakan titik akhir VPC yang menolak izin semua orang untuk membuat sistem file EFS melalui titik akhir. Kebijakan contoh juga memberikan izin kepada semua orang untuk melakukan semua tindakan lainnya.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "elasticfilesystem:CreateFileSystem",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

Network File System (NFS) tingkat pengguna, grup, dan izin

Setelah membuat sistem file, secara default hanya pengguna root (UID 0) yang telah membaca, menulis, dan mengeksekusi izin. Pengguna lain yang ingin memodifikasi sistem file harus secara eksplisit mendapatkan akses dari pengguna root. Anda dapat menggunakan titik akses untuk mengotomatiskan pembuatan direktori yang dapat ditulis oleh pengguna nonroot. Untuk informasi selengkapnya, lihat [Bekerja dengan titik akses](#).

Objek sistem file EFS memiliki mode gaya Unix yang terkait dengannya. Nilai mode ini mendefinisikan izin untuk melakukan tindakan pada objek tersebut. Pengguna yang akrab dengan sistem bergaya Unix dapat dengan mudah memahami bagaimana Amazon EFS berperilaku sehubungan dengan izin ini.

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 sistem file (yaitu, file, direktori, dan sebagainya) dimiliki oleh satu pemilik dan satu grup. Amazon EFS menggunakan numerik yang IDs dipetakan untuk memeriksa izin saat pengguna mencoba mengakses objek sistem file.

Note

Protokol NFS mendukung maksimum 16 grup IDs (GIDs) per pengguna dan tambahan apapun GIDs dipotong dari permintaan klien NFS. Untuk informasi selengkapnya, lihat [Akses ditolak ke file yang diizinkan pada sistem file NFS](#).

Berikut ini, Anda dapat menemukan contoh izin dan diskusi tentang pertimbangan izin NFS untuk Amazon EFS.

Topik

- [Izin file dan direktori](#)
- [Contoh kasus penggunaan dan izin sistem file EFS](#)
- [Izin ID pengguna dan grup untuk file dan direktori dalam sistem file](#)
- [Tidak ada akar terjepit](#)
- [Caching izin](#)
- [Mengubah kepemilikan objek sistem file](#)
- [Titik akses EFS](#)

Izin file dan direktori

File dan direktori dalam sistem file EFS mendukung izin baca, tulis, dan eksekusi standar UNIX berdasarkan ID pengguna dan grup yang ditegaskan oleh klien pemasangan NFSv4 .1, kecuali diganti oleh titik akses EFS. Untuk informasi selengkapnya, lihat [Network File System \(NFS\) tingkat pengguna, grup, dan izin](#).

Note

Secara default, lapisan kontrol akses ini bergantung pada kepercayaan klien NFSv4 .1 dalam pernyataan pengguna dan ID grup. Anda dapat menggunakan kebijakan dan

kebijakan identitas berbasis sumber daya AWS Identity and Access Management (IAM) untuk mengotorisasi klien NFS dan memberikan izin akses read-only, write, dan root. Anda dapat menggunakan titik akses EFS untuk mengganti informasi identitas pengguna dan grup sistem operasi yang disediakan oleh klien NFS. Untuk informasi selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#) dan [Membuat titik akses](#).

Sebagai contoh membaca, menulis, dan mengeksekusi izin untuk file dan direktori, Alice mungkin memiliki izin untuk membaca dan menulis ke file apa pun yang dia inginkan dalam direktori pribadinya pada sistem file. `/alice` Namun, dalam contoh ini Alice tidak diperbolehkan membaca atau menulis ke file apa pun di direktori pribadi Mark pada sistem file yang sama. `/mark` Baik Alice dan Mark diizinkan untuk membaca tetapi tidak menulis file di direktori `/share` bersama.

Contoh kasus penggunaan dan izin sistem file EFS

Setelah membuat sistem file EFS dan memasang target untuk sistem file di VPC, Anda dapat memasang sistem file jarak jauh secara lokal di instans Amazon EC2 Anda. `mount` Perintah dapat me-mount direktori apa pun di sistem file. Namun, ketika Anda pertama kali membuat sistem file, hanya ada satu direktori root di `/`. Pengguna root dan grup root memiliki direktori yang dipasang.

`mount` Perintah berikut memasang direktori root dari sistem file Amazon EFS, yang diidentifikasi oleh nama DNS sistem file, pada direktori `/efs-mount-point` lokal.

```
sudo mount -t nfs -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport file-
system-id.efs.aws-region.amazonaws.com:/ efs-mount-point
```

Mode izin awal memungkinkan:

- `read-write-execute` izin ke root pemilik
- `read-execute` izin ke root grup
- `read-execute` izin untuk orang lain

Hanya pengguna root yang dapat memodifikasi direktori ini. Pengguna root juga dapat memberikan izin kepada pengguna lain untuk menulis ke direktori ini, misalnya:

- Buat subdirektori per pengguna yang dapat ditulis. Untuk step-by-step instruksi, lihat [Tutorial: Membuat subdirektori per pengguna yang dapat ditulis](#).

- Izinkan pengguna untuk menulis ke root sistem file EFS. Seorang pengguna dengan hak akses root dapat memberikan pengguna lain akses ke sistem file.
- Untuk mengubah kepemilikan sistem file EFS menjadi pengguna dan grup non-root, gunakan yang berikut ini:

```
$ sudo chown user:group /EFSroot
```

- Untuk mengubah izin sistem file menjadi sesuatu yang lebih permisif, gunakan yang berikut ini:

```
$ sudo chmod 777 /EFSroot
```

Perintah ini memberikan read-write-execute hak istimewa kepada semua pengguna pada semua instans EC2 yang memiliki sistem file terpasang.

Izin ID pengguna dan grup untuk file dan direktori dalam sistem file

File dan direktori dalam sistem file EFS mendukung izin baca, tulis, dan eksekusi standar UNIX berdasarkan ID pengguna dan grup. IDs Ketika klien NFS memasang sistem file EFS tanpa menggunakan titik akses, ID pengguna dan ID grup yang disediakan oleh klien dipercaya. Anda dapat menggunakan titik akses EFS untuk mengganti ID pengguna dan grup yang IDs digunakan oleh klien NFS. Saat pengguna mencoba mengakses file dan direktori, Amazon EFS memeriksa pengguna IDs dan grup mereka IDs untuk memverifikasi bahwa setiap pengguna memiliki izin untuk mengakses objek. Amazon EFS juga menggunakan ini IDs untuk menunjukkan pemilik dan pemilik grup untuk file dan direktori baru yang dibuat pengguna. Amazon EFS tidak memeriksa nama pengguna atau grup — Amazon EFS hanya menggunakan pengidentifikasi numerik.

Note

Saat membuat pengguna pada instans EC2, Anda dapat menetapkan ID pengguna numerik (UID) dan ID grup (GID) apa pun kepada pengguna. Pengguna numerik IDs diatur dalam `/etc/passwd` file pada sistem Linux. Grup numerik IDs ada di `/etc/group` file. File-file ini menentukan pemetaan antara nama dan ID. Di luar instans EC2, Amazon EFS tidak melakukan autentikasi apa pun IDs, termasuk ID root 0.

Jika pengguna mengakses sistem file EFS dari dua instans EC2 yang berbeda, tergantung pada apakah UID untuk pengguna sama atau berbeda pada instance tersebut, Anda melihat perilaku yang berbeda, sebagai berikut:

- Jika ID pengguna sama pada kedua instans EC2, Amazon EFS menganggapnya menunjukkan pengguna yang sama, terlepas dari instans EC2 yang digunakan. Pengalaman pengguna saat mengakses sistem file sama dari kedua instans EC2.
- Jika ID pengguna tidak sama pada kedua instans EC2, Amazon EFS menganggap pengguna tersebut sebagai pengguna yang berbeda. Pengalaman pengguna tidak sama saat mengakses sistem file EFS dari dua instans EC2 yang berbeda.
- Jika dua pengguna berbeda pada instans EC2 yang berbeda berbagi ID, Amazon EFS menganggap mereka sebagai pengguna yang sama.

Anda dapat mempertimbangkan untuk mengelola pemetaan ID pengguna di seluruh instans EC2 secara konsisten. Pengguna dapat memeriksa ID numerik mereka menggunakan `id` perintah.

```
$ id

uid=502(joe) gid=502(joe) groups=502(joe)
```

Matikan ID Mapper

Utilitas NFS dalam sistem operasi termasuk daemon yang disebut ID Mapper yang mengelola pemetaan antara nama pengguna dan IDs. Di Amazon Linux, daemon dipanggil `rpc.idmapd` dan di Ubuntu disebut `idmapd`. Ini menerjemahkan ID pengguna dan grup ke dalam nama, dan sebaliknya. Namun, Amazon EFS hanya berurusan dengan ID numerik. Kami menyarankan Anda mematikan proses ini pada instans EC2 Anda. Di Amazon Linux, ID mapper biasanya dinonaktifkan, dan jika tidak mengaktifkannya. Untuk mematikan ID mapper, gunakan perintah yang ditunjukkan berikut.

```
$ service rpcidmapd status
$ sudo service rpcidmapd stop
```

Tidak ada akar terjepit

Secara default, root squashing dinonaktifkan pada sistem file EFS. Amazon EFS berperilaku seperti server Linux NFS dengan `no_root_squash`. Jika ID pengguna atau grup adalah 0, Amazon EFS memperlakukan pengguna tersebut sebagai root pengguna, dan melewati pemeriksaan izin

(memungkinkan akses dan modifikasi ke semua objek sistem file). Root squashing dapat diaktifkan pada koneksi klien ketika identitas AWS Identity and Access Management (AWS IAM) atau kebijakan sumber daya tidak mengizinkan akses ke tindakan. `ClientRootAccess` Ketika root squashing diaktifkan, pengguna root dikonversi ke pengguna dengan izin terbatas pada server NFS.

Untuk informasi selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses ke sistem file](#).

Aktifkan root squashing menggunakan otorisasi IAM untuk klien NFS

Anda dapat mengonfigurasi Amazon EFS untuk mencegah akses root ke sistem file EFS Anda untuk semua AWS prinsipal kecuali untuk satu workstation manajemen. Anda melakukan ini dengan mengonfigurasi otorisasi AWS Identity and Access Management (IAM) untuk klien Sistem File Jaringan (NFS).

Untuk melakukan ini memerlukan konfigurasi dua kebijakan izin IAM, sebagai berikut:

- Buat kebijakan sistem file EFS yang secara eksplisit memungkinkan akses baca dan tulis ke sistem file, dan secara implisit menolak akses root.
- Tetapkan identitas IAM ke workstation manajemen Amazon EC2 yang memerlukan akses root ke sistem file dengan menggunakan profil instans EC2. Untuk informasi selengkapnya tentang profil instans Amazon EC2, lihat [Menggunakan profil instans](#) di AWS Identity and Access Management Panduan Pengguna.
- Tetapkan kebijakan yang `AmazonElasticFileSystemClientFullAccess` AWS dikelola ke peran IAM dari workstation manajemen. Untuk informasi selengkapnya tentang kebijakan AWS dikelola untuk Amazon EFS, lihat [Manajemen identitas dan akses untuk Amazon EFS](#).

Untuk mengaktifkan root squashing menggunakan otorisasi IAM untuk klien NFS, gunakan prosedur berikut.

Untuk mencegah akses root ke sistem file

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Pilih Sistem file.
3. Pilih sistem file yang ingin Anda aktifkan root squashing.
4. Pada halaman detail sistem file, pilih Kebijakan sistem berkas, lalu pilih Edit. Halaman Kebijakan sistem file muncul.
5. Pilih Cegah akses root secara default* di bawah Opsi kebijakan. Objek kebijakan JSON muncul di editor Kebijakan.

6. Pilih Simpan untuk menyimpan kebijakan sistem file.

Klien yang tidak anonim bisa mendapatkan akses root ke sistem file melalui kebijakan berbasis identitas. Saat Anda melampirkan kebijakan `AmazonElasticFileSystemClientFullAccess` terkelola ke peran stasiun kerja, IAM memberikan akses root ke workstation berdasarkan kebijakan identitasnya.

Untuk mengaktifkan akses root dari workstation manajemen

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Buat peran untuk Amazon EC2 yang disebut. `EFS-client-root-access` IAM membuat profil instance dengan nama yang sama dengan peran EC2 yang Anda buat.
3. Tetapkan kebijakan AWS terkelola `AmazonElasticFileSystemClientFullAccess` ke peran EC2 yang Anda buat. Isi kebijakan ini ditampilkan sebagai berikut.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:DescribeMountTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

4. Lampirkan profil instans ke instans EC2 yang Anda gunakan sebagai workstation manajemen, seperti yang dijelaskan berikut. Untuk informasi selengkapnya, lihat [Melampirkan Peran IAM ke Instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
 - a. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
 - b. Di panel navigasi, pilih Instans.

- c. Pilih instance. Untuk Tindakan, pilih Pengaturan Instans, lalu pilih Lampirkan/Ganti peran IAM.
- d. Pilih peran IAM yang Anda buat pada langkah pertama `EFS-client-root-access`, dan pilih Terapkan.
5. Instal EFS mount helper di workstation manajemen. Untuk informasi lebih lanjut tentang EFS mount helper mount helper helper dan `amazon-efs-utils` paketnya, lihat. [Menginstal klien Amazon EFS](#)
6. Pasang sistem file EFS pada workstation manajemen dengan menggunakan perintah berikut dengan opsi `iam mount`.

```
$ sudo mount -t efs -o tls,iam file-system-id:/ efs-mount-point
```

Anda dapat mengonfigurasi instans Amazon EC2 untuk secara otomatis memasang sistem file dengan otorisasi IAM. Untuk informasi selengkapnya tentang pemasangan sistem file EFS dengan otorisasi IAM, lihat. [Pemasangan dengan otorisasi IAM](#)

Caching izin

Amazon EFS menyimpan izin file dalam cache untuk jangka waktu yang kecil. Akibatnya, mungkin ada jendela singkat di mana pengguna yang aksesnya dicabut baru-baru ini masih dapat mengakses objek itu.

Mengubah kepemilikan objek sistem file

Amazon EFS memberlakukan atribut `POSIXchown_restricted`. Ini berarti hanya pengguna root yang dapat mengubah pemilik objek sistem file. Root atau pengguna pemilik dapat mengubah grup pemilik objek sistem file. Namun, kecuali pengguna root, grup hanya dapat diubah menjadi salah satu yang pengguna pemilik adalah anggota.

Titik akses EFS

Titik akses menerapkan jalur pengguna, grup, dan sistem file sistem operasi ke permintaan sistem file apa pun yang dibuat menggunakan titik akses. Pengguna dan grup sistem operasi jalur akses mengesampingkan informasi identitas apa pun yang disediakan oleh klien NFS. Jalur sistem file diekspos ke klien sebagai direktori root titik akses. Pendekatan ini memastikan bahwa setiap aplikasi selalu menggunakan identitas sistem operasi yang benar dan direktori yang benar saat mengakses

kumpulan data berbasis file bersama. Aplikasi yang menggunakan titik akses hanya bisa mengakses data di direktori sendiri dan di bawah ini. Untuk informasi lebih lanjut tentang titik akses, lihat [Bekerja dengan titik akses](#).

Bekerja dengan titik akses

Titik akses EFS adalah titik masuk khusus aplikasi ke dalam sistem file EFS yang memudahkan pengelolaan akses aplikasi ke kumpulan data bersama. Titik akses dapat menerapkan identitas pengguna, termasuk grup POSIX pengguna, pada semua permintaan sistem file yang dibuat melalui titik akses. Titik akses juga dapat menerapkan direktori asal yang berbeda untuk sistem file sehingga klien hanya dapat mengakses data dalam direktori tertentu atau subdirektornya.

Anda dapat menggunakan kebijakan AWS Identity and Access Management (IAM) untuk menegaskan bahwa aplikasi tertentu menggunakan titik akses tertentu. Dengan menggabungkan kebijakan IAM dengan titik akses, Anda dapat dengan mudah memberikan akses yang aman ke set data tertentu untuk aplikasi Anda.

Anda dapat membuat titik akses untuk sistem file EFS yang ada menggunakan Konsol Manajemen AWS, the AWS Command Line Interface (AWS CLI), dan Amazon EFS API. Untuk step-by-step prosedur untuk membuat titik akses, lihat [Membuat titik akses](#).

Titik akses bekerja dengan target pemasangan

Anda harus membuat setidaknya satu target pemasangan di VPC Anda sebelum menggunakan titik akses. Target pemasangan menyediakan konektivitas jaringan ke sistem file EFS Anda sementara titik akses menyediakan kontrol akses dan titik masuk khusus aplikasi.

Titik akses mewarisi penempatan Availability Zone target mount.

- Grup keamanan diterapkan pada tingkat target pemasangan, bukan tingkat titik akses.
- Titik akses tersedia di semua Availability Zone di mana Anda memiliki target mount.
- Kunci kondisi IAM `elasticfilesystem:AccessedViaMountTarget` memastikan akses sistem file hanya terjadi melalui target pemasangan, yang berlaku untuk pemasangan langsung danudukan titik akses.

Anda menggunakan EFS mount helper saat memasang sistem file menggunakan titik akses. Dalam perintah mount, sertakan ID sistem file, ID titik akses, dan opsi `tls mount`, seperti yang ditunjukkan pada contoh berikut.

```
$ mount -t efs -o tls,iam,accesspoint=fsap-abcdef0123456789a fs-  
abc0123def456789a: /localmountpoint
```

Untuk informasi lebih lanjut tentang pemasangan sistem file menggunakan titik akses, lihat [Pemasangan dengan titik akses EFS](#).

Topik

- [Menegakkan identitas pengguna menggunakan titik akses](#)
- [Menegakkan direktori root dengan titik akses](#)
- [Menggunakan titik akses dalam kebijakan IAM](#)

Menegakkan identitas pengguna menggunakan titik akses

Anda dapat menggunakan titik akses untuk menegakkan informasi pengguna dan grup untuk semua permintaan sistem file yang dibuat melalui titik akses. Untuk mengaktifkan fitur ini, Anda perlu menentukan identitas sistem operasi yang akan diterapkan saat Anda membuat titik akses.

Sebagai bagian dari ini, Anda memberikan yang berikut:

- User ID — ID pengguna POSIX numerik untuk pengguna.
- ID Grup — ID grup POSIX numerik untuk pengguna.
- Kelompok sekunder IDs — Daftar opsional kelompok sekunder IDs.

Saat penegakan pengguna diaktifkan, Amazon EFS menggantikan pengguna dan grup klien NFS IDs dengan identitas yang dikonfigurasi pada titik akses untuk semua operasi sistem file. Penegakan pengguna juga melakukan hal berikut:

- Pemilik dan grup untuk file dan direktori baru diatur ke ID pengguna dan ID grup dari titik akses.
- EFS mempertimbangkan ID pengguna, ID grup, dan grup sekunder IDs dari titik akses saat mengevaluasi izin sistem file. EFS mengabaikan klien NFS. IDs

Important

Menegakkan identitas pengguna tunduk pada izin `ClientRootAccess` IAM.

Misalnya, dalam beberapa kasus Anda mungkin mengonfigurasi ID pengguna titik akses, ID grup, atau keduanya menjadi root (yaitu, menyetel UID, GID, atau keduanya ke 0). Dalam kasus seperti itu, Anda harus memberikan izin `ClientRootAccess` IAM kepada klien NFS.

Menegakkan direktori root dengan titik akses

Anda dapat menggunakan titik akses untuk mengganti direktori root untuk sistem file. Saat Anda menerapkan direktori root, klien NFS yang menggunakan titik akses menggunakan direktori root yang dikonfigurasi pada titik akses alih-alih direktori root sistem file.

Anda mengaktifkan fitur ini dengan mengatur `Path` atribut titik akses saat membuat titik akses. `PathAtribut` adalah jalur lengkap direktori root sistem file untuk semua permintaan sistem file yang dibuat melalui titik akses ini. Path lengkap tidak boleh melebihi 100 karakter panjangnya. Ini dapat mencakup hingga empat subdirektori.

Ketika Anda menentukan direktori root pada titik akses, itu menjadi direktori root dari sistem file untuk klien NFS yang memasang titik akses. Misalnya, misalkan direktori root dari titik akses Anda adalah `/data`. Dalam hal ini, pemasangan `fs-12345678:/` menggunakan titik akses memiliki efek yang sama seperti pemasangan `fs-12345678:/data` tanpa menggunakan titik akses.

Saat menentukan direktori root di titik akses Anda, pastikan bahwa izin direktori dikonfigurasi untuk memungkinkan pengguna titik akses berhasil memasang sistem file. Secara khusus, pastikan bahwa bit eksekusi diatur untuk pengguna atau grup titik akses, atau untuk semua orang. Misalnya, nilai izin direktori `755` memungkinkan pemilik pengguna direktori untuk mencantumkan file, membuat file, dan memasang, dan semua pengguna lain untuk membuat daftar file dan memasang.

Membuat direktori root untuk titik akses

Jika jalur direktori root untuk titik akses tidak ada di sistem file, Amazon EFS secara otomatis membuat direktori root tersebut dengan kepemilikan dan izin yang ditentukan. Amazon EFS tidak akan membuat direktori root jika Anda tidak menentukan kepemilikan direktori dan izin saat pembuatan. Pendekatan ini memungkinkan untuk menyediakan akses sistem file untuk pengguna atau aplikasi tertentu tanpa memasang sistem file Anda dari host Linux. Untuk membuat direktori root, Anda harus mengonfigurasi kepemilikan dan izin direktori root dengan menggunakan atribut berikut saat membuat titik akses:

- `OwnerUid`— ID pengguna POSIX numerik untuk digunakan sebagai pemilik direktori root.

- **OwnerGid**— ID grup POSIX numerik untuk digunakan sebagai grup pemilik direktori root.
- **Izin** — Mode Unix direktori. Konfigurasi umum adalah 755. Pastikan bit eksekusi diatur untuk pengguna titik akses sehingga mereka dapat me-mount. Konfigurasi ini memberikan izin pemilik direktori untuk memasukkan, membuat daftar, dan menulis file baru di direktori. Ini memberikan semua pengguna lain izin untuk memasukkan dan daftar file. Untuk informasi lebih lanjut tentang bekerja dengan file Unix dan mode direktori, lihat [Network File System \(NFS\) tingkat pengguna, grup, dan izin](#).

Amazon EFS membuat direktori root titik akses hanya jika OwnUid, ownGid, dan izin ditentukan untuk direktori. Jika Anda tidak memberikan informasi ini, Amazon EFS tidak membuat direktori root. Jika direktori root tidak ada, upaya untuk memasang menggunakan titik akses akan gagal.

Saat Anda memasang sistem file dengan titik akses, direktori root untuk titik akses dibuat jika direktori belum ada, asalkan direktori root OwnerUid dan Izin ditentukan saat titik akses dibuat. Jika direktori root titik akses sudah ada sebelum waktu pemasangan, izin yang ada tidak akan ditimpa oleh titik akses. Jika Anda menghapus direktori root, EFS membuatnya kembali saat berikutnya sistem file dipasang menggunakan titik akses.

Note

Jika Anda tidak menentukan kepemilikan dan izin untuk direktori root titik akses, Amazon EFS tidak akan membuat direktori root. Semua upaya untuk me-mount titik akses akan gagal.

Model keamanan untuk direktori root titik akses

Ketika penggantian direktori root berlaku, Amazon EFS berperilaku seperti server NFS Linux dengan opsi diaktifkan. `no_subtree_check`

Dalam protokol NFS, server menghasilkan pegangan file yang digunakan oleh klien sebagai referensi unik saat mengakses file. EFS secara aman menghasilkan pegangan file yang tidak dapat diprediksi dan spesifik untuk sistem file EFS. Saat penggantian direktori root dilakukan, Amazon EFS tidak mengungkapkan penanganan file untuk file di luar direktori root yang ditentukan. Namun, dalam beberapa kasus pengguna mungkin mendapatkan pegangan file untuk file di luar jalur akses mereka dengan menggunakan out-of-band mekanisme. Misalnya, mereka mungkin melakukannya jika mereka memiliki akses ke titik akses kedua. Jika mereka melakukan ini, mereka dapat melakukan operasi baca dan tulis pada file.

Kepemilikan file dan izin akses selalu diberlakukan, untuk akses ke file di dalam dan di luar direktori root titik akses pengguna.

Menggunakan titik akses dalam kebijakan IAM

Anda dapat menggunakan kebijakan IAM untuk menegaskan bahwa klien NFS tertentu, yang diidentifikasi oleh peran IAM-nya, hanya dapat mengakses titik akses tertentu. Untuk melakukan ini, Anda menggunakan kunci kondisi `elasticfilesystem:AccessPointArn` IAM. `AccessPointArn` itu adalah Nama Sumber Daya Amazon (ARN) dari titik akses tempat sistem file dipasang.

Berikut ini adalah contoh kebijakan sistem file yang memungkinkan peran IAM `app1` untuk mengakses sistem file menggunakan titik `fsap-01234567` akses. Kebijakan ini juga memungkinkan `app2` untuk menggunakan sistem file menggunakan titik akses `fsap-89abcdef`.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "MyFileSystemPolicy",
  "Statement": [
    {
      "Sid": "App1Access",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/app1" },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/*",
      "Condition": {
        "StringEquals": {
          "elasticfilesystem:AccessPointArn" :
            "arn:aws:elasticfilesystem:us-east-1:222233334444:access-point/fsap-01234567"
        }
      }
    },
    {
      "Sid": "App2Access",
      "Effect": "Allow",
```

```

    "Principal": { "AWS": "arn:aws:iam::111122223333:role/app2" },
    "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
    "Condition": {
        "StringEquals": {
            "elasticfilesystem:AccessPointArn" :
            "arn:aws:elasticfilesystem:us-east-1:222233334444:access-point/fsap-89abcdef"
        }
    }
}

```

Memblokir akses publik ke sistem file EFS

Fitur Amazon EFS memblokir akses publik menyediakan pengaturan untuk membantu Anda mengelola akses publik ke sistem file EFS. Secara default, sistem file EFS baru tidak mengizinkan akses publik. Namun, Anda dapat mengubah kebijakan sistem file untuk memungkinkan akses publik.

Important

Mengaktifkan akses Blokir Akses Publik membantu melindungi sumber daya Anda dengan mencegah akses publik diberikan melalui kebijakan sumber daya yang langsung dilampirkan ke sistem file. Selain mengaktifkan Blokir Akses Publik, periksa kebijakan berikut dengan cermat untuk mengonfirmasi bahwa kebijakan tersebut tidak memberikan akses publik:

- Kebijakan berbasis identitas yang dilampirkan pada AWS prinsipal terkait (misalnya, peran IAM)
- Kebijakan berbasis sumber daya yang dilampirkan pada AWS sumber daya terkait (misalnya, kunci (KMS AWS Key Management Service))

Topik

- [Memblokir akses publik dengan AWS Transfer Family](#)
- [Arti “publik”](#)

Memblokir akses publik dengan AWS Transfer Family

Saat Anda menggunakan Amazon EFS dengan AWS Transfer Family, permintaan akses sistem file yang diterima dari server Transfer Family yang dimiliki oleh akun berbeda dari sistem file akan diblokir jika sistem file mengizinkan akses publik. Amazon EFS mengevaluasi kebijakan IAM sistem file, dan jika kebijakan tersebut bersifat publik, kebijakan tersebut memblokir permintaan. Untuk mengizinkan AWS Transfer Family akses ke sistem file Anda, perbarui kebijakan sistem file Anda sehingga tidak dianggap publik.

Note

Menggunakan Transfer Family dengan Amazon EFS dinonaktifkan secara default untuk Akun AWS yang memiliki sistem file EFS dengan kebijakan yang memungkinkan akses publik yang dibuat sebelum 6 Januari 2021. Untuk mengaktifkan penggunaan Transfer Family untuk mengakses sistem file Anda, hubungi AWS Support.

Arti “publik”

Saat mengevaluasi apakah sistem file memungkinkan akses publik, Amazon EFS mengasumsikan bahwa kebijakan sistem file bersifat publik. Kemudian mengevaluasi kebijakan sistem file untuk menentukan apakah itu memenuhi syarat sebagai non-publik. Agar dianggap non-publik, kebijakan sistem file harus memberikan akses hanya ke nilai tetap (nilai yang tidak mengandung kartu liar) dari satu atau beberapa hal berikut:

- AWS Prinsipal, pengguna, peran, atau prinsipal layanan (misalnya, `aws:PrincipalOrgID`)
- `aws:SourceArn`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `elasticfilesystem:AccessedViaMountTarget`
- `aws:userid`, outside the pattern `"AROLEID:*`

Berdasarkan aturan ini, contoh kebijakan berikut dianggap publik.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
  "Statement": [
    {
      "Sid": "efs-statement-14a7191c-9401-40e7-a388-6af6cfb7dd9c",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientRootAccess"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/*"
    }
  ]
}
```

Anda dapat membuat kebijakan sistem file ini non-publik dengan menggunakan kunci kondisi EFS yang `elasticfilesystem:AccessedViaMountTarget` disetel ke `true`. Anda dapat menggunakan `elasticfilesystem:AccessedViaMountTarget` untuk mengizinkan tindakan EFS yang ditentukan kepada klien yang mengakses sistem file EFS menggunakan target pemasangan sistem file. Kebijakan non-publik berikut menggunakan kunci `elasticfilesystem:AccessedViaMountTarget` kondisi yang disetel ke `true`.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
  "Statement": [
    {
      "Sid": "efs-statement-14a7191c-9401-40e7-a388-6af6cfb7dd9c",
      "Effect": "Allow",
      "Principal": {
```

```
        "AWS": "*"
      },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientRootAccess"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    }
  ]
}
```

Untuk informasi selengkapnya tentang kunci kondisi Amazon EFS, lihat [Kunci kondisi EFS untuk klien](#). Untuk informasi selengkapnya tentang membuat kebijakan sistem file, lihat [Membuat kebijakan sistem file](#).

Isolasi jaringan untuk Amazon EFS

Sebagai layanan terkelola, Amazon Elastic File System 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 Amazon EFS 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.

Ini dapat APIs dipanggil dari lokasi jaringan mana pun, tetapi Amazon EFS mendukung kebijakan akses berbasis sumber daya yang dapat mencakup pembatasan berdasarkan alamat IP sumber. Anda juga dapat menggunakan kebijakan Amazon EFS untuk mengontrol akses dari titik akhir Amazon Virtual Private Cloud (Amazon VPC) tertentu, atau spesifik. VPCs Secara efektif, ini mengisolasi akses jaringan ke sumber daya Amazon EFS tertentu hanya dari VPC tertentu dalam AWS jaringan.

Kuota Amazon EFS

Berikut ini, Anda dapat mengetahui tentang kuota saat bekerja dengan Amazon EFS.

Topik

- [Kuota Amazon EFS yang dapat Anda tingkatkan](#)
- [Kuota sumber daya Amazon EFS yang tidak dapat Anda ubah](#)
- [Kuota untuk klien NFS](#)
- [Kuota untuk sistem file Amazon EFS](#)
- [Fitur NFSv4 0 dan 4.1 yang tidak didukung](#)
- [Pertimbangan tambahan](#)
- [Memecahkan masalah kesalahan operasi file yang terkait dengan kuota](#)

Kuota Amazon EFS yang dapat Anda tingkatkan

Service Quotas adalah AWS layanan yang membantu Anda mengelola kuota, atau batasan, dari satu lokasi. Di [konsol Service Quotas](#), Anda dapat melihat nilai batas Amazon EFS dan meminta peningkatan kuota untuk jumlah sistem file EFS di IOPS AWS Region dan yang telah dibaca untuk data yang sering diakses.

Anda juga dapat meminta peningkatan untuk kuota Amazon EFS berikut dengan menghubungi AWS Support. Untuk mempelajari selengkapnya, lihat [Meminta peningkatan kuota](#). Tim layanan Amazon EFS meninjau setiap permintaan secara individual.

- Jumlah sistem file untuk setiap akun pelanggan.
- Jumlah titik akses untuk setiap sistem file.
- IOPS baca maksimum per sistem file menggunakan throughput Elastis. Ketika IOPS baca untuk sistem file yang sering diakses ditingkatkan, baik IOPS baca untuk sistem file yang jarang diakses dan IOPS tulis juga meningkat.
- Throughput elastis per sistem file Regional untuk semua klien yang terhubung dalam file AWS Region.
- Throughput yang disediakan per sistem file Regional untuk semua klien yang terhubung dalam file. AWS Region

Tabel berikut mencantumkan kuota sumber daya default umum yang dapat Anda minta untuk diubah.

Sumber daya	Kuota bawaan
Jumlah sistem file untuk setiap akun pelanggan dalam AWS Region	1.000
Jumlah titik akses untuk setiap sistem file	10.000
IOPS maksimum per sistem file menggunakan throughput Elastis	Data yang jarang diakses dibaca: 90.000
	Data yang sering diakses dibaca: 250.000
	Menulis: 50.000
Anda dapat meminta peningkatan hingga 10 kali kuota default IOPS baca untuk data yang sering diakses. Peningkatan IOPS baca untuk data yang sering diakses juga menghasilkan peningkatan IOPS baca untuk data yang jarang diakses dan IOPS tulis.	

Tabel berikut mencantumkan kuota throughput elastis per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

Sistem file regional — Total throughput Elastis default per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

AWS Region	Throughput baca maksimum	Throughput penulisan maksimum (throughput terukur)
Wilayah AS Timur (Ohio)	60 gibibyte per detik () GiBps	5 GiBps
Wilayah AS Timur (Virginia Utara)		

AWS Region	Throughput baca maksimum	Throughput penulisan maksimum (throughput terukur)
Wilayah AS Barat (Oregon)		
Wilayah Asia Pacific (Mumbai)		
Wilayah Asia Pasifik (Seoul)		
Wilayah Asia Pasifik (Singapura)		
Wilayah Asia Pasifik (Sydney)		
Wilayah Asia Pacific (Tokyo)		
Wilayah Europe (Frankfurt)		
Wilayah Eropa (Irlandia)		
Wilayah Eropa (London)		
Semua lainnya Wilayah AWS	20 GiBps	1 GiBps

Tabel berikut mencantumkan kuota throughput yang disediakan per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

Sistem file regional - Total default throughput yang disediakan per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

AWS Region	Throughput baca maksimum	Throughput penulisan maksimum (throughput terukur)
Wilayah AS Timur (Ohio)	10 GiBps	3.33 GiBps
Wilayah AS Timur (Virginia Utara)		
Wilayah US West (Oregon)		

AWS Region	Throughput baca maksimum	Throughput penulisan maksimum (throughput terukur)
Wilayah Europe (Irlandia)		
Semua lainnya Wilayah AWS	3 GiBps	1 GiBps

Meminta peningkatan kuota

Untuk meminta peningkatan kuota ini AWS Dukungan, lakukan langkah-langkah berikut. Tim Amazon EFS meninjau setiap permintaan peningkatan kuota.

Meminta kenaikan kuota melalui AWS Dukungan

1. Buka halaman [AWS Dukungan Tengah](#), dan masuk jika perlu. Kemudian pilih Create Case.
2. Di bawah Buat kasus, pilih Peningkatan Batas Layanan.
3. Untuk Tipe Batas, pilih jenis batas yang akan ditingkatkan. Isi kolom yang diperlukan dalam formulir, lalu pilih metode kontak pilihan Anda.

Kuota sumber daya Amazon EFS yang tidak dapat Anda ubah

Kuota untuk beberapa sumber daya Amazon EFS tidak dapat diubah, termasuk:

- Kuota untuk sumber daya umum, seperti jumlah koneksi untuk setiap sistem file.
- Kuota throughput Elastis dan Provisioned per sistem file One Zone untuk semua klien yang terhubung dalam file. AWS Region
- Kuota throughput yang meledak per sistem file Regional atau Satu Zona untuk semua klien yang terhubung dalam file. AWS Region

Tabel berikut mencantumkan kuota sumber daya umum yang tidak dapat diubah.

Kuota sumber daya umum yang tidak dapat diubah

Sumber daya	Kuota
Jumlah koneksi untuk setiap sistem file	25.000
Jumlah target pemasangan untuk setiap sistem file di Availability Zone	1
Jumlah target pemasangan untuk setiap virtual private cloud (VPC)	1.400
Jumlah grup keamanan untuk setiap target pemasangan	5
Jumlah tag untuk setiap sistem file	50
Jumlah VPCs untuk setiap sistem file	1

Note

Klien juga dapat terhubung ke target mount yang ada di akun atau VPC yang berbeda dari sistem file. Untuk informasi selengkapnya, lihat [Memasang sistem file EFS dari yang lain Akun AWS atau VPC](#).

Tabel berikut mencantumkan total batas throughput Elastis dan Provisioned default per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

Sistem file One Zone — Total default Elastic dan Provisioned throughput per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

AWS Region	Throughput baca maksimum	Throughput penulisan maksimum (throughput terukur)
Semua Wilayah AWS	3 GiBps	1 GiBps

Tabel berikut mencantumkan total batas throughput Bursting per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

Sistem file Regional dan Satu Zona — Total throughput Bursting per sistem file untuk semua klien yang terhubung di masing-masing AWS Region

AWS Region	Throughput baca maksimum	Throughput penulisan maksimum
Wilayah AS Timur (Ohio)	5 GiBps	3 GiBps
Wilayah AS Timur (Virginia Utara)		
Wilayah AS Barat (Oregon)		
Wilayah Asia Pasifik (Sydney)		
Wilayah Eropa (Irlandia)		
Semua lainnya Wilayah AWS	3 GiBps	1 GiBps

Kuota untuk klien NFS

Kuota berikut untuk klien NFS berlaku, dengan asumsi klien Linux NFSv4 .1:

- Throughput baca dan tulis gabungan maksimum adalah 1.500 mebibytes per detik (MiBps) untuk sistem file yang menggunakan throughput Elastic dan dipasang menggunakan versi 2.0 atau yang lebih baru dari klien Amazon EFS (versi) amazon-efs-utils atau Amazon EFS CSI Driver (). aws-efs-csi-driver Throughput maksimum untuk semua sistem file lainnya adalah 500 MiBps. Untuk informasi selengkapnya tentang kinerja, lihat [Ringkasan kinerja](#). NFS client throughput dihitung sebagai jumlah total byte yang dikirim dan diterima, dengan ukuran permintaan NFS minimum 4 KB (setelah menerapkan tingkat pengukuran 1/3 untuk permintaan baca).
- Hingga 65.536 pengguna aktif untuk setiap klien dapat membuka file secara bersamaan.
- Hingga 65.536 file terbuka pada saat yang sama pada instance. Daftar isi direktori tidak dihitung sebagai membuka file.
- Setiap pemasangan unik pada klien dapat memperoleh hingga total 65.536 kunci per koneksi.
- Saat menyambung ke Amazon EFS, klien NFS yang berlokasi di lokasi atau di tempat lain AWS Region dapat mengamati throughput yang lebih rendah daripada saat menghubungkan ke EFS dari yang sama. AWS Region Efek ini karena peningkatan latensi jaringan. Latensi jaringan 1 ms

atau kurang diperlukan untuk mencapai throughput maksimum per klien. Gunakan layanan migrasi DataSync data saat memigrasikan kumpulan data besar dari server NFS lokal ke EFS.

- Protokol NFS mendukung maksimum 16 grup IDs (GIDs) per pengguna dan tambahan apa pun GIDs dipotong dari permintaan klien NFS. Untuk informasi selengkapnya, lihat [Akses ditolak ke file yang diizinkan pada sistem file NFS](#).
- Menggunakan Amazon EFS dengan Microsoft Windows tidak didukung.

Kuota untuk sistem file Amazon EFS

Kuota berikut khusus untuk sistem file Amazon EFS.

Sumber daya	Kuota
Panjang nama file, dalam byte	255
Panjang tautan simbolik (symlink), dalam byte	4,080
Jumlah hard link ke file	177
Ukuran satu file	52.673.613.135.872 byte (47,9 TiB)
Jumlah level untuk kedalaman direktori	1.000
Jumlah kunci pada satu file di semua instance dan pengguna	512
Batas karakter untuk setiap kebijakan sistem file	20.000
* Jumlah operasi file per detik untuk mode Tujuan Umum	250.000

*Untuk informasi selengkapnya tentang jumlah operasi file per detik untuk mode Tujuan Umum, lihat [Ringkasan kinerja](#).

Fitur NFSv4 0 dan 4.1 yang tidak didukung

Meskipun Amazon EFS tidak mendukung NFSv2, atau NFSv3, Amazon EFS mendukung NFSv4 .1 dan NFSv4 .0, kecuali untuk fitur berikut:

- pNFS
- Atribut yang diekspos
- Delegasi klien atau callback dari jenis apa pun
 - Operasi OPEN selalu kembali OPEN_DELEGATE_NONE sebagai tipe delegasi.
 - Operasi OPEN mengembalikan NFSERR_NOTSUPP untuk jenis CLAIM_DELEGATE_CUR dan CLAIM_DELEGATE_PREV klaim.
- Penguncian wajib

Semua kunci di Amazon EFS adalah penasihat, yang berarti bahwa operasi baca dan tulis tidak memeriksa kunci yang bertentangan sebelum operasi dijalankan.

- Tolak berbagi

NFS mendukung konsep penolakan saham. Penolakan berbagi terutama digunakan oleh klien Windows bagi pengguna untuk menolak akses orang lain ke file tertentu yang telah dibuka. Amazon EFS tidak mendukung ini, dan mengembalikan kesalahan NFS NFS4ERR_NOTSUPP untuk perintah OPEN apa pun yang menentukan nilai penolakan berbagi selain OPEN4_SHARE_DENY_NONE. Klien Linux NFS tidak menggunakan apa pun selain OPEN4_SHARE_DENY_NONE.

- Daftar kontrol akses (ACLs)
- Amazon EFS tidak memperbarui time_access atribut pada pembacaan file. Amazon EFS memperbarui time_access dalam acara berikut:
 - Saat file dibuat (inode dibuat)
 - Ketika klien NFS membuat panggilan eksplisit setattr
 - Pada penulisan ke inode yang disebabkan oleh, misalnya, perubahan ukuran file atau perubahan metadata file
 - Atribut inode apa pun diperbarui
- Namespace
- Cache balasan persisten
- Keamanan berbasis Kerberos
- NFSv4.1 retensi data
- Setuid pada direktori

- Jenis file yang tidak didukung saat menggunakan operasi CREATE: Blokir perangkat (NF4BLK), perangkat karakter (NF4CHR), direktori atribut (NF4ATTRDIR), dan atribut bernama (NAMEDATTR). NF4
- Atribut yang tidak didukung: FATTR4 _ARCHIVE, FATTR4 _FILES_AVAIL, FATTR4 _FILES_FREE, FATTR4 _FILES_TOTAL, _FS_LOCATIONS, _MIMETYPE, _QUOTA_AVAIL_HARD, _QUOTA_AVAIL_SOFT, _QUOTA_USED, _TIME_BACKUP, FATTR4 dan _ACL. FATTR4 FATTR4 FATTR4 FATTR4 FATTR4 FATTR4

Upaya untuk mengatur atribut ini menghasilkan NFS4ERR_ATTRNOTSUPP kesalahan yang dikirim kembali ke klien.

Pertimbangan tambahan

Selain itu, perhatikan hal berikut:

- Untuk daftar Wilayah AWS tempat Anda dapat membuat sistem file Amazon EFS, lihat [Panduan Referensi Umum AWS Pengguna](#).
- Amazon EFS tidak mendukung opsi nconnect pemasangan.
- Anda dapat memasang sistem file Amazon EFS dari server pusat data lokal menggunakan Direct Connect dan VPN. Lihat informasi yang lebih lengkap di [Tutorial: Pemasangan dengan klien lokal](#).

Memecahkan masalah kesalahan operasi file yang terkait dengan kuota

Saat Anda mengakses sistem file EFS, batasan tertentu pada file dalam sistem file berlaku. Melebihi batas ini menyebabkan kesalahan operasi file. Untuk informasi selengkapnya tentang batas berbasis file di Amazon EFS, lihat. [Kuota Amazon EFS](#)

Berikut ini, Anda dapat menemukan beberapa kesalahan operasi file umum dan batas yang terkait dengan setiap kesalahan.

Topik

- [Memeriksa file terbuka dan kunci file](#)
- [Perintah gagal dengan kesalahan “Kuota disk terlampaui”](#)
- [Perintah gagal dengan “kesalahan I/O”](#)

- [Perintah gagal dengan kesalahan “Nama file terlalu panjang”](#)
- [Perintah gagal dengan kesalahan “File tidak ditemukan”](#)
- [Perintah gagal dengan kesalahan “Terlalu banyak tautan”](#)
- [Perintah gagal dengan kesalahan “File terlalu besar”](#)

Memeriksa file terbuka dan kunci file

Untuk memecahkan masalah kesalahan pengoperasian file, Anda dapat memeriksa apakah sistem klien Anda telah mencapai batas dengan memeriksa file terbuka dan kunci file pada titik pemasangan EFS:

- Periksa file yang terbuka — Gunakan alat sistem operasi Anda untuk membuat daftar file yang terbuka di jalur pemasangan EFS. Ini membantu mengidentifikasi apakah Anda mendekati batas file terbuka. Misalnya, di Linux Anda dapat menggunakan:

```
lsof <efs-mount-path>
```

- Periksa kunci file — Gunakan alat pemantauan kunci sistem Anda untuk membuat daftar file dengan kunci di jalur pemasangan EFS. Ini membantu mengidentifikasi apakah batas kunci tercapai. Misalnya, di Linux Anda dapat menggunakan:

```
lslocks | grep <efs-mount-path>
```

Perintah ini akan menunjukkan kepada Anda penggunaan saat ini terhadap batas EFS, membantu Anda menentukan apakah kesalahan operasi file terkait dengan pencapaian batas sistem atau layanan.

Perintah gagal dengan kesalahan “Kuota disk terlampaui”

Amazon EFS saat ini tidak mendukung kuota disk pengguna. Kesalahan ini dapat terjadi jika salah satu dari batasan berikut telah terlampaui:

- Hingga 65.536 pengguna aktif dapat membuka file secara bersamaan. Akun pengguna yang masuk beberapa kali dihitung sebagai satu pengguna aktif.
- Hingga 65.536 file dapat dibuka sekaligus untuk sebuah instance. Daftar isi direktori tidak dihitung sebagai membuka file.

- Setiap pemasangan unik pada klien dapat memperoleh hingga total 65.536 kunci per koneksi.

Tindakan yang harus diambil

Jika Anda mengalami masalah ini, Anda dapat menyelesaikannya dengan mengidentifikasi batas mana yang sebelumnya Anda lampai, dan kemudian membuat perubahan untuk memenuhi batas itu. Untuk informasi selengkapnya, lihat [Kuota untuk klien NFS](#). Untuk memeriksa penggunaan Anda saat ini, lihat [Memeriksa file terbuka dan kunci file](#).

Perintah gagal dengan “kesalahan I/O”

Kesalahan ini terjadi ketika Anda mengalami salah satu masalah berikut:

- Lebih dari 65.536 akun pengguna aktif untuk setiap instance memiliki file yang terbuka sekaligus.

Tindakan yang harus diambil

Jika mengalami masalah ini, Anda dapat mengatasinya dengan memenuhi batas file terbuka yang didukung pada instans Anda. Untuk melakukannya, kurangi jumlah pengguna aktif yang memiliki file dari sistem file Amazon EFS Anda yang terbuka secara bersamaan pada instans Anda. Untuk memeriksa penggunaan Anda saat ini, lihat [Memeriksa file terbuka dan kunci file](#).

- AWS KMS Kunci mengenkripsi sistem file Anda telah dihapus.

Tindakan yang harus diambil

Jika Anda mengalami masalah ini, Anda tidak dapat lagi mendekripsi data yang dienkripsi di bawah kunci itu, yang berarti bahwa data menjadi tidak dapat dipulihkan.

Perintah gagal dengan kesalahan “Nama file terlalu panjang”

Kesalahan ini terjadi ketika ukuran nama file atau tautan simbolisnya (symlink) terlalu panjang. Nama file memiliki batasan berikut:

- Sebuah nama bisa sampai 255 byte panjang.
- Symlink dapat berukuran hingga 4080 byte.

Tindakan yang harus diambil

Jika mengalami masalah ini, Anda dapat mengatasinya dengan mengurangi ukuran nama file atau panjang symlink Anda untuk memenuhi batas yang didukung.

Perintah gagal dengan kesalahan “File tidak ditemukan”

Kesalahan ini terjadi karena beberapa versi 32-bit Oracle E-Business suite yang lebih lama menggunakan I/O antarmuka file 32-bit, dan EFS menggunakan nomor inode 64-bit. Panggilan sistem yang mungkin gagal termasuk ``stat ()`` dan ``readdir ()``.

Tindakan yang harus diambil

Jika Anda mengalami kesalahan ini, Anda dapat mengatasinya dengan menggunakan opsi `nfs.enable_ino64=0` kernel boot. Opsi ini mengompres nomor inode EFS 64-bit menjadi 32 bit. Opsi boot kernel ditangani secara berbeda untuk distribusi Linux yang berbeda. Di Amazon Linux, aktifkan opsi ini dengan menambahkan `nfs.enable_ino64=0` kernel ke `GRUB_CMDLINE_LINUX_DEFAULT` variabel di `/etc/default/grub`. Silakan berkonsultasi dengan distribusi Anda untuk dokumentasi spesifik tentang cara mengaktifkan opsi boot kernel.

Perintah gagal dengan kesalahan “Terlalu banyak tautan”

Kesalahan ini terjadi ketika ada terlalu banyak hard link ke file. Anda dapat memiliki hingga 177 hard link dalam satu file.

Tindakan yang harus diambil

Jika Anda mengalami masalah ini, Anda dapat mengatasinya dengan mengurangi jumlah hard link ke file untuk memenuhi batas yang didukung.

Perintah gagal dengan kesalahan “File terlalu besar”

Kesalahan ini terjadi ketika file terlalu besar. Satu file dapat berukuran hingga 52.673.613.135.872 byte (47,9 TiB).

Tindakan yang harus diambil

Jika mengalami masalah ini, Anda dapat mengatasinya dengan mengurangi ukuran file untuk memenuhi batas yang didukung.

Amazon EFS API

Amazon EFS API adalah protokol jaringan berbasis [HTTP \(RFC 2616\)](#). Untuk setiap panggilan API, Anda membuat permintaan HTTP ke titik akhir Amazon EFS API khusus wilayah untuk AWS Region tempat Anda ingin mengelola sistem file. API menggunakan dokumen JSON (RFC 4627) untuk badan permintaan/respons HTTP.

Amazon EFS API adalah model RPC. Dalam model ini, ada satu set operasi tetap dan sintaks untuk setiap operasi diketahui klien tanpa interaksi sebelumnya. Di bagian berikut, Anda dapat menemukan deskripsi dari setiap operasi API menggunakan notasi RPC abstrak. Masing-masing memiliki nama operasi yang tidak muncul di kabel. Untuk setiap operasi, topik menentukan pemetaan untuk elemen permintaan HTTP.

Operasi Amazon EFS spesifik yang menjadi peta permintaan tertentu ditentukan oleh kombinasi metode permintaan (GET, PUT, POST, atau DELETE) dan pola mana yang cocok dengan Request-URI. Jika operasinya PUT atau POST, Amazon EFS mengekstrak argumen panggilan dari segmen jalur request-URI, parameter kueri, dan objek JSON di badan permintaan.

Note

Meskipun nama operasi, seperti `CreateFileSystem`, tidak muncul di kawat, nama-nama ini berarti dalam kebijakan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses untuk Amazon EFS](#).

Nama operasi juga digunakan untuk memberi nama perintah dalam alat baris perintah dan elemen SDK AWS. APIs Misalnya, ada AWS CLI perintah bernama `create-file-system` yang memetakan ke `CreateFileSystem` operasi.

Nama operasi juga muncul di AWS CloudTrail log untuk panggilan Amazon EFS API.

Titik akhir API

Endpoint adalah URL yang berfungsi sebagai titik masuk untuk layanan AWS web. Amazon EFS mendukung jenis titik akhir berikut:

- IPv4 titik akhir
- Dual-stack (keduanya IPv4 dan IPv6) titik akhir
- [Titik akhir FIPS](#)

Saat Anda membuat permintaan, Anda dapat menentukan titik akhir yang akan digunakan. Jika Anda tidak menentukan titik akhir, IPv4 titik akhir digunakan secara default. Untuk menggunakan tipe titik akhir yang berbeda, Anda harus menentukannya dalam permintaan Anda. Untuk contoh cara melakukannya, lihat [Menentukan titik akhir](#). Untuk tabel titik akhir yang tersedia, lihat titik akhir [dan kuota Amazon EFS](#).

IPv4 titik akhir

IPv4 endpoint hanya mendukung IPv4 lalu lintas. IPv4 titik akhir tersedia untuk semua Wilayah. Format IPv4 titik akhir tergantung pada Wilayah.

- Untuk IPv4 titik akhir di Wilayah AWS komersial, formatnya adalah `elasticfilesystem.region.amazonaws.com`.

Misalnya, jika Anda menentukan `elasticfilesystem.us-west-2.amazonaws.com` sebagai titik akhir, kami mengarahkan permintaan Anda ke titik akhir Wilayah Barat (Oregon) AS (us-west-2).

- Untuk IPv4 titik akhir di Wilayah Tiongkok, formatnya adalah `elasticfilesystem.region.amazonaws.com.cn`

Misalnya, titik akhir Amazon EFS API untuk Wilayah Tiongkok (Beijing) adalah `elasticfilesystem.cn-north-1.amazonaws.com.cn`

Titik akhir tumpukan ganda (IPv4 dan IPv6)

Titik akhir dual-stack mendukung keduanya IPv4 dan lalu lintas. IPv6 Saat Anda membuat permintaan ke titik akhir dual-stack, URL endpoint akan diselesaikan ke alamat IPv6 atau IPv4 alamat, tergantung pada protokol yang digunakan oleh jaringan dan klien Anda. Format titik akhir dual-stack tergantung pada Region.

- Untuk titik akhir tumpukan ganda di Wilayah AWS komersial, formatnya adalah `efs.aws-region.api.aws`

Misalnya, titik akhir tumpukan ganda untuk Wilayah Barat AS (Oregon) (us-west-2) adalah `efs.eu-west-2.api.aws`

- Untuk titik akhir tumpukan ganda di Wilayah Tiongkok, formatnya adalah `efs.region.api.amazonaws.com.cn`

Misalnya, titik akhir tumpukan ganda untuk Wilayah Tiongkok (Beijing) adalah `efs.cn-north-1.api.amazonwebservices.com.cn`

Menentukan titik akhir

Contoh berikut menunjukkan cara menentukan titik akhir untuk Wilayah AS Barat (California Utara) (`us-west-1`) menggunakan untuk Amazon EFS. AWS CLI

IPv4

```
aws efs get-rest-apis --region us-west-1 --endpoint-url https://elasticfilesystem.us-west-1.amazonaws.com
```

Tumpukan ganda

```
aws efs get-rest-apis --region us-west-1 --endpoint-url https://efs.us-west-1.api.aws
```

Versi API

Versi API yang digunakan untuk panggilan diidentifikasi oleh segmen jalur pertama dari URI permintaan, dan bentuknya adalah tanggal ISO 8601. Sebagai contoh, lihat [CreateFileSystem](#).

Dokumentasi menjelaskan versi API 2015-02-01.

Topik terkait

Bagian berikut memberikan deskripsi operasi API, cara membuat tanda tangan untuk otentikasi permintaan, dan cara memberikan izin untuk operasi API ini menggunakan kebijakan IAM.

- [Manajemen identitas dan akses untuk Amazon EFS](#)
- [Tindakan](#)
- [Tipe Data](#)

Bekerja dengan tingkat permintaan API kueri untuk Amazon EFS

Permintaan Amazon EFS API dibatasi untuk masing-masing Akun AWS berdasarkan per wilayah untuk membantu kinerja layanan. Semua panggilan Amazon EFS API bersama-sama, baik yang

berasal dari aplikasi, konsol Amazon EFS AWS CLI, atau Amazon EFS, tidak boleh melebihi tingkat permintaan API maksimum yang diizinkan. Tingkat permintaan API maksimum dapat bervariasi Wilayah AWS. Permintaan API yang dibuat dikaitkan dengan yang mendasarinya Akun AWS.

Jika permintaan API melebihi tingkat permintaan API untuk kategorinya, permintaan akan mengembalikan kode `ThrottlingException` kesalahan. Untuk mencegah kesalahan ini, pastikan aplikasi Anda tidak mencoba lagi permintaan API dengan kecepatan tinggi. Anda dapat melakukan ini dengan berhati-hati saat melakukan polling dan dengan menggunakan percobaan ulang backoff eksponensial.

Pemungutan Suara

Aplikasi Anda mungkin perlu memanggil operasi API berulang kali untuk memeriksa pembaruan dalam status. Sebelum Anda memulai polling, berikan waktu permintaan untuk berpotensi selesai. Saat Anda memulai polling, gunakan interval tidur yang sesuai di antara permintaan yang berurutan. Untuk hasil terbaik, gunakan interval tidur yang meningkat.

Pemrosesan coba ulang atau batch

Aplikasi Anda mungkin perlu mencoba ulang permintaan API setelah gagal, atau untuk memproses beberapa sumber daya (misalnya, semua sistem file Amazon EFS Anda). Untuk menurunkan tingkat permintaan API, gunakan interval tidur yang sesuai di antara permintaan yang berurutan. Untuk hasil terbaik, gunakan interval tidur yang meningkat atau variabel.

Menghitung interval tidur

Ketika Anda harus melakukan polling atau mencoba lagi permintaan API, sebaiknya gunakan algoritme backoff eksponensial untuk menghitung interval tidur antara panggilan API. Ide di balik backoff eksponensial adalah menggunakan waktu tunggu yang semakin lama antara percobaan ulang untuk respons kesalahan yang berurutan. Untuk informasi selengkapnya, dan contoh implementasi algoritme ini, lihat [Coba lagi perilaku](#) di Referensi Umum Amazon Web Services

Tindakan

Tindakan berikut didukung:

- [CreateAccessPoint](#)
- [CreateFileSystem](#)

- [CreateMountTarget](#)
- [CreateReplicationConfiguration](#)
- [CreateTags](#)
- [DeleteAccessPoint](#)
- [DeleteFileSystem](#)
- [DeleteFileSystemPolicy](#)
- [DeleteMountTarget](#)
- [DeleteReplicationConfiguration](#)
- [DeleteTags](#)
- [DescribeAccessPoints](#)
- [DescribeAccountPreferences](#)
- [DescribeBackupPolicy](#)
- [DescribeFileSystemPolicy](#)
- [DescribeFileSystems](#)
- [DescribeLifecycleConfiguration](#)
- [DescribeMountTargets](#)
- [DescribeMountTargetSecurityGroups](#)
- [DescribeReplicationConfigurations](#)
- [DescribeTags](#)
- [ListTagsForResource](#)
- [ModifyMountTargetSecurityGroups](#)
- [PutAccountPreferences](#)
- [PutBackupPolicy](#)
- [PutFileSystemPolicy](#)
- [PutLifecycleConfiguration](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateFileSystem](#)
- [UpdateFileSystemProtection](#)

CreateAccessPoint

Menciptakan titik akses EFS. Titik akses adalah tampilan khusus aplikasi ke dalam sistem file EFS yang menerapkan pengguna dan grup sistem operasi, serta jalur sistem file, untuk setiap permintaan sistem file yang dibuat melalui titik akses. Pengguna dan grup sistem operasi menimpa informasi identitas apa pun yang disediakan oleh klien NFS. Jalur sistem file dipaparkan sebagai direktori root titik akses. Aplikasi yang menggunakan titik akses hanya dapat mengakses data di direktori aplikasi sendiri dan subdirektori apa pun. Sistem file dapat memiliki maksimum 10.000 titik akses kecuali Anda meminta peningkatan. Untuk mempelajari selengkapnya, lihat [Memasang sistem file menggunakan titik akses EFS](#).

Note

Jika beberapa permintaan untuk membuat titik akses pada sistem file yang sama dikirim secara berurutan, dan sistem file mendekati batas titik akses, Anda mungkin mengalami respons pelambatan untuk permintaan ini. Ini untuk memastikan bahwa sistem file tidak melebihi batas titik akses yang dinyatakan.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:CreateAccessPoint`.

Titik akses dapat ditandai pada pembuatan. Jika tag ditentukan dalam tindakan pembuatan, IAM melakukan otorisasi tambahan pada `elasticfilesystem:TagResource` tindakan untuk memverifikasi apakah pengguna memiliki izin untuk membuat tag. Oleh karena itu, Anda harus memberikan izin eksplisit untuk menggunakan tindakan. `elasticfilesystem:TagResource` Untuk informasi selengkapnya, lihat [Memberikan izin untuk menandai sumber daya selama pembuatan](#).

Minta Sintaks

```
POST /2015-02-01/access-points HTTP/1.1
Content-type: application/json
```

```
{
  "ClientToken": "string",
  "FileSystemId": "string",
  "PosixUser": {
    "Gid": number,
    "SecondaryGids": [ number ],
```

```
    "Uid": number
  },
  "RootDirectory": {
    "CreationInfo": {
      "OwnerGid": number,
      "OwnerUid": number,
      "Permissions": "string"
    },
    "Path": "string"
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

ClientToken

Serangkaian hingga 64 karakter ASCII yang digunakan Amazon EFS untuk memastikan pembuatan idempoten.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: . +

Wajib: Ya

FileSystemId

ID dari sistem file EFS yang menyediakan akses ke titik akses.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

PosixUser

Pengguna dan grup sistem operasi diterapkan ke semua permintaan sistem file yang dibuat menggunakan titik akses.

Tipe: Objek [PosixUser](#)

Wajib: Tidak

RootDirectory

Menentukan direktori pada sistem file EFS yang diekspos oleh titik akses sebagai direktori root sistem file Anda ke klien NFS menggunakan titik akses. Klien yang menggunakan jalur akses hanya dapat mengakses direktori root dan di bawahnya. Jika `RootDirectory` > yang `Path` ditentukan tidak ada, Amazon EFS membuatnya dan menerapkan `CreationInfo` pengaturan saat klien terhubung ke titik akses. Saat menentukan `aRootDirectory`, Anda harus memberikan `Path`, dan `CreationInfo`

Amazon EFS membuat direktori root hanya jika Anda telah menyediakan `CreationInfo`: `OwnUid`, `ownGid`, dan izin untuk direktori tersebut. Jika Anda tidak memberikan informasi ini, Amazon EFS tidak membuat direktori root. Jika direktori root tidak ada, upaya untuk memasang menggunakan titik akses akan gagal.

Tipe: Objek [RootDirectory](#)

Wajib: Tidak

Tags

Membuat tag yang terkait dengan titik akses. Setiap tag adalah pasangan kunci-nilai, setiap kunci harus unik. Untuk informasi selengkapnya, lihat [Menandai AWS sumber daya](#) di Panduan Referensi AWS Umum.

Tipe: Array objek [Tag](#)

Wajib: Tidak

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "AccessPointArn": "string",
  "AccessPointId": "string",
  "ClientToken": "string",
  "FileSystemId": "string",
  "LifeCycleState": "string",
  "Name": "string",
  "OwnerId": "string",
  "PosixUser": {
    "Gid": number,
    "SecondaryGids": [ number ],
    "Uid": number
  },
  "RootDirectory": {
    "CreationInfo": {
      "OwnerGid": number,
      "OwnerUid": number,
      "Permissions": "string"
    },
    "Path": "string"
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[AccessPointArn](#)

Nama Sumber Daya Amazon (ARN) unik yang terkait dengan titik akses.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}$`

AccessPointId

ID titik akses, yang ditetapkan oleh Amazon EFS.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

ClientToken

String buram ditentukan dalam permintaan untuk memastikan pembuatan yang idempotensi.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

FileSystemId

ID dari sistem file EFS di mana titik akses berlaku.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

LifeCycleState

Mengidentifikasi fase siklus hidup dari titik akses.

Tipe: String

Nilai yang Valid: `creating` | `available` | `updating` | `deleting` | `deleted` | `error`

Name

Nama titik akses. Ini adalah nilai Name tag.

Tipe: String

OwnerId

Mengidentifikasi Akun AWS yang memiliki sumber daya titik akses.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^\d{12})|(\d{4}-\d{4}-\d{4})$`

PosixUser

Identitas POSIX lengkap, termasuk ID pengguna, ID grup, dan grup sekunder IDs pada titik akses yang digunakan untuk semua operasi file oleh klien NFS menggunakan titik akses.

Tipe: Objek [PosixUser](#)

RootDirectory

Direktori pada sistem file EFS yang diekspos oleh titik akses sebagai direktori root ke klien NFS menggunakan titik akses.

Tipe: Objek [RootDirectory](#)

Tags

Tag yang terkait dengan titik akses, disajikan sebagai array objek Tag.

Tipe: Array objek [Tag](#)

Kesalahan

AccessPointAlreadyExists

Dikembalikan jika titik akses yang Anda coba buat sudah ada, dengan token pembuatan yang Anda berikan dalam permintaan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

AccessPointLimitExceeded

Dikembalikan jika Akun AWS telah menciptakan jumlah maksimum titik akses yang diizinkan per sistem file. Untuk informasi lebih lanjut, lihat. <https://docs.aws.amazon.com/efs/latest/ug/limits.html#limits-efs-resources-per-account-per-region>

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 403

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak "tersedia".

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ThrottlingException

Dikembalikan ketika tindakan `CreateAccessPoint` API dipanggil terlalu cepat dan jumlah Titik Akses pada sistem file mendekati [batas 120](#).

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 429

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

CreateFileSystem

Membuat sistem file baru yang kosong. Operasi ini memerlukan token pembuatan dalam permintaan yang digunakan Amazon EFS untuk memastikan pembuatan idempoten (memanggil operasi dengan token pembuatan yang sama tidak berpengaruh). Jika sistem file saat ini tidak ada yang dimiliki oleh pemanggil Akun AWS dengan token pembuatan yang ditentukan, operasi ini melakukan hal berikut:

- Membuat sistem file baru yang kosong. Sistem file akan memiliki ID yang ditetapkan Amazon EFS, dan status siklus hidup awal. `creating`
- Kembali dengan deskripsi sistem file yang dibuat.

Jika tidak, operasi ini mengembalikan `FileSystemAlreadyExists` kesalahan dengan ID dari sistem file yang ada.

Note

Untuk kasus penggunaan dasar, Anda dapat menggunakan UUID yang dibuat secara acak untuk token pembuatan.

Operasi idempoten memungkinkan Anda untuk mencoba lagi `CreateFileSystem` panggilan tanpa risiko membuat sistem file tambahan. Ini bisa terjadi ketika panggilan awal gagal dengan cara yang membuatnya tidak pasti apakah sistem file benar-benar dibuat atau tidak. Contohnya mungkin batas waktu tingkat transport terjadi atau koneksi Anda disetel ulang. Selama Anda menggunakan token pembuatan yang sama, jika panggilan awal berhasil membuat sistem file, klien dapat mengetahui keberadaannya dari `FileSystemAlreadyExists` kesalahan.

Untuk informasi selengkapnya, lihat [Membuat sistem file](#) di Panduan Pengguna Amazon EFS.

Note

`CreateFileSystem` Panggilan kembali saat status siklus hidup sistem file masih `creating` Anda dapat memeriksa status pembuatan sistem file dengan memanggil [DescribeFileSystems](#) operasi, yang antara lain mengembalikan status sistem file.

Operasi ini menerima `PerformanceMode` parameter opsional yang Anda pilih untuk sistem file Anda. Kami merekomendasikan `generalPurpose` `PerformanceMode` untuk semua sistem file.

maxIOMode ini adalah tipe kinerja generasi sebelumnya yang dirancang untuk beban kerja yang sangat paralel yang dapat mentolerir latensi yang lebih tinggi daripada mode. generalPurpose MaxIOmode tidak didukung untuk sistem file One Zone atau sistem file yang menggunakan throughput Elastis.

Tidak PerformanceMode dapat diubah setelah sistem file dibuat. Untuk informasi selengkapnya, lihat [mode kinerja Amazon EFS](#).

Anda dapat mengatur mode throughput untuk sistem file menggunakan ThroughputMode parameter.

Setelah sistem file dibuat sepenuhnya, Amazon EFS menetapkan status siklus hidupnyaavailable, di mana Anda dapat membuat satu atau beberapa target mount untuk sistem file di VPC Anda. Untuk informasi selengkapnya, lihat [CreateMountTarget](#). Anda memasang sistem file Amazon EFS pada EC2 instance di VPC Anda dengan menggunakan target pemasangan. Untuk informasi selengkapnya, lihat [Amazon EFS: Cara Kerjanya](#).

Operasi ini memerlukan izin untuk tindakan elasticfilesystem:CreateFileSystem.

Sistem file dapat ditandai pada pembuatan. Jika tag ditentukan dalam tindakan pembuatan, IAM melakukan otorisasi tambahan pada elasticfilesystem:TagResource tindakan untuk memverifikasi apakah pengguna memiliki izin untuk membuat tag. Oleh karena itu, Anda harus memberikan izin eksplisit untuk menggunakan tindakan. elasticfilesystem:TagResource Untuk informasi selengkapnya, lihat [Memberikan izin untuk menandai sumber daya selama pembuatan](#).

Minta Sintaks

```
POST /2015-02-01/file-systems HTTP/1.1
Content-type: application/json

{
  "AvailabilityZoneName": "string",
  "Backup": boolean,
  "CreationToken": "string",
  "Encrypted": boolean,
  "KmsKeyId": "string",
  "PerformanceMode": "string",
  "ProvisionedThroughputInMibps": number,
  "Tags": [
    {
```

```
    "Key": "string",  
    "Value": "string"  
  },  
  "ThroughputMode": "string"  
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

AvailabilityZoneName

Untuk sistem file One Zone, tentukan AWS Availability Zone untuk membuat sistem file. Gunakan format `us-east-1a` untuk menetapkan Availability Zone. Untuk informasi selengkapnya tentang sistem file One Zone, lihat [tipe sistem file EFS](#) di Panduan Pengguna Amazon EFS.

Note

Sistem file One Zone tidak tersedia di semua Availability Zone di Wilayah AWS mana Amazon EFS tersedia.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

Wajib: Tidak

Backup

Menentukan apakah backup otomatis diaktifkan pada sistem file yang Anda buat. Tetapkan nilainya `true` untuk mengaktifkan pencadangan otomatis. Jika Anda membuat sistem file One Zone, backup otomatis diaktifkan secara default. Untuk informasi selengkapnya, lihat [Pencadangan otomatis](#) di Panduan Pengguna Amazon EFS.

Default-nya adalah `false`. Namun, jika Anda menentukan `AvailabilityZoneName`, defaultnya adalah `true`.

 Note

AWS Backup tidak tersedia di semua Wilayah AWS tempat Amazon EFS tersedia.

Tipe: Boolean

Wajib: Tidak

CreationToken

Sebuah string hingga 64 karakter ASCII. Amazon EFS menggunakan ini untuk memastikan kreasi idempoten.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

Wajib: Ya

Encrypted

Nilai Boolean yang, jika `BETUL`, menciptakan sistem file terenkripsi. Saat membuat sistem file terenkripsi, Anda memiliki opsi untuk menentukan kunci yang ada (AWS Key Management Service kunci KMS). Jika Anda tidak menentukan kunci KMS, maka kunci KMS default untuk Amazon EFS, `/aws/elasticfilesystem`, digunakan untuk melindungi sistem file terenkripsi.

Tipe: Boolean

Wajib: Tidak

KmsKeyId

ID kunci KMS yang ingin Anda gunakan untuk melindungi sistem file terenkripsi. Parameter ini diperlukan hanya jika Anda ingin menggunakan kunci KMS non-default. Jika parameter ini tidak ditentukan, kunci KMS default untuk Amazon EFS digunakan. Anda dapat menentukan ID kunci KMS menggunakan format berikut:

- ID kunci - Pengidentifikasi unik dari kunci, misalnya `1234abcd-12ab-34cd-56ef-1234567890ab`.

- ARN - Amazon Resource Name (ARN) untuk kunci, misalnya `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- Alias kunci - Nama tampilan yang dibuat sebelumnya untuk kunci, misalnya `alias/projectKey1`.
- ARN alias kunci - ARN untuk alias kunci, misalnya `arn:aws:kms:us-west-2:444455556666:alias/projectKey1`.

[Jika Anda menggunakan `KmsKeyId`, Anda harus mengatur parameter `Encrypted` ke `true`.
\[CreateFileSystem\]\(#\)](#)

Important

EFS hanya menerima kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris dengan sistem file Amazon EFS.

Tipe: String

Batasan Panjang: Panjang maksimum 2048.

Pola: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

Wajib: Tidak

[PerformanceMode](#)

Mode performa sistem file. Kami merekomendasikan mode `generalPurpose` kinerja untuk semua sistem file. Sistem file yang menggunakan mode performa `maxIO` dapat menskalakan ke tingkat throughput dan operasi agregat per detik yang lebih tinggi dengan kompromi latensi yang sedikit lebih tinggi untuk sebagian besar operasi file. Mode performa tidak dapat diubah setelah sistem file dibuat. `maxIO` mode ini tidak didukung pada sistem file One Zone.

Important

Karena latensi per operasi yang lebih tinggi dengan Max I/O, sebaiknya gunakan mode kinerja Tujuan Umum untuk semua sistem file.

Default-nya adalah `generalPurpose`.

Tipe: String

Nilai yang Valid: `generalPurpose` | `maxIO`

Wajib: Tidak

ProvisionedThroughputInMibps

Throughput, diukur dalam mebibytes per detik (MiBps), yang ingin Anda sediakan untuk sistem file yang Anda buat. Harus diisi jika `ThroughputMode` diatur ke `provisioned`. Nilai yang valid adalah 1-3414 MiBps, dengan batas atas tergantung pada Wilayah. Untuk meningkatkan batas ini, hubungi Dukungan. Untuk informasi selengkapnya, lihat [Kuota Amazon EFS yang dapat Anda tingkatkan](#) di Panduan Pengguna Amazon EFS.

Tipe: Ganda

Rentang Valid: Nilai minimum 1.0.

Wajib: Tidak

Tags

Gunakan untuk membuat satu tanda atau lebih yang terkait dengan sistem file. Setiap tanda adalah pasangan nilai-kunci yang ditentukan pengguna. Nama sistem file Anda pada pembuatan dengan menyertakan `"Key": "Name"`, `"Value": "{value}"` pasangan nilai-kunci. Setiap kunci harus unik. Untuk informasi selengkapnya, lihat [Menandai AWS sumber daya](#) di Panduan Referensi AWS Umum.

Tipe: Array objek [Tag](#)

Wajib: Tidak

ThroughputMode

Menentukan modus throughput untuk sistem file. Modusnya bisa `bursting`, `provisioned`, atau `elastic`. Jika Anda mengatur `ThroughputMode` ke `provisioned`, Anda juga harus mengatur nilai `ProvisionedThroughputInMibps`. Setelah Anda membuat sistem file, Anda dapat mengurangi throughput `Provisioned` sistem file Anda atau mengubah antara mode throughput, dengan batasan waktu tertentu. Untuk informasi selengkapnya, lihat [Menentukan throughput dengan mode yang disediakan](#) di Panduan Pengguna Amazon EFS.

Default-nya adalah `bursting`.

Tipe: String

Nilai yang Valid: bursting | provisioned | elastic

Wajib: Tidak

Sintaksis Respons

HTTP/1.1 201

Content-type: application/json

```
{
  "AvailabilityZoneId": "string",
  "AvailabilityZoneName": "string",
  "CreationTime": number,
  "CreationToken": "string",
  "Encrypted": boolean,
  "FileSystemArn": "string",
  "FileSystemId": "string",
  "FileSystemProtection": {
    "ReplicationOverwriteProtection": "string"
  },
  "KmsKeyId": "string",
  "LifecycleState": "string",
  "Name": "string",
  "NumberOfMountTargets": number,
  "OwnerId": "string",
  "PerformanceMode": "string",
  "ProvisionedThroughputInMibps": number,
  "SizeInBytes": {
    "Timestamp": number,
    "Value": number,
    "ValueInArchive": number,
    "ValueInIA": number,
    "ValueInStandard": number
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "ThroughputMode": "string"
```

```
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respon HTTP 201.

Layanan mengembalikan data berikut dalam format JSON.

[AvailabilityZoneId](#)

Pengidentifikasi unik dan konsisten dari Availability Zone di mana sistem file berada, dan hanya berlaku untuk sistem file One Zone. Misalnya, use1-az1 adalah ID Availability Zone untuk AWS Region us-east-1, dan memiliki lokasi yang sama di setiap. Akun AWS

Tipe: String

[AvailabilityZoneName](#)

Menjelaskan AWS Availability Zone di mana sistem file berada, dan hanya berlaku untuk sistem file One Zone. Untuk informasi selengkapnya, lihat [Menggunakan kelas penyimpanan EFS](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: . +

[CreationTime](#)

Waktu sistem file dibuat, dalam hitungan detik (sejak 1970-01-01T 00:00:00 Z).

Tipe: Timestamp

[CreationToken](#)

String buram ditentukan dalam permintaan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: . +

[Encrypted](#)

Nilai Boolean yang, jika benar, menunjukkan bahwa sistem file dienkrpsi.

Jenis: Boolean

[FileSystemArn](#)

Nama Sumber Daya Amazon (ARN) untuk sistem file EFS, dalam format.

`arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id` Contoh dengan data sampel: `arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567`

Tipe: String

[FileSystemId](#)

ID sistem file, yang ditetapkan oleh Amazon EFS.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[FileSystemProtection](#)

Menjelaskan perlindungan pada sistem file.

Tipe: Objek [FileSystemProtectionDescription](#)

[KmsKeyId](#)

ID yang AWS KMS key digunakan untuk melindungi sistem file terenkripsi.

Tipe: String

Batasan Panjang: Panjang maksimum 2048.

Pola: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

[LifeCycleState](#)

Fase siklus hidup dari sistem file.

Tipe: String

Nilai yang Valid: `creating` | `available` | `updating` | `deleting` | `deleted` | `error`

Name

Anda dapat menambahkan tag ke sistem file, termasuk Name tag. Untuk informasi selengkapnya, lihat [CreateFileSystem](#). Jika sistem file memiliki Name tag, Amazon EFS mengembalikan nilai di bidang ini.

Tipe: String

Batasan Panjang: Panjang maksimum 256.

Pola: `^([\p{L}\p{Z}\p{N}_.:\/+=\-\@]*)$`

NumberOfMountTargets

Jumlah target mount saat ini yang dimiliki sistem file. Untuk informasi selengkapnya, lihat [CreateMountTarget](#).

Jenis: Integer

Rentang yang Valid: Nilai minimum 0.

OwnerId

Akun AWS Yang menciptakan sistem file.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

PerformanceMode

Mode performa sistem file.

Tipe: String

Nilai yang Valid: `generalPurpose` | `maxIO`

ProvisionedThroughputInMibps

Jumlah throughput yang disediakan, diukur dalam MiBps, untuk sistem file. Berlaku untuk sistem file menggunakan `ThroughputMode` set `toprovisioned`.

Tipe: Ganda

Rentang Valid: Nilai minimum 1.0.

SizeInBytes

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan dalam sistem file, di Value bidangnya, dan waktu di mana ukuran itu ditentukan di bidangnyaTimestamp.

TimestampNilainya adalah bilangan bulat detik sejak 1970-01-01T 00:00:00 Z.

SizeInBytesNilai tidak mewakili ukuran snapshot yang konsisten dari sistem file, tetapi pada akhirnya konsisten ketika tidak ada penulisan ke sistem file. Artinya, SizeInBytes mewakili ukuran sebenarnya hanya jika sistem file tidak dimodifikasi untuk jangka waktu lebih dari beberapa jam. Jika tidak, nilainya bukan ukuran yang tepat dari sistem file pada setiap titik waktu.

Tipe: Objek [FileSystemSize](#)

Tags

Tag yang terkait dengan sistem file, disajikan sebagai array Tag objek.

Tipe: Array objek [Tag](#)

ThroughputMode

Menampilkan mode throughput sistem file. Untuk informasi selengkapnya, lihat [Mode throughput](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Nilai yang Valid: `bursting` | `provisioned` | `elastic`

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemAlreadyExists

Dikembalikan jika sistem file yang Anda coba buat sudah ada, dengan token pembuatan yang Anda berikan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

FileSystemLimitExceeded

Dikembalikan jika Akun AWS telah membuat jumlah maksimum sistem file yang diizinkan per akun.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 403

InsufficientThroughputCapacity

Dikembalikan jika tidak ada kapasitas yang cukup untuk menyediakan throughput tambahan. Nilai ini mungkin dikembalikan saat Anda mencoba membuat sistem file dalam mode throughput yang disediakan, saat Anda mencoba meningkatkan throughput yang disediakan dari sistem file yang ada, atau saat Anda mencoba mengubah sistem file yang ada dari Bursting Throughput ke mode Throughput Terprovisioned. Coba lagi nanti.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 503

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 telah tercapai. MiB/s

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

UnsupportedAvailabilityZone

Dikembalikan jika fungsionalitas Amazon EFS yang diminta tidak tersedia di Availability Zone yang ditentukan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Contoh

Buat sistem file EFS terenkripsi

Contoh berikut mengirimkan permintaan POST untuk membuat sistem file di us-west-2 Wilayah dengan backup otomatis diaktifkan. Permintaan ditentukan myFileSystem1 sebagai token penciptaan untuk idempotensi.

Permintaan Sampel

```
POST /2015-02-01/file-systems HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215117Z
Authorization: <...>
Content-Type: application/json
Content-Length: 42

{
  "CreationToken" : "myFileSystem1",
  "PerformanceMode" : "generalPurpose",
  "Backup": true,
  "Encrypted": true,
  "Tags": [
    {
```

```
        "Key": "Name",
        "Value": "Test Group1"
    }
]
```

Contoh Respons

```
HTTP/1.1 201 Created
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 319

{
  "ownerId":"251839141158",
  "CreationToken":"myFileSystem1",
  "Encrypted": true,
  "PerformanceMode" : "generalPurpose",
  "fileSystemId":"fs-01234567",
  "CreationTime":"1403301078",
  "LifeCycleState":"creating",
  "numberOfMountTargets":0,
  "SizeInBytes":{"
    "Timestamp": 1403301078,
    "Value": 29313618372,
    "ValueInArchive": 201156,
    "ValueInIA": 675432,
    "ValueInStandard": 29312741784
  },
  "Tags":[
    {
      "Key": "Name",
      "Value": "Test Group1"
    }
  ],
  "ThroughputMode": "elastic"
}
```

Buat sistem file EFS terenkripsi dengan ketersediaan One Zone

Contoh berikut mengirimkan permintaan POST untuk membuat sistem file di us-west-2 Wilayah dengan backup otomatis diaktifkan. Sistem file akan memiliki penyimpanan One Zone di us-west-2b Availability Zone.

Permintaan Sampel

```
POST /2015-02-01/file-systems HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215117Z
Authorization: <...>
Content-Type: application/json
Content-Length: 42
```

```
{
  "CreationToken" : "myFileSystem2",
  "PerformanceMode" : "generalPurpose",
  "Backup": true,
  "AvailabilityZoneName": "us-west-2b",
  "Encrypted": true,
  "ThroughputMode": "elastic",
  "Tags":[
    {
      "Key": "Name",
      "Value": "Test Group1"
    }
  ]
}
```

Contoh Respons

```
HTTP/1.1 201 Created
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 319
```

```
{
  "ownerId":"251839141158",
  "CreationToken":"myFileSystem1",
  "Encrypted": true,
  "AvailabilityZoneId": "usew2-az2",
  "AvailabilityZoneName": "us-west-2b",
  "PerformanceMode" : "generalPurpose",
  "fileSystemId":"fs-01234567",
  "CreationTime":"1403301078",
  "LifeCycleState":"creating",
  "numberOfMountTargets":0,
  "SizeInBytes":{
```



```
    "Timestamp": 1403301078,  
    "Value": 29313618372,  
    "ValueInArchive": 201156,  
    "ValueInIA": 675432,  
    "ValueInStandard": 29312741784  
  },  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "Test Group1"  
    }  
  ],  
  "ThroughputMode": "elastic"  
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

CreateMountTarget

Membuat target mount untuk sistem file. Anda kemudian dapat me-mount sistem file pada EC2 instance dengan menggunakan target mount. Untuk informasi selengkapnya tentang target pemasangan, lihat [Mengelola target pemasangan](#).

Untuk membuat target pemasangan untuk sistem file EFS, status siklus hidup sistem file harus `available`. Untuk informasi selengkapnya, lihat [DescribeFileSystems](#).

Setelah membuat target pemasangan, Amazon EFS mengembalikan respons yang menyertakan alamat IP `MountTargetId` dan (`IpAddress` atau `Ipv6Address`). Anda menggunakan alamat IP ini saat memasang sistem file dalam sebuah EC2 instance. Anda juga dapat menggunakan nama DNS target mount saat memasang sistem file. EC2 Instance tempat Anda memasang sistem file dengan menggunakan target mount dapat menyelesaikan nama DNS target mount ke alamat IP-nya. Untuk informasi selengkapnya, lihat [Cara kerja Amazon EFS](#).

Perhatikan bahwa Anda dapat membuat target mount untuk sistem file hanya dalam satu VPC, dan hanya ada satu target mount per Availability Zone. Untuk informasi selengkapnya, lihat [Membuat target pemasangan](#).

Jika permintaan memenuhi persyaratan, Amazon EFS melakukan hal berikut:

- Membuat target mount baru di subnet yang ditentukan.
- Membuat antarmuka jaringan baru di subnet dengan following:
 - Deskripsi di Mount target `fsmt-id` for file system `fs-id` `fsmt-id` mana ID target mount, dan `fs-id` merupakan `FileSystemId`.
 - `requesterManaged` properti antarmuka jaringan diatur ke `true`, dan `requesterId` nilai diatur ke EFS.

Setiap target mount memiliki satu antarmuka jaringan yang dikelola pemohon yang EC2 sesuai. Setelah antarmuka jaringan dibuat, Amazon EFS menetapkan `NetworkInterfaceId` bidang dalam deskripsi target pemasangan ke ID antarmuka jaringan, dan alamat IP ke alamatnya. Jika pembuatan antarmuka jaringan gagal, seluruh `CreateMountTarget` operasi gagal.

Note

`CreateMountTarget` Panggilan kembali hanya setelah membuat antarmuka jaringan, tetapi saat status target mount masih `creating`, Anda dapat memeriksa status pembuatan target

mount dengan memanggil [DescribeMountTargets](#) operasi, yang antara lain mengembalikan status target mount.

Operasi ini memerlukan izin untuk tindakan berikut pada sistem file:

- `elasticfilesystem:CreateMountTarget`

Operasi ini juga memerlukan izin untuk EC2 tindakan Amazon berikut:

- `ec2:DescribeSubnets`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`

Minta Sintaks

```
POST /2015-02-01/mount-targets HTTP/1.1
Content-type: application/json
```

```
{
  "FileSystemId": "string",
  "IpAddress": "string",
  "IpAddressType": "string",
  "Ipv6Address": "string",
  "SecurityGroups": [ "string" ],
  "SubnetId": "string"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

[FileSystemId](#)

ID sistem file yang akan digunakan untuk membuat target pemasangan.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

IpAddress

Jika target `IpAddressType` untuk mount adalah IPv4 (`IPV4_ONLY` atau `DUAL_STACK`), maka tentukan IPv4 alamat yang akan digunakan. Jika Anda tidak menentukan `IpAddress`, Amazon EFS memilih alamat IP yang tidak digunakan dari subnet yang ditentukan untuk `SubnetId`.

Tipe: String

Batasan Panjang: Panjang minimum 7. Panjang maksimum 15.

Pola: `^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

Wajib: Tidak

IpAddressType

Jenis alamat IP untuk target pemasangan. Nilai yang mungkin adalah `IPV4_ONLY` (hanya IPv4 alamat), `IPV6_ONLY` (hanya IPv6 alamat), dan `DUAL_STACK` (dual-stack, keduanya IPv4 dan IPv6 alamat). Jika Anda tidak menentukan `IpAddressType`, maka `IPV4_ONLY` digunakan.

Note

`IpAddressType` harus cocok dengan jenis IP subnet. Selain itu, `IpAddressType` parameter mengesampingkan nilai yang ditetapkan sebagai alamat IP default untuk subnet di VPC. Misalnya, jika `IpAddressType` is `IPV4_ONLY` dan `AssignIpv6AddressOnCreation` is `true`, maka IPv4 digunakan untuk target mount. Untuk informasi selengkapnya, lihat [Memodifikasi atribut pengalamatan IP subnet Anda](#).

Tipe: String

Nilai yang Valid: `IPV4_ONLY` | `IPV6_ONLY` | `DUAL_STACK`

Wajib: Tidak

Ipv6Address

Jika target `IpAddressType` untuk mount adalah IPv6 (`IPV6_ONLY` atau `DUAL_STACK`), maka tentukan IPv6 alamat yang akan digunakan. Jika Anda tidak menentukan `Ipv6Address`, Amazon EFS memilih alamat IP yang tidak digunakan dari subnet yang ditentukan untuk `SubnetId`.

Tipe: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 39.

Wajib: Tidak

SecurityGroups

Grup keamanan VPC IDs, dari formulir `sg-xxxxxxx`. Ini harus untuk VPC yang sama dengan subnet yang ditentukan. Jumlah maksimum grup keamanan tergantung pada kuota akun. Untuk informasi selengkapnya, lihat [Kuota VPC](#) Amazon di Panduan Pengguna Amazon VPC (lihat tabel Grup Keamanan). Jika Anda tidak menentukan grup keamanan, Amazon EFS menggunakan grup keamanan default untuk VPC subnet.

Tipe: Array string

Anggota Array: Jumlah maksimum 100 item.

Kendala Panjang: Panjang minimum 11. Panjang maksimum 43.

Pola: `^sg-[0-9a-f]{8,40}`

Wajib: Tidak

SubnetId

ID subnet yang ditambahkan target pemasangan. Untuk sistem file One Zone, gunakan subnet yang terkait dengan Availability Zone sistem file. Jenis subnet harus tipe yang sama dengan `IpAddressType`.

Tipe: String

Kendala Panjang: Panjang minimum 15. Panjang maksimum 47.

Pola: `^subnet-[0-9a-f]{8,40}$`

Diperlukan: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "AvailabilityZoneId": "string",
  "AvailabilityZoneName": "string",
  "FileSystemId": "string",
  "IpAddress": "string",
  "Ipv6Address": "string",
  "LifeCycleState": "string",
  "MountTargetId": "string",
  "NetworkInterfaceId": "string",
  "OwnerId": "string",
  "SubnetId": "string",
  "VpcId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[AvailabilityZoneId](#)

Pengidentifikasi unik dan konsisten dari Availability Zone tempat target mount berada. Misalnya, use1-az1 adalah ID AZ untuk Wilayah us-east-1 dan memiliki lokasi yang sama di setiap wilayah. Akun AWS

Tipe: String

[AvailabilityZoneName](#)

Nama Availability Zone di mana target mount berada. Availability Zones dipetakan secara independen ke nama masing-masing Akun AWS. Misalnya, Availability Zone us-east-1a untuk Anda Akun AWS mungkin bukan lokasi yang sama dengan us-east-1a yang lain Akun AWS.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: . +

[FileSystemId](#)

ID sistem file tempat target mount dimaksudkan.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[IpAddress](#)

IPv4 Alamat untuk target pemasangan.

Tipe: String

Batasan Panjang: Panjang minimum 7. Panjang maksimum 15.

Pola: `^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

[Ipv6Address](#)

IPv6 Alamat untuk target pemasangan.

Tipe: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 39.

[LifeCycleState](#)

Status siklus hidup target pemasangan.

Tipe: String

Nilai yang Valid: `creating | available | updating | deleting | deleted | error`

[MountTargetId](#)

ID target pemasangan yang ditetapkan sistem.

Tipe: String

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsm-t-[0-9a-f]{8,40}$`

NetworkInterfaceId

ID antarmuka jaringan yang dibuat Amazon EFS saat membuat target pemasangan.

Tipe: String

OwnerId

Akun AWS ID yang memiliki sumber daya.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: $^(\backslash d\{12\}) | (\backslash d\{4\} - \backslash d\{4\} - \backslash d\{4\})\$$

SubnetId

ID subnet target mount.

Tipe: String

Kendala Panjang: Panjang minimum 15. Panjang maksimum 47.

Pola: $^subnet - [0-9a-f]\{8,40\}\$$

VpcId

ID virtual private cloud (VPC) tempat target mount dikonfigurasi.

Tipe: String

Kesalahan

AvailabilityZonesMismatch

Dikembalikan jika Availability Zone yang ditentukan untuk target mount berbeda dari Availability Zone sistem file. Untuk informasi selengkapnya, lihat [Redundansi penyimpanan Regional dan Satu Zona](#).

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

IpAddressInUse

Dikembalikan jika permintaan ditentukan alamat IP (`IpAddress` atau `Ipv6Address`) yang sudah digunakan di subnet.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

MountTargetConflict

Dikembalikan jika target pemasangan akan melanggar salah satu batasan yang ditentukan berdasarkan target pemasangan sistem file yang ada.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

NetworkInterfaceLimitExceeded

Akun panggilan telah mencapai batas untuk antarmuka jaringan elastis untuk spesifik AWS Region. Hapus beberapa antarmuka jaringan atau minta kuota akun dinaikkan. Untuk informasi selengkapnya, lihat [Kuota VPC Amazon](#) di Panduan Pengguna Amazon VPC (lihat entri Antarmuka jaringan per Wilayah di tabel Antarmuka jaringan).

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

NoFreeAddressesInSubnet

Tidak Ipv6Address dikembalikan IPAddress atau disediakan dalam permintaan dan tidak ada alamat IP gratis di subnet yang ditentukan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

SecurityGroupLimitExceeded

Dikembalikan jika jumlah SecurityGroups yang ditentukan dalam permintaan lebih besar dari batas, yang didasarkan pada kuota akun. Hapus beberapa grup keamanan atau minta kuota akun dinaikkan. Untuk informasi selengkapnya, lihat [Kuota VPC](#) Amazon di Panduan Pengguna Amazon VPC (lihat tabel Grup Keamanan).

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

SecurityGroupNotFound

Dikembalikan jika salah satu grup keamanan yang ditentukan tidak ada di virtual private cloud (VPC) subnet.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

SubnetNotFound

Dikembalikan jika tidak ada subnet dengan ID SubnetId yang disediakan dalam permintaan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

UnsupportedAvailabilityZone

Dikembalikan jika fungsionalitas Amazon EFS yang diminta tidak tersedia di Availability Zone yang ditentukan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Contoh

Buat target pemasangan di IPv4 alamat yang tersedia di subnet

Permintaan berikut hanya menentukan sistem file dan subnet untuk target mount. Target dibuat pada IPv4 alamat yang tersedia di subnet yang ditentukan, dengan grup keamanan default yang terkait dengan VPC.

Permintaan Sampel

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{"SubnetId": "subnet-748c5d03", "FileSystemId": "fs-01234567"}
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
  "OwnerId": "231243201240",
  "MountTargetId": "fsmt-55a4413c",
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-01234567",
  "LifecycleState": "available",
  "IpAddress": "172.31.22.183"
  "NetworkInterfaceId": "eni-1bcb7772"
  "AvailabilityZoneId": "eus1-az2",
  "AvailabilityZoneName": "eu-south-1b",
  "VpcId": "vpc-08d45b31fa009a15e"
}
```

Buat target pemasangan di IPv4 alamat tertentu

Permintaan berikut menentukan sistem file, subnet, grup keamanan, dan IPv4 alamat yang akan digunakan untuk target mount.

Permintaan Sampel

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-01234567",
  "IpAddress": "10.0.2.42",
  "SecurityGroups": [
    "sg-01234567"
  ]
}
```


Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
  "OwnerId": "251839141158",
  "MountTargetId": "fsmt-9a13661e",
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-fd04ff94",
  "LifeCycleState": "available",
  "IpAddress": "10.0.2.42",
  "NetworkInterfaceId": "eni-1bcb7772",
  "AvailabilityZoneId": "eus1-az2",
  "AvailabilityZoneName": "eu-south-1b",
  "VpcId": "vpc-08d45b31fa009a15e"
}
```

Buat target pemasangan di IPv6 alamat tertentu

Permintaan berikut menentukan sistem file, subnet, grup keamanan, dan IPv6 alamat yang akan digunakan untuk target mount.

Permintaan Sampel

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-01234567",
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "IpAddressType": "IPV6_ONLY",
  "SecurityGroups": [
    "sg-01234567"
  ]
}
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
  "OwnerId": "251839141158",
  "MountTargetId": "fsmt-9a13661e",
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-fd04ff94",
  "LifeCycleState": "available",
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "NetworkInterfaceId": "eni-1bcb7772",
  "AvailabilityZoneId": "eus1-az2",
  "AvailabilityZoneName": "eu-south-1b",
  "VpcId": "vpc-08d45b31fa009a15e"
}
```

Buat target pemasangan di IPv4 alamat yang tersedia di subnet dual-stack

Permintaan berikut menentukan sistem file, subnet, dan jenis alamat untuk target mount. Target dibuat pada IPv4 alamat yang tersedia pada subnet dual-stack yang ditentukan, dengan grup keamanan default yang terkait dengan VPC.

Permintaan Sampel

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{"SubnetId": "subnet-748c5d03", "FileSystemId": "fs-01234567"}
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252
```

```
{
  "OwnerId": "251839141158",
  "MountTargetId": "fsmt-55a4413c",
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-fd04ff94",
  "LifecycleState": "available",
  "IpAddress": "172.31.22.183"
  "Ipv6Address": "2a05:d01a:419:8611:3e57:75ab:5719:b238",
  "NetworkInterfaceId": "eni-01234567",
  "AvailabilityZoneId": "eus1-az2",
  "AvailabilityZoneName": "eu-south-1b",
  "VpcId": "vpc-08d45b31fa009a15e"
}
```

Buat target mount pada Ipv4 tertentu dan IPv6 alamat pada subnet dual-stack

Permintaan berikut menentukan sistem file, subnet, grup keamanan, IPv4 alamat, IPv6 alamat untuk target mount. Target dibuat pada yang ditentukan IPv4 dan IPv6 alamat pada subnet dual-stack.

Permintaan Sampel

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-01234567",
  "IpAddressType": "DUAL_STACK",
  "IpAddress": "10.0.1.25",
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "SecurityGroups": [
    "sg-01234567"
  ]
}
```

Contoh Respons

```
HTTP/1.1 200 OK
```

```
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
  "OwnerId": "231243201240",
  "MountTargetId": "fsmt-55a4413c",
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-01234567",
  "LifecycleState": "available",
  "IpAddress": "10.0.1.25",
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "NetworkInterfaceId": "eni-01234567",
  "AvailabilityZoneId": "eus1-az2",
  "AvailabilityZoneName": "eu-south-1b",
  "VpcId": "vpc-08d45b31fa009a15e"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

CreateReplicationConfiguration

Membuat konfigurasi replikasi ke sistem file EFS baru atau yang sudah ada. Untuk informasi selengkapnya, lihat [replikasi Amazon EFS](#) di Panduan Pengguna Amazon EFS. Konfigurasi replikasi menentukan hal berikut:

- Sistem file sumber — Sistem file EFS yang ingin Anda tiru.
- Sistem file tujuan — Sistem file tujuan tempat sistem file sumber direplikasi. Hanya ada satu sistem file tujuan dalam konfigurasi replikasi.

Note

Sistem file dapat menjadi bagian dari hanya satu konfigurasi replikasi.

Parameter tujuan untuk konfigurasi replikasi bergantung pada apakah Anda mereplikasi ke sistem file baru atau ke sistem file yang ada, dan jika Anda mereplikasi di seluruh Akun AWS. Untuk informasi selengkapnya, lihat [DestinationToCreate](#).

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:CreateReplicationConfiguration`. Selain itu, izin lain diperlukan tergantung pada bagaimana Anda mereplikasi sistem file. Untuk informasi selengkapnya, lihat [izin yang diperlukan untuk replikasi](#) di Panduan Pengguna Amazon EFS.

Minta Sintaks

```
POST /2015-02-01/file-systems/SourceFileSystemId/replication-configuration HTTP/1.1
Content-type: application/json
```

```
{
  "Destinations": [
    {
      "AvailabilityZoneName": "string",
      "FileSystemId": "string",
      "KmsKeyId": "string",
      "Region": "string",
      "RoleArn": "string"
    }
  ]
}
```

```
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[SourceFileSystemId](#)

Menentukan sistem file Amazon EFS yang ingin Anda replikasi. Sistem file ini belum dapat menjadi sumber atau sistem file tujuan dalam konfigurasi replikasi lain.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

[Destinations](#)

Sebuah array objek konfigurasi tujuan. Hanya satu objek konfigurasi tujuan yang didukung.

Tipe: Array objek [DestinationToCreate](#)

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "CreationTime": number,
  "Destinations": [
    {
      "FileSystemId": "string",
      "LastReplicatedTimestamp": number,
```

```
    "OwnerId": "string",
    "Region": "string",
    "RoleArn": "string",
    "Status": "string",
    "StatusMessage": "string"
  }
],
"OriginalSourceFileSystemArn": "string",
"SourceFileSystemArn": "string",
"SourceFileSystemId": "string",
"SourceFileSystemOwnerId": "string",
"SourceFileSystemRegion": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

CreationTime

Menjelaskan kapan konfigurasi replikasi dibuat.

Tipe: Timestamp

Destinations

Sebuah array objek tujuan. Hanya satu objek tujuan yang didukung.

Tipe: Array objek [Destination](#)

OriginalSourceFileSystemArn

Nama Sumber Daya Amazon (ARN) dari sistem file EFS sumber asli dalam konfigurasi replikasi.

Tipe: String

SourceFileSystemArn

Nama Sumber Daya Amazon (ARN) dari sistem file sumber saat ini dalam konfigurasi replikasi.

Tipe: String

SourceFileSystemId

ID dari sistem file Amazon EFS sumber yang sedang direplikasi.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

SourceFileSystemOwnerId

ID Akun AWS tempat sistem file sumber berada.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

SourceFileSystemRegion

AWS Region Di mana sistem file EFS sumber berada.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

ConflictException

Dikembalikan jika sistem file sumber dalam replikasi dienkripsi tetapi sistem file tujuan tidak dienkripsi.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

FileSystemLimitExceeded

Dikembalikan jika Akun AWS telah membuat jumlah maksimum sistem file yang diizinkan per akun.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 403

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

`InsufficientThroughputCapacity`

Dikembalikan jika tidak ada kapasitas yang cukup untuk menyediakan throughput tambahan. Nilai ini mungkin dikembalikan saat Anda mencoba membuat sistem file dalam mode throughput yang disediakan, saat Anda mencoba meningkatkan throughput yang disediakan dari sistem file yang ada, atau saat Anda mencoba mengubah sistem file yang ada dari Bursting Throughput ke mode Throughput Terprovisioned. Coba lagi nanti.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 503

`InternalServerError`

Dikembalikan jika terjadi kesalahan di sisi server.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ReplicationNotFound

Dikembalikan jika sistem file yang ditentukan tidak memiliki konfigurasi replikasi.

ErrorCode

ReplicationNotFound

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 telah tercapai. MiB/s

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

UnsupportedAvailabilityZone

Dikembalikan jika fungsionalitas Amazon EFS yang diminta tidak tersedia di Availability Zone yang ditentukan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia AWS Region di mana permintaan dibuat.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

CreateTags

Note

DEPRECATED - tidak digunakan lagi dan CreateTags tidak dipertahankan. Untuk membuat tag untuk resource EFS, gunakan aksi [TagResource](#) API.

Membuat atau menerima tag yang terkait dengan sistem file. Setiap tag adalah pasangan nilai kunci. Jika kunci tag yang ditentukan dalam permintaan sudah ada di sistem file, operasi ini menerima nilainya dengan nilai yang disediakan dalam permintaan. Jika Anda menambahkan Name tag ke sistem file Anda, Amazon EFS mengembalikannya sebagai respons terhadap [DescribeFileSystems](#) operasi.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:CreateTags`.

Minta Sintaks

```
POST /2015-02-01/create-tags/FileSystemId HTTP/1.1
Content-type: application/json
```

```
{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[FileSystemId](#)

ID sistem file yang tagnya ingin Anda ubah (String). Operasi ini memodifikasi tag saja, bukan sistem file.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

Tags

Array Tag objek untuk ditambahkan. Setiap Tag objek adalah pasangan kunci-nilai.

Tipe: Array objek [Tag](#)

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 204
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan isi HTTP kosong.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DeleteAccessPoint

Menghapus titik akses yang ditentukan. Setelah penghapusan selesai, klien baru tidak dapat lagi terhubung ke titik akses. Klien yang terhubung ke titik akses pada saat penghapusan akan terus berfungsi sampai mereka mengakhiri koneksi mereka.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DeleteAccessPoint`.

Minta Sintaks

```
DELETE /2015-02-01/access-points/AccessPointId HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

AccessPointId

ID titik akses yang ingin Anda hapus.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 204
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan isi HTTP kosong.

Kesalahan

AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

InternalServerError

Kembali jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DeleteFileSystem

Menghapus sistem file, secara permanen memutuskan akses ke isinya. Setelah kembali, sistem file tidak ada lagi dan Anda tidak dapat mengakses konten apa pun dari sistem file yang dihapus.

Anda perlu menghapus target pemasangan yang terpasang ke sistem file secara manual sebelum Anda dapat menghapus sistem file EFS. Langkah ini dilakukan untuk Anda saat Anda menggunakan AWS konsol untuk menghapus sistem file.

Note

Anda tidak dapat menghapus sistem file yang merupakan bagian dari konfigurasi replikasi EFS. Anda perlu menghapus konfigurasi replikasi terlebih dahulu.

Anda tidak dapat menghapus sistem file yang sedang digunakan. Artinya, jika sistem file memiliki target mount, Anda harus menghapusnya terlebih dahulu. Untuk informasi selengkapnya, lihat [DescribeMountTargets](#) dan [DeleteMountTarget](#).

Note

DeleteFileSystemPanggilan kembali saat status sistem file masih `deleting`. Anda dapat memeriksa status penghapusan sistem file dengan memanggil [DescribeFileSystems](#) operasi, yang mengembalikan daftar sistem file di akun Anda. Jika Anda meneruskan ID sistem file atau token penciptaan untuk sistem file yang dihapus, [DescribeFileSystems](#) mengembalikan `404 FileSystemNotFound` kesalahan.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DeleteFileSystem`.

Minta Sintaks

```
DELETE /2015-02-01/file-systems/FileSystemId HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

ID sistem file yang ingin Anda hapus.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 204
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan isi HTTP kosong.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

`FileSystemInUse`

Dikembalikan jika sistem file memiliki target mount.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

`FileSystemNotFound`

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Contoh

Hapus sistem file

Contoh berikut mengirimkan permintaan DELETE ke `file-systems` endpoint (`elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems/fs-01234567`) untuk menghapus sistem file yang ID-nya `fs-01234567`.

Permintaan Sampel

```
DELETE /2015-02-01/file-systems/fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T233021Z
Authorization: <...>
```

Contoh Respons

```
HTTP/1.1 204 No Content
x-amzn-RequestId: a2d125b3-7ebd-4d6a-ab3d-5548630bff33
```

Content-Length: 0

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DeleteFileSystemPolicy

Menghapus FileSystemPolicy untuk sistem file yang ditentukan. Default FileSystemPolicy akan berlaku setelah kebijakan yang ada dihapus. Untuk informasi selengkapnya tentang kebijakan sistem file default, lihat [Menggunakan Kebijakan Berbasis Sumber Daya dengan EFS](#).

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DeleteFileSystemPolicy`.

Minta Sintaks

```
DELETE /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[FileSystemId](#)

Menentukan sistem file EFS untuk menghapus file. FileSystemPolicy

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DeleteMountTarget

Menghapus target mount yang ditentukan.

Operasi ini secara paksa merusak setiap mount sistem file dengan menggunakan target mount yang sedang dihapus, yang dapat mengganggu instance atau aplikasi yang menggunakan mount tersebut. Untuk menghindari aplikasi terputus secara tiba-tiba, Anda dapat mempertimbangkan untuk melepasudukan target pemasangan, jika memungkinkan. Operasi ini juga menghapus antarmuka jaringan terkait. Penulisan yang tidak berkomitmen mungkin hilang, tetapi melanggar target pemasangan menggunakan operasi ini tidak merusak sistem file itu sendiri. Sistem file yang Anda buat tetap ada. Anda dapat memasang EC2 instance di VPC Anda dengan menggunakan target pemasangan lain.

Operasi ini memerlukan izin untuk tindakan berikut pada sistem file:

- `elasticfilesystem:DeleteMountTarget`

Note

`DeleteMountTarget` Panggilan kembali saat status target mount `deleting`. Anda dapat memeriksa penghapusan target mount dengan memanggil [DescribeMountTargets](#) operasi, yang menampilkan daftar deskripsi target mount untuk sistem file yang diberikan.

Operasi ini juga memerlukan izin untuk EC2 tindakan Amazon berikut pada antarmuka jaringan target pemasangan:

- `ec2:DeleteNetworkInterface`

Minta Sintaks

```
DELETE /2015-02-01/mount-targets/MountTargetId HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

MountTargetId

ID target mount untuk dihapus (String).

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsm-[0-9a-f]{8,40}$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 204
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan isi HTTP kosong.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

DependencyTimeout

Layanan habis waktu mencoba memenuhi permintaan, dan klien harus mencoba panggilan lagi.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 504

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil.
Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

Contoh

Hapus target pemasangan sistem file

Contoh berikut mengirimkan permintaan DELETE untuk menghapus target mount tertentu.

Permintaan Sampel

```
DELETE /2015-02-01/mount-targets/fsmt-9a13661e HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T232908Z
Authorization: <...>
```

Contoh Respons

```
HTTP/1.1 204 No Content
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DeleteReplicationConfiguration

Menghapus konfigurasi replikasi. Menghapus konfigurasi replikasi mengakhiri proses replikasi. Setelah konfigurasi replikasi dihapus, sistem file tujuan menjadi Writeable dan perlindungan penyimpanan replikasi diaktifkan kembali. Untuk informasi selengkapnya, lihat [Menghapus konfigurasi replikasi](#).

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DeleteReplicationConfiguration`.

Minta Sintaks

```
DELETE /2015-02-01/file-systems/SourceFileSystemId/replication-configuration?
deletionMode=DeletionMode HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[DeletionMode](#)

Saat mereplikasi di seluruh Akun AWS atau di seluruh Wilayah AWS, Amazon EFS menghapus konfigurasi replikasi dari akun sumber dan tujuan atau Region (ALL_CONFIGURATIONS) secara default. Jika ada masalah konfigurasi atau izin yang mencegah Amazon EFS menghapus konfigurasi replikasi dari kedua sisi, Anda dapat menggunakan LOCAL_CONFIGURATION_ONLY mode untuk menghapus konfigurasi replikasi hanya dari sisi lokal (akun atau Wilayah tempat penghapusan dilakukan).

Note

Hanya gunakan LOCAL_CONFIGURATION_ONLY mode jika Amazon EFS tidak dapat menghapus konfigurasi replikasi di akun sumber dan tujuan atau Wilayah. Menghapus konfigurasi lokal membuat konfigurasi di akun lain atau Wilayah tidak dapat dipulihkan. Selain itu, jangan gunakan mode ini untuk akun yang sama, replikasi wilayah yang sama karena hal itu menghasilkan kesalahan pengecualian. BadRequest

Nilai Valid: ALL_CONFIGURATIONS | LOCAL_CONFIGURATION_ONLY

SourceFileSystemId

ID dari sistem file sumber dalam konfigurasi replikasi.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 204
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan isi HTTP kosong.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ReplicationNotFound

Dikembalikan jika sistem file yang ditentukan tidak memiliki konfigurasi replikasi.

ErrorCode

ReplicationNotFound

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DeleteTags

Note

DEPRECATED - tidak digunakan lagi dan DeleteTags tidak dipertahankan. Untuk menghapus tag dari sumber daya EFS, gunakan tindakan [UntagResource](#) API.

Menghapus tag yang ditentukan dari sistem file. Jika DeleteTags permintaan menyertakan kunci tag yang tidak ada, Amazon EFS mengabaikannya dan tidak menyebabkan kesalahan. Untuk informasi selengkapnya tentang tag dan batasan terkait, lihat [Pembatasan tag](#) di Panduan AWS Manajemen Penagihan dan Biaya Pengguna.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DeleteTags`.

Minta Sintaks

```
POST /2015-02-01/delete-tags/FileSystemId HTTP/1.1
Content-type: application/json

{
  "TagKeys": [ "string" ]
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[FileSystemId](#)

ID sistem file yang tagnya ingin Anda hapus (String).

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

[TagKeys](#)

Daftar kunci tag untuk dihapus.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `^(?![aA]{1}[wW]{1}[sS]{1}:)([\\p{L}\\p{Z}\\p{N}_.: /+=\\-@]+)$`

Diperlukan: Ya

Sintaksis Respons

```
HTTP/1.1 204
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan isi HTTP kosong.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeAccessPoints

Mengembalikan deskripsi titik akses Amazon EFS tertentu jika `AccessPointId` disediakan. Jika Anda memberikan `EFSFileSystemId`, EFS akan mengembalikan deskripsi semua titik akses untuk sistem file tersebut. Anda dapat memberikan salah satu `AccessPointId` atau `FileSystemId` dalam permintaan, tetapi tidak keduanya.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeAccessPoints`.

Minta Sintaks

```
GET /2015-02-01/access-points?  
AccessPointId=AccessPointId&FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken  
HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

AccessPointId

(Opsional) Menentukan titik akses EFS untuk dijelaskan dalam tanggapan; saling eksklusif dengan `FileSystemId`.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

FileSystemId

(Opsional) Jika Anda menyediakan `FileSystemId`, EFS mengembalikan semua titik akses untuk sistem file tersebut; saling eksklusif dengan `AccessPointId`.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[MaxResults](#)

(Opsional) Saat mengambil semua titik akses untuk sistem file, Anda dapat secara opsional menentukan `MaxItems` parameter untuk membatasi jumlah objek yang dikembalikan dalam respons. Nilai default-nya adalah 100.

Rentang yang Valid: Nilai minimum 1.

[NextToken](#)

`NextToken` hadir jika responsnya diberi halaman. Anda dapat menggunakan `NextMarker` dalam permintaan berikutnya untuk mengambil halaman berikutnya dari deskripsi titik akses.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `.+`

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "AccessPoints": [
    {
      "AccessPointArn": "string",
      "AccessPointId": "string",
      "ClientToken": "string",
      "FileSystemId": "string",
      "LifeCycleState": "string",
      "Name": "string",
      "OwnerId": "string",
      "PosixUser": {
        "Gid": number,
        "SecondaryGids": [ number ],
        "Uid": number
      },
      "RootDirectory": {
```

```
    "CreationInfo": {
      "OwnerId": number,
      "OwnerUid": number,
      "Permissions": "string"
    },
    "Path": "string"
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
},
"NextToken": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[AccessPoints](#)

Array deskripsi titik akses.

Tipe: Array objek [AccessPointDescription](#)

[NextToken](#)

Hadir jika ada lebih banyak titik akses daripada yang dikembalikan dalam respons. Anda dapat menggunakan NextMarker dalam permintaan berikutnya untuk mengambil deskripsi tambahan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

Kesalahan

AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeAccountPreferences

Mengembalikan pengaturan preferensi akun untuk Akun AWS yang terkait dengan pengguna yang membuat permintaan, saat ini AWS Region.

Minta Sintaks

```
GET /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

MaxResults

(Opsional) Saat mengambil preferensi akun, Anda dapat menentukan `MaxItems` parameter secara opsional untuk membatasi jumlah objek yang dikembalikan dalam respons. Nilai default-nya adalah 100.

Tipe: Integer

Rentang yang Valid: Nilai minimum 1.

Wajib: Tidak

NextToken

(Opsional) Anda dapat menggunakan `NextToken` permintaan berikutnya untuk mengambil halaman Akun AWS preferensi berikutnya jika payload respons diberi paginasi.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

Diperlukan: Tidak

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "ResourceIdPreference": {
    "ResourceIdType": "string",
    "Resources": [ "string" ]
  }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[NextToken](#)

Hadir jika ada lebih banyak catatan daripada yang dikembalikan dalam respons. Anda dapat menggunakan NextToken dalam permintaan berikutnya untuk mengambil deskripsi tambahan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

[ResourceIdPreference](#)

Menjelaskan pengaturan preferensi ID sumber daya untuk Akun AWS yang terkait dengan pengguna yang membuat permintaan, saat ini AWS Region.

Tipe: Objek [ResourceIdPreference](#)

Kesalahan

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeBackupPolicy

Mengembalikan kebijakan cadangan untuk sistem file EFS yang ditentukan.

Minta Sintaks

```
GET /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[FileSystemId](#)

Menentukan sistem file EFS mana untuk mengambil file. BackupPolicy

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "BackupPolicy": {
    "Status": "string"
  }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[BackupPolicy](#)

Menjelaskan kebijakan pencadangan sistem file, yang menunjukkan apakah pencadangan otomatis diaktifkan atau dimatikan.

Tipe: Objek [BackupPolicy](#)

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

PolicyNotFound

Dikembalikan jika kebijakan IAM atau kebijakan sumber daya yang ditentukan tidak dapat ditemukan atau diakses. Bergantung pada tindakan yang dilakukan, kesalahan juga dapat dikembalikan jika kebijakan sistem file default berlaku untuk sistem file EFS atau jika tidak ada kebijakan cadangan yang ditentukan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia AWS Region di mana permintaan dibuat.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)

- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeFileSystemPolicy

Mengembalikan FileSystemPolicy untuk sistem file EFS tertentu.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeFileSystemPolicy`.

Minta Sintaks

```
GET /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

Menentukan sistem file EFS mana yang akan mengambil untuk. FileSystemPolicy

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "FileSystemId": "string",
  "Policy": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

FileSystemId

Menentukan sistem file EFS yang FileSystemPolicy berlaku.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Policy

JSON diformat FileSystemPolicy untuk sistem file EFS.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 20000.

Pola: `[\s\S]+`

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

PolicyNotFound

Dikembalikan jika kebijakan IAM atau kebijakan sumber daya yang ditentukan tidak dapat ditemukan atau diakses. Bergantung pada tindakan yang dilakukan, kesalahan juga dapat dikembalikan jika kebijakan sistem file default berlaku untuk sistem file EFS atau jika tidak ada kebijakan cadangan yang ditentukan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

Contoh

Contoh

Contoh ini menggambarkan salah satu penggunaan. `DescribeFileSystemPolicy`

Permintaan Sampel

```
GET /2015-02-01/file-systems/fs-01234567/policy HTTP/1.1
```

Contoh Respons

```
{
  "FileSystemId": "fs-01234567",
  "Policy": "{
    "Version": "2012-10-17",
    "Id": "efs-policy-wizard-cdef0123-aaaa-6666-5555-444455556666",
```

```

    "Statement": [
      {
        "Sid": "efs-statement-abcdef01-1111-bbbb-2222-111122224444",
        "Effect" : "Deny",
        "Principal": {
          "AWS": "*"
        },
        "Action": "*",
        "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-
system/fs-01234567",
        "Condition": {
          "Bool": {
            "aws:SecureTransport": "false"
          }
        }
      },
      {
        "Sid": "efs-statement-01234567-aaaa-3333-4444-111122223333",
        "Effect": "Allow",
        "Principal": {
          "AWS": "*"
        },
        "Action": [
          "elasticfilesystem:ClientMount",
          "elasticfilesystem:ClientWrite"
        ],
        "Resource" : "arn:aws:elasticfilesystem:us-east-2:111122223333:file-
system/fs-01234567"
      }
    ]
  }
}

```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)

- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeFileSystems

Mengembalikan deskripsi sistem file Amazon EFS tertentu jika sistem file `CreationToken` atau `FileSystemId` disediakan. Jika tidak, ia mengembalikan deskripsi semua sistem file yang dimiliki oleh pemanggil Akun AWS di titik AWS Region akhir yang Anda panggil.

Saat mengambil semua deskripsi sistem file, Anda dapat menentukan `MaxItems` parameter secara opsional untuk membatasi jumlah deskripsi dalam respons. Nomor ini secara otomatis diatur ke 100. Jika lebih banyak deskripsi sistem file tetap ada, Amazon EFS mengembalikan `NextMarker`, token buram, dalam respons. Dalam hal ini, Anda harus mengirim permintaan berikutnya dengan parameter `Marker` permintaan yang diatur ke nilai `NextMarker`.

Untuk mengambil daftar deskripsi sistem file Anda, operasi ini digunakan dalam proses iteratif, di mana `DescribeFileSystems` dipanggil pertama tanpa `Marker` dan kemudian operasi terus memanggilnya dengan `Marker` parameter diatur ke nilai `NextMarker` dari dari respon sebelumnya sampai respon memiliki tidak. `NextMarker`

Urutan sistem file yang dikembalikan dalam respons satu `DescribeFileSystems` panggilan dan urutan sistem file yang dikembalikan di seluruh respons iterasi multi-panggilan tidak ditentukan.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeFileSystems`.

Minta Sintaks

```
GET /2015-02-01/file-systems?
CreationToken=CreationToken&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems
HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

CreationToken

(Opsional) Membatasi daftar ke sistem file dengan token penciptaan ini (String). Anda menentukan token pembuatan saat membuat sistem file Amazon EFS.

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: . +

FileSystemId

(Opsional) ID dari sistem file yang deskripsinya ingin Anda ambil (String).

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Marker

(Opsional) Token pagination buram dikembalikan dari DescribeFileSystems operasi sebelumnya (String). Jika ada, menentukan untuk melanjutkan daftar dari mana panggilan kembali telah ditinggalkan.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `.+`

MaxItems

(Opsional) Menentukan jumlah maksimum sistem file untuk kembali dalam respon (integer). Nomor ini secara otomatis diatur ke 100. Responsnya diberi halaman 100 per halaman jika Anda memiliki lebih dari 100 sistem file.

Rentang yang Valid: Nilai minimum 1.

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "FileSystems": [
    {
      "AvailabilityZoneId": "string",
      "AvailabilityZoneName": "string",
      "CreationTime": number,
      "CreationToken": "string",
```

```

    "Encrypted": boolean,
    "FileSystemArn": "string",
    "FileSystemId": "string",
    "FileSystemProtection": {
        "ReplicationOverwriteProtection": "string"
    },
    "KmsKeyId": "string",
    "LifeCycleState": "string",
    "Name": "string",
    "NumberOfMountTargets": number,
    "OwnerId": "string",
    "PerformanceMode": "string",
    "ProvisionedThroughputInMibps": number,
    "SizeInBytes": {
        "Timestamp": number,
        "Value": number,
        "ValueInArchive": number,
        "ValueInIA": number,
        "ValueInStandard": number
    },
    "Tags": [
        {
            "Key": "string",
            "Value": "string"
        }
    ],
    "ThroughputMode": "string"
}
},
"Marker": "string",
"NextMarker": "string"
}

```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[FileSystems](#)

Array deskripsi sistem file.

Tipe: Array objek [FileSystemDescription](#)

Marker

Hadir jika disediakan oleh pemanggil dalam permintaan (String).

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: . +

NextMarker

Hadir jika ada lebih banyak sistem file daripada dikembalikan dalam respons (String). Anda dapat menggunakan `NextMarker` dalam permintaan berikutnya untuk mengambil deskripsi.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: . +

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Kembali jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Contoh

Ambil daftar 10 sistem file

Contoh berikut mengirimkan permintaan GET ke `file-systems` endpoint

(`elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems`).

Permintaan menentukan parameter `MaxItems` query untuk membatasi jumlah deskripsi sistem file untuk 10.

Permintaan Sampel

```
GET /2015-02-01/file-systems?MaxItems=10 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191208Z
Authorization: <...>
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 499
{
  "FileSystems":[
    {
      "OwnerId":"251839141158",
      "CreationToken":"MyFileSystem1",
      "FileSystemId":"fs-01234567",
      "PerformanceMode" : "generalPurpose",
      "CreationTime":"1403301078",
      "LifeCycleState":"created",
      "Name":"my first file system",
      "NumberOfMountTargets":1,
      "SizeInBytes":{
        "Timestamp": 1403301078,
        "Value": 29313618372,
        "ValueInArchive": 201156,
        "ValueInIA": 675432,
        "ValueInStandard": 29312741784
      }
    }
  ]
}
```

```
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeLifecycleConfiguration

Mengembalikan LifecycleConfiguration objek saat ini untuk sistem file EFS tertentu. Manajemen siklus hidup menggunakan LifecycleConfiguration objek untuk mengidentifikasi kapan harus memindahkan file antar kelas penyimpanan. Untuk sistem file tanpa LifecycleConfiguration objek, panggilan mengembalikan array kosong dalam respons.

Operasi ini memerlukan izin untuk `elasticfilesystem:DescribeLifecycleConfiguration` operasi.

Minta Sintaks

```
GET /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

ID dari sistem file yang LifecycleConfiguration objeknya ingin Anda ambil (String).

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "LifecyclePolicies": [
```



```
{
  "TransitionToArchive": "string",
  "TransitionToIA": "string",
  "TransitionToPrimaryStorageClass": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[LifecyclePolicies](#)

Array kebijakan manajemen siklus hidup. EFS mendukung maksimal satu kebijakan per sistem file.

Tipe: Array objek [LifecyclePolicy](#)

Anggota Array: Jumlah maksimum 3 item.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Contoh

Mengambil konfigurasi siklus hidup untuk sistem file

Permintaan berikut mengambil LifecycleConfiguration objek untuk sistem file tertentu.

Permintaan Sampel

```
GET /2015-02-01/file-systems/fs-01234567/lifecycle-configuration HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20181120T221118Z
Authorization: <...>
```

Contoh Respons

```
HTTP/1.1 200 OK
    x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
    Content-Type: application/json
    Content-Length: 86
{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "AFTER_270_DAYS"
    },
    {
      "TransitionToIA": "AFTER_14_DAYS"
    },
    {
      "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
    }
  ]
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)

- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeMountTargets

Mengembalikan deskripsi semua target mount saat ini, atau target mount tertentu, untuk sistem file. Saat meminta semua target mount saat ini, urutan target mount yang dikembalikan dalam respons tidak ditentukan.

Operasi ini memerlukan izin untuk `elasticfilesystem:DescribeMountTargets` tindakan, baik pada ID sistem file yang Anda tentukan `FileSystemId`, atau pada sistem file target mount yang Anda tentukan. `MountTargetId`

Perintah CLI yang sesuai adalah. `describe-mount-targets`

Minta Sintaks

```
GET /2015-02-01/mount-targets?  
AccessPointId=AccessPointId&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems&MountTarg  
HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

AccessPointId

(Opsional) ID titik akses yang target mount yang ingin Anda daftarkan. Itu harus dimasukkan dalam permintaan Anda jika `MountTargetId` ada `FileSystemId` atau tidak termasuk dalam permintaan Anda. Menerima ID titik akses atau ARN sebagai input.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

FileSystemId

(Opsional) ID dari sistem file yang target mount Anda ingin daftar (String). Itu harus dimasukkan dalam permintaan Anda jika `AccessPointId` atau `MountTargetId` tidak termasuk. Menerima ID sistem file atau ARN sebagai input.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Marker

(Opsional) Token pagination buram dikembalikan dari DescribeMountTargets operasi sebelumnya (String). Jika ada, ini menentukan untuk melanjutkan daftar dari mana panggilan kembali sebelumnya ditinggalkan.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `.+`

MaxItems

(Opsional) Jumlah maksimum target pemasangan untuk kembali dalam respons. Saat ini, nomor ini secara otomatis diatur ke 10, dan nilai lainnya diabaikan. Respons diberi paginasi pada 100 per halaman jika Anda memiliki lebih dari 100 target pemasangan.

Rentang yang Valid: Nilai minimum 1.

MountTargetId

(Opsional) ID dari target mount yang ingin Anda jelaskan (String). Itu harus dimasukkan dalam permintaan Anda jika FileSystemId tidak termasuk. Menerima ID target mount atau ARN sebagai input.

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsmt-[0-9a-f]{8,40}$`

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "Marker": "string",
  "MountTargets": [
```

```

{
  "AvailabilityZoneId": "string",
  "AvailabilityZoneName": "string",
  "FileSystemId": "string",
  "IpAddress": "string",
  "Ipv6Address": "string",
  "LifeCycleState": "string",
  "MountTargetId": "string",
  "NetworkInterfaceId": "string",
  "OwnerId": "string",
  "SubnetId": "string",
  "VpcId": "string"
},
"NextMarker": "string"
}

```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Marker

Jika permintaan termasuk `Marker`, respon mengembalikan nilai dalam bidang ini.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

MountTargets

Mengembalikan target mount sistem file sebagai array `MountTargetDescription` objek.

Tipe: Array objek [MountTargetDescription](#)

NextMarker

Jika ada nilai, ada lebih banyak target mount untuk dikembalikan. Dalam permintaan berikutnya, Anda dapat memberikan nilai ini `Marker` dalam permintaan Anda untuk mengambil set target mount berikutnya.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: . +

Kesalahan

AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil.

Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

Contoh

Ambil deskripsi target mount yang dibuat untuk sistem file

Permintaan berikut mengambil deskripsi target mount yang dibuat untuk sistem file yang ditentukan.

Permintaan Sampel

```
GET /2015-02-01/mount-targets?FileSystemId=fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191252Z
Authorization: <...>
```

Contoh Respons

```
HTTP/1.1 200 OK
```

```
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 357

{
  "MountTargets": [
    {
      "OwnerId": "251839141158",
      "MountTargetId": "fsmt-01234567",
      "FileSystemId": "fs-01234567",
      "SubnetId": "subnet-01234567",
      "LifeCycleState": "added",
      "IpAddress": "10.0.2.42",
      "NetworkInterfaceId": "eni-1bcb7772"
    }
  ]
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeMountTargetSecurityGroups

Mengembalikan grup keamanan yang saat ini berlaku untuk target mount. Operasi ini mengharuskan antarmuka jaringan target pemasangan telah dibuat dan status siklus hidup target pemasangan tidak `deleted`.

Operasi ini memerlukan izin untuk tindakan berikut:

- `elasticfilesystem:DescribeMountTargetSecurityGroups` tindakan pada sistem file target mount.
- `ec2:DescribeNetworkInterfaceAttribute` aksi pada antarmuka jaringan target mount.

Minta Sintaks

```
GET /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

MountTargetId

ID target mount yang grup keamanannya ingin Anda ambil.

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsmt-[0-9a-f]{8,40}$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json
```

```
{  
  "SecurityGroups": [ "string" ]  
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[SecurityGroups](#)

Berbagai kelompok keamanan.

Tipe: Array string

Anggota Array: Jumlah maksimum 100 item.

Kendala Panjang: Panjang minimum 11. Panjang maksimum 43.

Pola: `^sg-[0-9a-f]{8,40}`

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

`IncorrectMountTargetState`

Dikembalikan jika target pemasangan tidak dalam keadaan yang benar untuk operasi.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

`InternalServerError`

Dikembalikan jika terjadi kesalahan di sisi server.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil.

Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

Contoh

Mengambil grup keamanan yang berlaku untuk sistem file

Contoh berikut mengambil grup keamanan yang berlaku untuk antarmuka jaringan yang terkait dengan target mount.

Permintaan Sampel

```
GET /2015-02-01/mount-targets/fsmt-9a13661e/security-groups HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T223513Z
Authorization: <...>
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

Content-Length: 57

```
{  
  "SecurityGroups" : [  
    "sg-188d9f74"  
  ]  
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeReplicationConfigurations

Mengambil konfigurasi replikasi untuk sistem file tertentu. Jika sistem file tidak ditentukan, semua konfigurasi replikasi untuk Akun AWS in AWS Region diambil.

Minta Sintaks

```
GET /2015-02-01/file-systems/replication-configurations?  
FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[FileSystemId](#)

Anda dapat mengambil konfigurasi replikasi untuk sistem file tertentu dengan memberikan ID sistem file-nya. Untuk replikasi lintas akun, lintas wilayah, akun hanya dapat menjelaskan konfigurasi replikasi untuk sistem file di Wilayahnya sendiri.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[MaxResults](#)

(Opsional) Untuk membatasi jumlah objek yang dikembalikan dalam respons, Anda dapat menentukan `MaxItems` parameternya. Nilai default-nya adalah 100.

Rentang yang Valid: Nilai minimum 1.

[NextToken](#)

`NextToken` hadir jika responsnya diberi halaman. Anda dapat menggunakan `NextToken` permintaan berikutnya untuk mengambil halaman output berikutnya.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `.+`

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "Replications": [
    {
      "CreationTime": number,
      "Destinations": [
        {
          "FileSystemId": "string",
          "LastReplicatedTimestamp": number,
          "OwnerId": "string",
          "Region": "string",
          "RoleArn": "string",
          "Status": "string",
          "StatusMessage": "string"
        }
      ],
      "OriginalSourceFileSystemArn": "string",
      "SourceFileSystemArn": "string",
      "SourceFileSystemId": "string",
      "SourceFileSystemOwnerId": "string",
      "SourceFileSystemRegion": "string"
    }
  ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

Anda dapat menggunakan NextToken dari respons sebelumnya dalam permintaan berikutnya untuk mengambil deskripsi tambahan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: . +

Replications

Kumpulan konfigurasi replikasi yang dikembalikan.

Tipe: Array objek [ReplicationConfigurationDescription](#)

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ReplicationNotFound

Dikembalikan jika sistem file yang ditentukan tidak memiliki konfigurasi replikasi.

ErrorCode

ReplicationNotFound

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia AWS Region di mana permintaan dibuat.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)

- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeTags

Note

DEPRECATED - DescribeTags Tindakan ini tidak digunakan lagi dan tidak dipertahankan. Untuk melihat tag yang terkait dengan sumber daya EFS, gunakan tindakan ListTagsForResource API.

Mengembalikan tag yang terkait dengan sistem file. Urutan tag yang dikembalikan dalam respons satu DescribeTags panggilan dan urutan tag yang dikembalikan di seluruh respons iterasi beberapa panggilan (saat menggunakan pagination) tidak ditentukan.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeTags`.

Minta Sintaks

```
GET /2015-02-01/tags/FileSystemId?Marker=Marker&MaxItems=MaxItems HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

ID dari sistem file yang tag set Anda ingin mengambil.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Marker

(Opsional) Token pagination buram dikembalikan dari DescribeTags operasi sebelumnya (String). Jika ada, ini menentukan untuk melanjutkan daftar dari mana panggilan sebelumnya ditinggalkan.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: . +

[MaxItems](#)

(Opsional) Jumlah maksimum tag sistem file untuk dikembalikan dalam respons. Saat ini, nomor ini secara otomatis diatur ke 100, dan nilai lainnya diabaikan. Responsnya diberi halaman 100 per halaman jika Anda memiliki lebih dari 100 tag.

Rentang yang Valid: Nilai minimum 1.

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "Marker": "string",
  "NextMarker": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[Marker](#)

Jika permintaan menyertakan `aMarker`, respons mengembalikan nilai itu di bidang ini.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: . +

[NextMarker](#)

Jika ada nilai, ada lebih banyak tag untuk dikembalikan. Dalam permintaan berikutnya, Anda dapat memberikan nilai `NextMarker` sebagai nilai `Marker` parameter dalam permintaan berikutnya untuk mengambil set tag berikutnya.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: . +

[Tags](#)

Mengembalikan tag yang terkait dengan sistem file sebagai array Tag objek.

Tipe: Array objek [Tag](#)

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Contoh

Mengambil tag yang terkait dengan sistem file

Permintaan berikut mengambil tag (pasangan kunci-nilai) yang terkait dengan sistem file yang ditentukan.

Permintaan Sampel

```
GET /2015-02-01/tags/fs-01234567/ HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215404Z
Authorization: <...>
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 288

{
  "Tags": [
    {
      "Key": "Name",
      "Value": "my first file system"
    },
    {
      "Key": "Fleet",
      "Value": "Development"
    },
    {
      "Key": "Developer",
      "Value": "Alice"
    }
  ]
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

ListTagsForResource

Daftar semua tag untuk sumber daya EFS tingkat atas. Anda harus memberikan ID sumber daya yang ingin Anda ambil tag.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeAccessPoints`.

Minta Sintaks

```
GET /2015-02-01/resource-tags/ResourceId?MaxResults=MaxResults&NextToken=NextToken
HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[MaxResults](#)

(Opsional) Menentukan jumlah maksimum objek tag untuk kembali dalam respon. Nilai default-nya adalah 100.

Rentang yang Valid: Nilai minimum 1.

[NextToken](#)

(Opsional) Anda dapat menggunakan NextToken permintaan berikutnya untuk mengambil halaman berikutnya dari deskripsi titik akses jika payload respons diberi paginasi.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

[ResourceId](#)

Menentukan sumber daya EFS yang ingin Anda ambil tag untuk. Anda dapat mengambil tag untuk sistem file EFS dan titik akses menggunakan titik akhir API ini.

Batasan Panjang: 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})$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "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.

[NextToken](#)

NextToken hadir jika payload respons diberi paginasi. Anda dapat menggunakan NextToken permintaan berikutnya untuk mengambil halaman berikutnya dari deskripsi titik akses.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

[Tags](#)

Array tag untuk sumber daya EFS tertentu.

Tipe: Array objek [Tag](#)

Kesalahan

AccessPointNotFound

Dikembalikan jika AccessPointId nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

ModifyMountTargetSecurityGroups

Memodifikasi kumpulan grup keamanan yang berlaku untuk target mount.

Saat Anda membuat target pemasangan, Amazon EFS juga membuat antarmuka jaringan baru. Untuk informasi selengkapnya, lihat [CreateMountTarget](#). Operasi ini menggantikan grup keamanan yang berlaku untuk antarmuka jaringan yang terkait dengan target pemasangan, dengan yang SecurityGroups disediakan dalam permintaan. Operasi ini mengharuskan antarmuka jaringan target pemasangan telah dibuat dan status siklus hidup target pemasangan tidak `deleted`.

Operasi memerlukan izin untuk tindakan berikut:

- `elasticfilesystem:ModifyMountTargetSecurityGroupstindakan` pada sistem file target mount.
- `ec2:ModifyNetworkInterfaceAttributeaksi` pada antarmuka jaringan target mount.

Minta Sintaks

```
PUT /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
Content-type: application/json

{
  "SecurityGroups": [ "string" ]
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[MountTargetId](#)

ID target mount yang grup keamanannya ingin Anda ubah.

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsmt-[0-9a-f]{8,40}$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

[SecurityGroups](#)

Array grup keamanan VPC. IDs

Tipe: Array string

Anggota Array: Jumlah maksimum 100 item.

Kendala Panjang: Panjang minimum 11. Panjang maksimum 43.

Pola: `^sg-[0-9a-f]{8,40}`

Diperlukan: Tidak

Sintaksis Respons

```
HTTP/1.1 204
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan isi HTTP kosong.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

`IncorrectMountTargetState`

Dikembalikan jika target pemasangan tidak dalam keadaan yang benar untuk operasi.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

`InternalServerError`

Dikembalikan jika terjadi kesalahan di sisi server.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

SecurityGroupLimitExceeded

Dikembalikan jika jumlah SecurityGroups yang ditentukan dalam permintaan lebih besar dari batas, yang didasarkan pada kuota akun. Hapus beberapa grup keamanan atau minta kuota akun dinaikkan. Untuk informasi selengkapnya, lihat [Kuota VPC](#) Amazon di Panduan Pengguna Amazon VPC (lihat tabel Grup Keamanan).

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

SecurityGroupNotFound

Dikembalikan jika salah satu grup keamanan yang ditentukan tidak ada di virtual private cloud (VPC) subnet.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Contoh

Ganti grup keamanan target mount

Contoh berikut menggantikan grup keamanan yang berlaku untuk antarmuka jaringan yang terkait dengan target mount.

Permintaan Sampel

```
PUT /2015-02-01/mount-targets/fsmt-9a13661e/security-groups HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T223446Z
Authorization: <...>
```

```
Content-Type: application/json
```

```
Content-Length: 57
```

```
{
  "SecurityGroups" : [
    "sg-188d9f74"
  ]
}
```

Contoh Respons

```
HTTP/1.1 204 No Content
```

```
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

PutAccountPreferences

Gunakan operasi ini untuk mengatur preferensi akun saat ini AWS Region untuk menggunakan sumber daya panjang 17 karakter (63 bit) atau 8 karakter pendek (32 bit) IDs untuk sistem file EFS baru dan memasang sumber daya target. Semua sumber daya yang IDs ada tidak terpengaruh oleh perubahan apa pun yang Anda buat. Anda dapat menyetel preferensi ID selama periode keikutsertaan saat EFS beralih ke sumber daya yang panjang. IDs Untuk informasi selengkapnya, lihat [Mengelola sumber daya Amazon EFS IDs](#).

Note

Mulai Oktober 2021, Anda akan menerima kesalahan jika Anda mencoba mengatur preferensi akun untuk menggunakan ID sumber daya format 8 karakter pendek. Hubungi AWS dukungan jika Anda menerima kesalahan dan harus menggunakan IDs kependekan dari sistem file dan memasang sumber daya target.

Minta Sintaks

```
PUT /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json
```

```
{
  "ResourceIdType": "string"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

[ResourceIdType](#)

Menentukan preferensi ID sumber daya EFS yang akan disetel untuk pengguna Akun AWS, saat ini AWS Region, baik LONG_ID (17 karakter), atau SHORT_ID (8 karakter).

 Note

Mulai Oktober 2021, Anda akan menerima kesalahan saat menyetel preferensi akun keSHORT_ID. Hubungi AWS dukungan jika Anda menerima kesalahan dan harus menggunakan IDs kependekan dari sistem file dan memasang sumber daya target.

Tipe: String

Nilai yang Valid: LONG_ID | SHORT_ID

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "ResourceIdPreference": {
    "ResourceIdType": "string",
    "Resources": [ "string" ]
  }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ResourceIdPreference

Menjelaskan jenis sumber daya dan preferensi ID-nya untuk pengguna Akun AWS, saat ini AWS Region.

Tipe: Objek ResourceIdPreference

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

PutBackupPolicy

Memperbarui kebijakan cadangan sistem file. Gunakan tindakan ini untuk memulai atau menghentikan pencadangan otomatis sistem file.

Minta Sintaks

```
PUT /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
Content-type: application/json

{
  "BackupPolicy": {
    "Status": "string"
  }
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

Menentukan sistem file EFS mana yang akan memperbarui kebijakan pencadangan.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

BackupPolicy

Kebijakan cadangan termasuk dalam PutBackupPolicy permintaan.

Tipe: Objek BackupPolicy

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "BackupPolicy": {
    "Status": "string"
  }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[BackupPolicy](#)

Menjelaskan kebijakan pencadangan sistem file, yang menunjukkan apakah pencadangan otomatis diaktifkan atau dimatikan.

Tipe: Objek [BackupPolicy](#)

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

`FileSystemNotFound`

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

`IncorrectFileSystemLifecycleState`

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

`ErrorCode`

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

`Message`

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia AWS Region di mana permintaan dibuat.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

PutFileSystemPolicy

Menerapkan Amazon EFS FileSystemPolicy ke sistem file Amazon EFS. Kebijakan sistem file adalah kebijakan berbasis sumber daya IAM dan dapat berisi beberapa pernyataan kebijakan. Sistem file selalu memiliki persis satu kebijakan sistem file, yang dapat berupa kebijakan default atau kumpulan kebijakan eksplisit atau diperbarui menggunakan operasi API ini. Kebijakan sistem file EFS memiliki batas 20.000 karakter. Ketika kebijakan eksplisit disetel, kebijakan tersebut akan mengesampingkan kebijakan default. Untuk informasi selengkapnya tentang kebijakan sistem file default, lihat [Kebijakan sistem file EFS default](#).

Note

Kebijakan sistem file EFS memiliki batas 20.000 karakter.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:PutFileSystemPolicy`.

Minta Sintaks

```
PUT /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
Content-type: application/json

{
  "BypassPolicyLockoutSafetyCheck": boolean,
  "Policy": "string"
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[FileSystemId](#)

ID sistem file EFS yang ingin Anda buat atau perbarui FileSystemPolicy untuk.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

BypassPolicyLockoutSafetyCheck

(Opsional) Boolean yang menentukan apakah akan melewati pemeriksaan keamanan FileSystemPolicy lockout atau tidak. Pemeriksaan keamanan penguncian menentukan apakah kebijakan dalam permintaan akan mengunci, atau mencegah, prinsipal IAM yang membuat permintaan dari membuat PutFileSystemPolicy permintaan future pada sistem file ini. Setel BypassPolicyLockoutSafetyCheck ke True hanya ketika Anda bermaksud untuk mencegah prinsipal IAM yang membuat permintaan dari membuat PutFileSystemPolicy permintaan berikutnya pada sistem file ini. Nilai default-nya adalah False.

Tipe: Boolean

Wajib: Tidak

Policy

FileSystemPolicyYang Anda ciptakan. Menerima definisi kebijakan berformat JSON. Kebijakan sistem file EFS memiliki batas 20.000 karakter. Untuk mengetahui selengkapnya tentang elemen yang membentuk kebijakan sistem file, lihat [Kebijakan berbasis sumber daya dalam Amazon EFS](#).

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 20000.

Pola: `[\s\S]+`

Diperlukan: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "FileSystemId": "string",
  "Policy": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

FileSystemId

Menentukan sistem file EFS yang FileSystemPolicy berlaku.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Policy

JSON diformat FileSystemPolicy untuk sistem file EFS.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 20000.

Pola: `[\s\S]+`

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada

pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

InvalidPolicyException

Dikembalikan jika cacat atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter yang diperlukan hilang. `FileSystemPolicy` Dikembalikan jika terjadi kesalahan pemeriksaan keamanan penguncian kebijakan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan

yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

Contoh

Buat EFS FileSystemPolicy

Permintaan berikut membuat sebuah FileSystemPolicy yang memungkinkan semua AWS prinsipal untuk me-mount sistem file EFS yang ditentukan dengan izin baca dan tulis.

Permintaan Sampel

```
PUT /2015-02-01/file-systems/fs-01234567/file-system-policy HTTP/1.1
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite"
      ],
      "Principal": {
        "AWS": ["*"]
      },
    }
  ]
}
```

Contoh Respons

```
{
  "Version": "2012-10-17",
  "Id": "1",
  "Statement": [
    {
      "Sid": "efs-statement-abcdef01-1111-bbbb-2222-111122224444",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientMount",

```

```
        "elasticfilesystem:ClientWrite"
    ],
    "Principal": {
        "AWS": ["*"]
    },
    "Resource": "arn:aws:elasticfilesystem:us-east-1:1111222233334444:file-
system/fs-01234567"
    }
]
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

PutLifecycleConfiguration

Gunakan tindakan ini untuk mengelola penyimpanan untuk sistem file Anda. A LifecycleConfiguration terdiri dari satu atau lebih LifecyclePolicy objek yang mendefinisikan sebagai berikut:

- **TransitionToIA**— Kapan harus memindahkan file dalam sistem file dari penyimpanan utama (kelas penyimpanan standar) ke penyimpanan Akses Jarang (IA).
- **TransitionToArchive**— Kapan harus memindahkan file dalam sistem file dari kelas penyimpanan mereka saat ini (baik IA atau penyimpanan Standar) ke penyimpanan Arsip.

Sistem file tidak dapat bertransisi ke penyimpanan Arsip sebelum beralih ke penyimpanan IA. Oleh karena itu, TransitionToArchive tidak boleh diatur atau harus lebih lambat dari TransitionTo IA.

Note

Kelas penyimpanan Arsip hanya tersedia untuk sistem file yang menggunakan mode throughput Elastis dan mode kinerja Tujuan Umum.

- **TransitionToPrimaryStorageClass**— Apakah akan memindahkan file dalam sistem file kembali ke penyimpanan utama (kelas penyimpanan standar) setelah diakses di penyimpanan IA atau Arsip.

Untuk informasi selengkapnya, lihat [Mengelola penyimpanan sistem file](#).

Setiap sistem file Amazon EFS mendukung satu konfigurasi siklus hidup, yang berlaku untuk semua file dalam sistem file. Jika LifecycleConfiguration objek sudah ada untuk sistem file yang ditentukan, PutLifecycleConfiguration panggilan memodifikasi konfigurasi yang ada. PutLifecycleConfigurationPanggilan dengan LifecyclePolicies array kosong di badan permintaan menghapus semua yang ada LifecycleConfiguration. Dalam permintaan, tentukan yang berikut ini:

- ID untuk sistem file yang Anda aktifkan, nonaktifkan, atau modifikasi manajemen siklus hidup.
- LifecyclePoliciesArray LifecyclePolicy objek yang menentukan kapan harus memindahkan file ke penyimpanan IA, ke penyimpanan Arsip, dan kembali ke penyimpanan utama.

 Note

Amazon EFS mengharuskan setiap LifecyclePolicy objek hanya memiliki satu transisi, sehingga LifecyclePolicies array perlu terstruktur dengan LifecyclePolicy objek terpisah. Lihat contoh permintaan di bagian berikut untuk informasi selengkapnya.

Operasi ini memerlukan izin untuk `elasticfilesystem:PutLifecycleConfiguration` operasi.

Untuk menerapkan LifecycleConfiguration objek ke sistem file terenkripsi, Anda memerlukan AWS Key Management Service izin yang sama seperti ketika Anda membuat sistem file terenkripsi.

Minta Sintaks

```
PUT /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
Content-type: application/json
```

```
{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "string",
      "TransitionToIA": "string",
      "TransitionToPrimaryStorageClass": "string"
    }
  ]
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

ID dari sistem file tempat Anda membuat LifecycleConfiguration objek (String).

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

[LifecyclePolicies](#)

Sebuah array `LifecyclePolicy` objek yang mendefinisikan `LifecycleConfiguration` objek sistem file. `LifecycleConfiguration` objek menginformasikan manajemen siklus hidup berikut ini:

- **TransitionToIA**— Kapan harus memindahkan file dalam sistem file dari penyimpanan utama (kelas penyimpanan standar) ke penyimpanan Akses Jarang (IA).
- **TransitionToArchive**— Kapan harus memindahkan file dalam sistem file dari kelas penyimpanan mereka saat ini (baik IA atau penyimpanan Standar) ke penyimpanan Arsip.

Sistem file tidak dapat bertransisi ke penyimpanan Arsip sebelum beralih ke penyimpanan IA. Oleh karena itu, `TransitionToArchive` tidak boleh diatur atau harus lebih lambat dari `TransitionTo IA`.

Note

Kelas penyimpanan Arsip hanya tersedia untuk sistem file yang menggunakan mode throughput Elastis dan mode kinerja Tujuan Umum.

- **TransitionToPrimaryStorageClass**— Apakah akan memindahkan file dalam sistem file kembali ke penyimpanan utama (kelas penyimpanan standar) setelah diakses di penyimpanan IA atau Arsip.

Note

Saat menggunakan perintah `put-lifecycle-configuration` CLI atau tindakan `PutLifecycleConfiguration` API, Amazon EFS mengharuskan setiap `LifecyclePolicy` objek hanya memiliki satu transisi. Ini berarti bahwa dalam badan permintaan, `LifecyclePolicies` harus terstruktur sebagai array `LifecyclePolicy` objek, satu objek untuk setiap transisi penyimpanan. Lihat contoh permintaan di bagian berikut untuk informasi selengkapnya.

Tipe: Array objek [LifecyclePolicy](#)

Anggota Array: Jumlah maksimum 3 item.

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "string",
      "TransitionToIA": "string",
      "TransitionToPrimaryStorageClass": "string"
    }
  ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[LifecyclePolicies](#)

Array kebijakan manajemen siklus hidup. EFS mendukung maksimal satu kebijakan per sistem file.

Tipe: Array objek [LifecyclePolicy](#)

Anggota Array: Jumlah maksimum 3 item.

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak "tersedia".

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Contoh

Buat konfigurasi siklus hidup

Contoh berikut membuat LifecyclePolicy objek menggunakan PutLifecycleConfiguration tindakan. Contoh ini membuat kebijakan siklus hidup yang menginstruksikan EFS untuk melakukan hal berikut:

- Pindahkan semua file dalam sistem file yang belum diakses di penyimpanan Standar dalam 30 hari terakhir ke penyimpanan IA.
- Pindahkan semua file dalam sistem file yang belum diakses di penyimpanan Standar dalam 90 hari terakhir ke penyimpanan Arsip.
- Pindahkan file kembali ke penyimpanan Standar setelah diakses di penyimpanan IA atau Arsip. Kelas penyimpanan Arsip hanya tersedia untuk sistem file yang menggunakan mode throughput Elastis dan mode kinerja Tujuan Umum.

Untuk informasi selengkapnya, lihat [kelas penyimpanan EFS](#) dan [Mengelola penyimpanan sistem file](#).

Permintaan Sampel

```
PUT /2015-02-01/file-systems/fs-0123456789abcdefb/lifecycle-configuration HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20181122T232908Z
Authorization: <...>
Content-type: application/json
Content-Length: 86

{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "AFTER_90_DAYS"
    },
    {
      "TransitionToIA": "AFTER_30_DAYS"
    },
    {
      "TransitionToPrimaryStorage": "AFTER_1_ACCESS"
    }
  ]
}
```

```
}
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-type: application/json
Content-Length: 86

{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "AFTER_90_DAYS"
    },
    {
      "TransitionToIA": "AFTER_30_DAYS"
    },
    {
      "TransitionToPrimaryStorage": "AFTER_1_ACCESS"
    }
  ]
}
```

Contoh Permintaan CLI put-lifecycle-configuration

Contoh ini menggambarkan salah satu penggunaan. PutLifecycleConfiguration

Permintaan Sampel

```
aws efs put-lifecycle-configuration \
  --file-system-id fs-0123456789abcdefb \
  --lifecycle-policies "[{"TransitionToArchive":"AFTER_90_DAYS"},
  {"TransitionToIA":"AFTER_30_DAYS"},
  {"TransitionToPrimaryStorageClass":"AFTER_1_ACCESS"}]"
--region us-west-2 \
--profile adminuser
```

Contoh Respons

```
{
  "LifecyclePolicies": [
```

```
{
  "TransitionToArchive": "AFTER_90_DAYS"
},
{
  "TransitionToIA": "AFTER_30_DAYS"
},
{
  "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
}
]
```

Nonaktifkan manajemen siklus hidup

Contoh berikut menonaktifkan manajemen siklus hidup untuk sistem file yang ditentukan.

Permintaan Sampel

```
PUT /2015-02-01/file-systems/fs-01234567/lifecycle-configuration HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20181122T232908Z
Authorization: <...>
Content-type: application/json
Content-Length: 86

{
  "LifecyclePolicies": [ ]
}
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-type: application/json
Content-Length: 86

{
  "LifecyclePolicies": [ ]
}
```


Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

TagResource

Membuat tag untuk sumber daya EFS. Anda dapat membuat tag untuk sistem file EFS dan titik akses menggunakan operasi API ini.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:TagResource`.

Minta Sintaks

```
POST /2015-02-01/resource-tags/ResourceId HTTP/1.1
Content-type: application/json
```

```
{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

ResourceId

ID yang menentukan sumber daya EFS yang ingin Anda buat tag.

Batasan Panjang: 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})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

Tags

Array Tag objek untuk ditambahkan. Setiap Tag objek adalah pasangan kunci-nilai.

Tipe: Array objek [Tag](#)

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Contoh

Buat Tag pada Sistem File

Permintaan berikut membuat tiga tag ("key1", "key2", dan "key3") pada sistem file yang ditentukan.

Permintaan Sampel

```
POST /2015-02-01/tag-resource/fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160
```

```
{
  "Tags": [
    {
      "Key": "key1",
      "Value": "value1"
    },
    {
      "Key": "key2",
      "Value": "value2"
```

```
    },  
    {  
      "Key": "key3",  
      "Value": "value3"  
    }  
  ]  
}
```

Contoh Respons

```
HTTP/1.1 204 no content  
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

UntagResource

Menghapus tag dari sumber daya EFS. Anda dapat menghapus tag dari sistem file EFS dan titik akses menggunakan operasi API ini.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:UntagResource`.

Minta Sintaks

```
DELETE /2015-02-01/resource-tags/ResourceId?tagKeys=TagKeys HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[ResourceId](#)

Menentukan sumber daya EFS yang ingin Anda hapus tag dari.

Batasan Panjang: 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})$`

Wajib: Ya

[TagKeys](#)

Kunci pasangan tag nilai kunci yang ingin Anda hapus dari sumber daya EFS yang ditentukan.

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `^(?![aA]{1}[wW]{1}[sS]{1}:)([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]+)$`

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS
ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

UpdateFileSystem

Memperbarui mode throughput atau jumlah throughput yang disediakan dari sistem file yang ada.

Minta Sintaks

```
PUT /2015-02-01/file-systems/FileSystemId HTTP/1.1
Content-type: application/json

{
  "ProvisionedThroughputInMibps": number,
  "ThroughputMode": "string"
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

ID sistem file yang ingin Anda perbarui.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

ProvisionedThroughputInMibps

(Opsional) Throughput, diukur dalam mebibytes per detik (MiBps), yang ingin Anda sediakan untuk sistem file yang Anda buat. Harus diisi jika ThroughputMode diatur ke provisioned. Nilai yang valid adalah 1-3414 MiBps, dengan batas atas tergantung pada Wilayah. Untuk meningkatkan batas ini, hubungi Dukungan. Untuk informasi selengkapnya, lihat [Kuota Amazon EFS yang dapat Anda tingkatkan](#) di Panduan Pengguna Amazon EFS.

Tipe: Ganda

Rentang Valid: Nilai minimum 1.0.

Wajib: Tidak

ThroughputMode

(Opsional) Memperbarui mode throughput sistem file. Jika Anda tidak memperbarui mode throughput Anda, Anda tidak perlu memberikan nilai ini dalam permintaan Anda. Jika Anda mengubah ThroughputMode keprovisioned, Anda juga harus menetapkan nilai untukProvisionedThroughputInMibps.

Tipe: String

Nilai yang Valid: bursting | provisioned | elastic

Wajib: Tidak

Sintaksis Respons

HTTP/1.1 202

Content-type: application/json

```
{
  "AvailabilityZoneId": "string",
  "AvailabilityZoneName": "string",
  "CreationTime": number,
  "CreationToken": "string",
  "Encrypted": boolean,
  "FileSystemArn": "string",
  "FileSystemId": "string",
  "FileSystemProtection": {
    "ReplicationOverwriteProtection": "string"
  },
  "KmsKeyId": "string",
  "LifeCycleState": "string",
  "Name": "string",
  "NumberOfMountTargets": number,
  "OwnerId": "string",
  "PerformanceMode": "string",
  "ProvisionedThroughputInMibps": number,
  "SizeInBytes": {
```

```
    "Timestamp": number,
    "Value": number,
    "ValueInArchive": number,
    "ValueInIA": number,
    "ValueInStandard": number
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "ThroughputMode": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 202.

Layanan mengembalikan data berikut dalam format JSON.

AvailabilityZoneId

Pengidentifikasi unik dan konsisten dari Availability Zone di mana sistem file berada, dan hanya berlaku untuk sistem file One Zone. Misalnya, use1-az1 adalah ID Availability Zone untuk AWS Region us-east-1, dan memiliki lokasi yang sama di setiap Akun AWS

Tipe: String

AvailabilityZoneName

Menjelaskan AWS Availability Zone di mana sistem file berada, dan hanya berlaku untuk sistem file One Zone. Untuk informasi selengkapnya, lihat [Menggunakan kelas penyimpanan EFS](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

CreationTime

Waktu sistem file dibuat, dalam hitungan detik (sejak 1970-01-01T 00:00:00 Z).

Tipe: Timestamp

CreationToken

String buram ditentukan dalam permintaan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

Encrypted

Nilai Boolean yang, jika benar, menunjukkan bahwa sistem file dienkrpsi.

Jenis: Boolean

FileSystemArn

Nama Sumber Daya Amazon (ARN) untuk sistem file EFS, dalam format.

`arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id` Contoh dengan data sampel: `arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567`

Tipe: String

FileSystemId

ID sistem file, yang ditetapkan oleh Amazon EFS.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

FileSystemProtection

Menjelaskan perlindungan pada sistem file.

Tipe: Objek [FileSystemProtectionDescription](#)

KmsKeyId

ID yang AWS KMS key digunakan untuk melindungi sistem file terenkrpsi.

Tipe: String

Batasan Panjang: Panjang maksimum 2048.

Pola: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

LifeCycleState

Fase siklus hidup dari sistem file.

Tipe: String

Nilai yang Valid: `creating | available | updating | deleting | deleted | error`

Name

Anda dapat menambahkan tag ke sistem file, termasuk Name tag. Untuk informasi selengkapnya, lihat [CreateFileSystem](#). Jika sistem file memiliki Name tag, Amazon EFS mengembalikan nilai di bidang ini.

Tipe: String

Batasan Panjang: Panjang maksimum 256.

Pola: `^([\p{L}\p{Z}\p{N}_.:/+@-]*)$`

NumberOfMountTargets

Jumlah target mount saat ini yang dimiliki sistem file. Untuk informasi selengkapnya, lihat [CreateMountTarget](#).

Jenis: Integer

Rentang yang Valid: Nilai minimum 0.

OwnerId

Akun AWS Yang menciptakan sistem file.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

[PerformanceMode](#)

Mode performa sistem file.

Tipe: String

Nilai yang Valid: `generalPurpose` | `maxIO`

[ProvisionedThroughputInMibps](#)

Jumlah throughput yang disediakan, diukur dalam MiBps, untuk sistem file. Berlaku untuk sistem file menggunakan `ThroughputMode` set `toprovisioned`.

Tipe: Ganda

Rentang Valid: Nilai minimum 1.0.

[SizeInBytes](#)

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan dalam sistem file, di `Value` bidangnya, dan waktu di mana ukuran itu ditentukan di `Timestamp`.

`TimestampNilainya` adalah bilangan bulat detik sejak 1970-01-01T 00:00:00 Z.

`SizeInBytesNilai` tidak mewakili ukuran snapshot yang konsisten dari sistem file, tetapi pada akhirnya konsisten ketika tidak ada penulisan ke sistem file. Artinya, `SizeInBytes` mewakili ukuran sebenarnya hanya jika sistem file tidak dimodifikasi untuk jangka waktu lebih dari beberapa jam. Jika tidak, nilainya bukan ukuran yang tepat dari sistem file pada setiap titik waktu.

Tipe: Objek [FileSystemSize](#)

[Tags](#)

Tag yang terkait dengan sistem file, disajikan sebagai array Tag objek.

Tipe: Array objek [Tag](#)

[ThroughputMode](#)

Menampilkan mode throughput sistem file. Untuk informasi selengkapnya, lihat [Mode throughput](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Nilai yang Valid: `bursting` | `provisioned` | `elastic`

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifeCycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InsufficientThroughputCapacity

Dikembalikan jika tidak ada kapasitas yang cukup untuk menyediakan throughput tambahan. Nilai ini mungkin dikembalikan saat Anda mencoba membuat sistem file dalam mode throughput yang disediakan, saat Anda mencoba meningkatkan throughput yang disediakan dari sistem file yang ada, atau saat Anda mencoba mengubah sistem file yang ada dari Bursting Throughput ke mode Throughput Terprovisioned. Coba lagi nanti.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 503

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 telah tercapai. MiB/s

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

TooManyRequests

Dikembalikan jika Anda tidak menunggu setidaknya 24 jam sebelum mengubah mode throughput, atau mengurangi nilai Throughput yang Disediakan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 429

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

UpdateFileSystemProtection

Memperbarui perlindungan pada sistem file.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:UpdateFileSystemProtection`.

Minta Sintaks

```
PUT /2015-02-01/file-systems/FileSystemId/protection HTTP/1.1
Content-type: application/json

{
  "ReplicationOverwriteProtection": "string"
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

FileSystemId

ID sistem file yang akan diperbarui.

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

ReplicationOverwriteProtection

Status perlindungan penyimpanan replikasi sistem file.

- **ENABLED**— Sistem file tidak dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file dapat ditulis. Perlindungan penyimpanan replikasi secara default **ENABLED**.

- **DISABLED**— Sistem file dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dapat dimodifikasi oleh replikasi EFS.
- **REPLICATING**— Sistem file digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dimodifikasi hanya oleh replikasi EFS.

Jika konfigurasi replikasi dihapus, perlindungan penimpanan replikasi sistem file diaktifkan kembali dan sistem file menjadi dapat ditulis.

Tipe: String

Nilai yang Valid: ENABLED | DISABLED | REPLICATING

Wajib: Tidak

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "ReplicationOverwriteProtection": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[ReplicationOverwriteProtection](#)

Status perlindungan penimpanan replikasi sistem file.

- **ENABLED**— Sistem file tidak dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file dapat ditulis. Perlindungan penimpanan replikasi secara default **ENABLED**.
- **DISABLED**— Sistem file dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dapat dimodifikasi oleh replikasi EFS.
- **REPLICATING**— Sistem file digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dimodifikasi oleh replikasi EFS.

Jika konfigurasi replikasi dihapus, perlindungan penyimpanan replikasi sistem file diaktifkan kembali, sistem file menjadi dapat ditulis.

Tipe: String

Nilai yang Valid: ENABLED | DISABLED | REPLICATING

Kesalahan

BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 404

IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

InsufficientThroughputCapacity

Dikembalikan jika tidak ada kapasitas yang cukup untuk menyediakan throughput tambahan. Nilai ini mungkin dikembalikan saat Anda mencoba membuat sistem file dalam mode throughput yang disediakan, saat Anda mencoba meningkatkan throughput yang disediakan dari sistem file yang ada, atau saat Anda mencoba mengubah sistem file yang ada dari Bursting Throughput ke mode Throughput Terprovisioned. Coba lagi nanti.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 503

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 500

ReplicationAlreadyExists

Dikembalikan jika sistem file sudah termasuk dalam konfigurasi replikasi. >

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 409

ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 telah tercapai. MiB/s

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 400

TooManyRequests

Dikembalikan jika Anda tidak menunggu setidaknya 24 jam sebelum mengubah mode throughput, atau mengurangi nilai Throughput yang Disediakan.

ErrorCode

Kode kesalahan adalah string yang secara unik mengidentifikasi kondisi kesalahan. Ini dimaksudkan untuk dibaca dan dipahami oleh program yang mendeteksi dan menangani kesalahan berdasarkan tipe.

Message

Pesan kesalahan berisi deskripsi umum kondisi kesalahan dalam bahasa Inggris. Ini ditujukan untuk audiens manusia. Program sederhana menampilkan pesan secara langsung kepada pengguna akhir jika mereka mengalami kondisi kesalahan yang tidak mereka ketahui atau tidak peduli untuk menanganinya. Program-program canggih dengan penanganan kesalahan yang lebih lengkap dan internasionalisasi yang tepat lebih cenderung mengabaikan pesan kesalahan.

Kode Status HTTP: 429

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Tipe Data

tipe data berikut didukung:

- [AccessPointDescription](#)
- [BackupPolicy](#)
- [CreationInfo](#)
- [Destination](#)
- [DestinationToCreate](#)
- [FileSystemDescription](#)
- [FileSystemProtectionDescription](#)
- [FileSystemSize](#)
- [LifecyclePolicy](#)
- [MountTargetDescription](#)
- [PosixUser](#)
- [ReplicationConfigurationDescription](#)
- [ResourceIdPreference](#)
- [RootDirectory](#)
- [Tag](#)

AccessPointDescription

Memberikan deskripsi titik akses sistem file EFS.

Daftar Isi

AccessPointArn

Nama Sumber Daya Amazon (ARN) unik yang terkait dengan titik akses.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}$`

Wajib: Tidak

AccessPointId

ID titik akses, yang ditetapkan oleh Amazon EFS.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

Wajib: Tidak

ClientToken

String buram ditentukan dalam permintaan untuk memastikan pembuatan yang idempotensi.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

Wajib: Tidak

FileSystemId

ID dari sistem file EFS di mana titik akses berlaku.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Tidak

LifeCycleState

Mengidentifikasi fase siklus hidup dari titik akses.

Tipe: String

Nilai yang Valid: `creating | available | updating | deleting | deleted | error`

Wajib: Tidak

Name

Nama titik akses. Ini adalah nilai Name tag.

Tipe: String

Wajib: Tidak

OwnerId

Mengidentifikasi Akun AWS yang memiliki sumber daya titik akses.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

Wajib: Tidak

PosixUser

Identitas POSIX lengkap, termasuk ID pengguna, ID grup, dan grup sekunder IDs pada titik akses yang digunakan untuk semua operasi file oleh klien NFS menggunakan titik akses.

Tipe: Objek [PosixUser](#)

Wajib: Tidak

RootDirectory

Direktori pada sistem file EFS yang diekspos oleh titik akses sebagai direktori root ke klien NFS menggunakan titik akses.

Tipe: Objek [RootDirectory](#)

Wajib: Tidak

Tags

Tag yang terkait dengan titik akses, disajikan sebagai array objek Tag.

Tipe: Array objek [Tag](#)

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BackupPolicy

Kebijakan cadangan untuk sistem file yang digunakan untuk membuat backup harian otomatis. Jika status memiliki nilai `ENABLED`, sistem file sedang dicadangkan secara otomatis. Untuk informasi selengkapnya, lihat [Pencadangan otomatis](#).

Daftar Isi

Status

Menjelaskan status kebijakan cadangan sistem file.

- **ENABLED**— EFS secara otomatis mencadangkan sistem file.
- **ENABLING**— EFS menyalakan backup otomatis untuk sistem file.
- **DISABLED**— Cadangan otomatis dimatikan untuk sistem file.
- **DISABLING**— EFS mematikan backup otomatis untuk sistem file.

Tipe: String

Nilai yang Valid: `ENABLED` | `ENABLING` | `DISABLED` | `DISABLING`

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CreationInfo

Diperlukan jika `RootDirectory > Path` yang ditentukan tidak ada. Menentukan POSIX IDs dan izin untuk diterapkan ke titik akses `>`. `RootDirectory Path` Jika direktori root titik akses tidak ada, EFS membuatnya dengan pengaturan ini saat klien terhubung ke titik akses. Ketika menentukan `CreationInfo`, Anda harus menyertakan nilai untuk semua properti.

Amazon EFS membuat direktori root hanya jika Anda telah menyediakan `CreationInfo`: `OwnUid`, `ownGid`, dan izin untuk direktori tersebut. Jika Anda tidak memberikan informasi ini, Amazon EFS tidak membuat direktori root. Jika direktori root tidak ada, upaya untuk memasang menggunakan titik akses akan gagal.

Important

Jika Anda tidak memberikan `CreationInfo` dan `RootDirectory` yang ditentukan tidak ada, upaya untuk memasang sistem file menggunakan titik akses akan gagal.

Daftar Isi

OwnerGid

Menentukan ID grup POSIX untuk diterapkan ke `RootDirectory`. Menerima nilai dari 0 hingga 2^{32} (4294967295).

Tipe: Long

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Ya

OwnerUid

Menentukan ID pengguna POSIX untuk diterapkan ke `RootDirectory`. Menerima nilai dari 0 hingga 2^{32} (4294967295).

Tipe: Long

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Ya

Permissions

Menentukan izin POSIX untuk diterapkan ke `RootDirectory`, dalam format angka oktal yang mewakili bit mode file.

Tipe: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 4.

Pola: `^[0-7]{3,4}$`

Diperlukan: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Destination

Menjelaskan sistem file tujuan dalam konfigurasi replikasi.

Daftar Isi

FileSystemId

ID dari sistem file Amazon EFS tujuan.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

Region

AWS Region Di mana sistem file tujuan berada.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

Wajib: Ya

Status

Menjelaskan status konfigurasi replikasi. Untuk informasi selengkapnya tentang status replikasi, lihat [Melihat detail replikasi](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Nilai yang Valid: ENABLED | ENABLING | DELETING | ERROR | PAUSED | PAUSING

Wajib: Ya

LastReplicatedTimestamp

Waktu ketika sinkronisasi terbaru berhasil diselesaikan pada sistem file tujuan. Setiap perubahan data pada sistem file sumber yang terjadi sebelum waktu ini telah berhasil direplikasi ke sistem file tujuan. Setiap perubahan yang terjadi setelah waktu ini mungkin tidak sepenuhnya direplikasi.

Tipe: Timestamp

Wajib: Tidak

OwnerId

ID Akun AWS di mana sistem file tujuan berada.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

Wajib: Tidak

RoleArn

Nama Sumber Daya Amazon (ARN) dari peran IAM di akun sumber yang memungkinkan Amazon EFS melakukan replikasi atas namanya. Ini opsional untuk replikasi akun yang sama dan diperlukan untuk replikasi lintas akun.

Tipe: String

Batasan Panjang: Panjang maksimum 2048.

Pola: `arn:(aws[a-zA-Z-]*)?:iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

Wajib: Tidak

StatusMessage

Pesan yang memberikan rincian tentang PAUSED atau ERROR status konfigurasi tujuan replikasi. Untuk informasi selengkapnya tentang pesan status replikasi, lihat [Melihat detail replikasi](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DestinationToCreate

Menjelaskan sistem file tujuan baru atau yang sudah ada untuk konfigurasi replikasi.

- Jika Anda ingin mereplikasi ke sistem file baru, jangan tentukan ID Sistem File untuk sistem file tujuan. Amazon EFS menciptakan sistem file baru yang kosong. Untuk penyimpanan One Zone, tentukan Availability Zone untuk membuat sistem file. Untuk menggunakan AWS Key Management Service kunci selain kunci KMS default, kemudian tentukan itu. Untuk informasi selengkapnya, lihat [Mengonfigurasi replikasi ke sistem file Amazon EFS baru](#) di Panduan Pengguna Amazon EFS.

Note

Setelah sistem file dibuat, Anda tidak dapat mengubah kunci KMS atau mode kinerja.

- Jika Anda ingin mereplikasi ke sistem file yang ada di akun yang sama dengan sistem file sumber, maka Anda perlu memberikan ID atau Nama Sumber Daya Amazon (ARN) dari sistem file yang akan direplikasi. Perlindungan penimpanan replikasi sistem file harus dinonaktifkan. Untuk informasi selengkapnya, lihat [Mereplikasi ke sistem file yang ada](#) di Panduan Pengguna Amazon EFS.
- Jika Anda mereplikasi sistem file ke sistem file yang berada di akun yang berbeda dari sistem file sumber (replikasi lintas akun), Anda perlu menyediakan ARN untuk sistem file dan peran IAM yang memungkinkan Amazon EFS melakukan replikasi pada akun tujuan. Perlindungan penimpanan replikasi sistem file harus dinonaktifkan. Untuk informasi selengkapnya, lihat [Mereplikasi Akun AWS](#) di Panduan Pengguna Amazon EFS.

Daftar Isi

AvailabilityZoneName

Untuk membuat sistem file yang menggunakan penyimpanan One Zone, tentukan nama Availability Zone untuk membuat sistem file tujuan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: . +

Wajib: Tidak

FileSystemId

ID atau ARN dari sistem file yang akan digunakan untuk tujuan. Untuk replikasi lintas akun, ini harus ARN. Replikasi tanpa replikasi sistem file harus dinonaktifkan. Jika tidak ada ID atau ARN yang ditentukan, maka sistem file baru dibuat.

Note

Saat pertama kali mengonfigurasi replikasi ke sistem file yang ada, Amazon EFS menulis data ke atau menghapus data yang ada dari sistem file tujuan agar sesuai dengan data dalam sistem file sumber. Jika Anda tidak ingin mengubah data dalam sistem file tujuan, maka Anda harus mereplikasi ke sistem file baru sebagai gantinya. Untuk informasi selengkapnya, lihat <https://docs.aws.amazon.com/efs/latest/ug/create-replication.html>.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Tidak

KmsKeyId

Tentukan kunci AWS Key Management Service (AWS KMS) yang ingin Anda gunakan untuk mengenkripsi sistem file tujuan. Jika Anda tidak menentukan kunci KMS, Amazon EFS menggunakan kunci KMS default Anda untuk Amazon EFS. /aws/elasticfilesystem ID ini dapat berupa salah satu dari berikut:

- ID Kunci - Pengidentifikasi unik kunci, misalnya `1234abcd-12ab-34cd-56ef-1234567890ab`.
- ARN - ARN untuk kunci, misalnya. `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`
- Alias kunci - Nama tampilan yang dibuat sebelumnya untuk kunci, misalnya `alias/projectKey1`.
- Alias kunci ARN - ARN untuk alias kunci, misalnya. `arn:aws:kms:us-west-2:444455556666:alias/projectKey1`

Tipe: String

Batasan Panjang: Panjang maksimum 2048.

Pola: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

Wajib: Tidak

Region

Untuk membuat sistem file yang menggunakan penyimpanan Regional, tentukan AWS Region tempat untuk membuat sistem file tujuan. Wilayah harus diaktifkan untuk Akun AWS yang memiliki sistem file sumber. Untuk informasi selengkapnya, lihat [Mengelola Wilayah AWS](#) dalam Panduan Referensi Referensi AWS Umum.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

Wajib: Tidak

RoleArn

Nama Sumber Daya Amazon (ARN) dari peran IAM di akun sumber yang memungkinkan Amazon EFS melakukan replikasi atas namanya. Ini opsional untuk replikasi akun yang sama dan diperlukan untuk replikasi lintas akun.

Tipe: String

Batasan Panjang: Panjang maksimum 2048.

Pola: `arn:(aws[a-zA-Z-]*)?:iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FileSystemDescription

Deskripsi sistem file.

Daftar Isi

CreationTime

Waktu sistem file dibuat, dalam hitungan detik (sejak 1970-01-01T 00:00:00 Z).

Tipe: Timestamp

Wajib: Ya

CreationToken

String buram ditentukan dalam permintaan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

Wajib: Ya

FileSystemId

ID sistem file, yang ditetapkan oleh Amazon EFS.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

LifeCycleState

Fase siklus hidup dari sistem file.

Tipe: String

Nilai yang Valid: `creating` | `available` | `updating` | `deleting` | `deleted` | `error`

Wajib: Ya

NumberOfMountTargets

Jumlah target mount saat ini yang dimiliki sistem file. Untuk informasi selengkapnya, lihat [CreateMountTarget](#).

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 0.

Wajib: Ya

OwnerId

Akun AWS Yang menciptakan sistem file.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

Wajib: Ya

PerformanceMode

Mode performa sistem file.

Tipe: String

Nilai yang Valid: `generalPurpose` | `maxIO`

Wajib: Ya

SizeInBytes

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan dalam sistem file, di `Value` bidangnya, dan waktu di mana ukuran itu ditentukan di `Timestamp` bidangnya.

`TimestampNilainya` adalah bilangan bulat detik sejak 1970-01-01T 00:00:00 Z.

`SizeInBytesNilai` tidak mewakili ukuran snapshot yang konsisten dari sistem file, tetapi pada akhirnya konsisten ketika tidak ada penulisan ke sistem file. Artinya, `SizeInBytes` mewakili

ukuran sebenarnya hanya jika sistem file tidak dimodifikasi untuk jangka waktu lebih dari beberapa jam. Jika tidak, nilainya bukan ukuran yang tepat dari sistem file pada setiap titik waktu.

Tipe: Objek [FileSystemSize](#)

Wajib: Ya

Tags

Tag yang terkait dengan sistem file, disajikan sebagai array Tag objek.

Tipe: Array objek [Tag](#)

Wajib: Ya

AvailabilityZoneId

Pengidentifikasi unik dan konsisten dari Availability Zone di mana sistem file berada, dan hanya berlaku untuk sistem file One Zone. Misalnya, use1-az1 adalah ID Availability Zone untuk AWS Region us-east-1, dan memiliki lokasi yang sama di setiap. Akun AWS

Tipe: String

Wajib: Tidak

AvailabilityZoneName

Menjelaskan AWS Availability Zone di mana sistem file berada, dan hanya berlaku untuk sistem file One Zone. Untuk informasi selengkapnya, lihat [Menggunakan kelas penyimpanan EFS](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: . +

Wajib: Tidak

Encrypted

Nilai Boolean yang, jika benar, menunjukkan bahwa sistem file dienkripsi.

Tipe: Boolean

Wajib: Tidak

FileSystemArn

Nama Sumber Daya Amazon (ARN) untuk sistem file EFS, dalam format.

`arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id` Contoh dengan data sampel: `arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567`

Tipe: String

Wajib: Tidak

FileSystemProtection

Menjelaskan perlindungan pada sistem file.

Tipe: Objek [FileSystemProtectionDescription](#)

Wajib: Tidak

KmsKeyId

ID yang AWS KMS key digunakan untuk melindungi sistem file terenkripsi.

Tipe: String

Batasan Panjang: Panjang maksimum 2048.

Pola: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

Wajib: Tidak

Name

Anda dapat menambahkan tag ke sistem file, termasuk Name tag. Untuk informasi selengkapnya, lihat [CreateFileSystem](#). Jika sistem file memiliki Name tag, Amazon EFS mengembalikan nilai di bidang ini.

Jenis: String

Batasan Panjang: Panjang maksimum 256.

Pola: `^([\p{L}\p{Z}\p{N}_.:/+@-]* )$`

Wajib: Tidak

ProvisionedThroughputInMibps

Jumlah throughput yang disediakan, diukur dalam MiBps, untuk sistem file. Berlaku untuk sistem file menggunakan ThroughputMode set to provisioned.

Tipe: Ganda

Rentang Valid: Nilai minimum 1.0.

Wajib: Tidak

ThroughputMode

Menampilkan mode throughput sistem file. Untuk informasi selengkapnya, lihat [Mode throughput](#) di Panduan Pengguna Amazon EFS.

Tipe: String

Nilai yang Valid: bursting | provisioned | elastic

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FileSystemProtectionDescription

Menjelaskan perlindungan pada sistem file.

Daftar Isi

ReplicationOverwriteProtection

Status perlindungan penyimpanan replikasi sistem file.

- **ENABLED**— Sistem file tidak dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file dapat ditulis. Perlindungan penyimpanan replikasi secara default **ENABLED**.
- **DISABLED**— Sistem file dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dapat dimodifikasi oleh replikasi EFS.
- **REPLICATING**— Sistem file digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dimodifikasi oleh replikasi EFS.

Jika konfigurasi replikasi dihapus, perlindungan penyimpanan replikasi sistem file diaktifkan kembali, sistem file menjadi dapat ditulis.

Tipe: String

Nilai yang Valid: **ENABLED** | **DISABLED** | **REPLICATING**

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FileSystemSize

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan dalam sistem file, di Value bidangnya, dan waktu di mana ukuran itu ditentukan di bidangnyaTimestamp. Nilai tidak mewakili ukuran snapshot yang konsisten dari sistem file, tetapi pada akhirnya konsisten ketika tidak ada penulisan ke sistem file. Artinya, nilai mewakili ukuran sebenarnya hanya jika sistem file tidak dimodifikasi untuk jangka waktu lebih dari beberapa jam. Jika tidak, nilainya belum tentu ukuran yang tepat dari sistem file kapan saja.

Daftar Isi

Value

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan dalam sistem file.

Tipe: Long

Rentang yang Valid: Nilai minimum 0.

Wajib: Ya

Timestamp

Waktu di mana ukuran data, yang dikembalikan di Value lapangan, ditentukan. Nilainya adalah bilangan bulat detik sejak 1970-01-01T 00:00:00 Z.

Tipe: Timestamp

Wajib: Tidak

ValueInArchive

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan di kelas penyimpanan Arsip.

Tipe: Long

Rentang yang Valid: Nilai minimum 0.

Wajib: Tidak

ValueInIA

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan di kelas penyimpanan Akses Jarang.

Tipe: Long

Rentang yang Valid: Nilai minimum 0.

Wajib: Tidak

ValueInStandard

Ukuran terukur terbaru yang diketahui (dalam byte) data yang disimpan di kelas penyimpanan Standar.

Tipe: Long

Rentang yang Valid: Nilai minimum 0.

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LifecyclePolicy

Menjelaskan kebijakan yang digunakan oleh manajemen siklus hidup yang menentukan kapan harus mentransisikan file masuk dan keluar dari kelas penyimpanan. Untuk informasi selengkapnya, lihat [Mengelola penyimpanan sistem file](#).

Note

Saat menggunakan perintah `put-lifecycle-configuration` CLI atau tindakan `PutLifecycleConfiguration` API, Amazon EFS mengharuskan setiap `LifecyclePolicy` objek hanya memiliki satu transisi. Ini berarti bahwa dalam badan permintaan, `LifecyclePolicies` harus terstruktur sebagai array `LifecyclePolicy` objek, satu objek untuk setiap transisi. Untuk informasi selengkapnya, lihat contoh permintaan di [PutLifecycleConfiguration](#).

Daftar Isi

TransitionToArchive

Jumlah hari setelah file terakhir diakses di penyimpanan utama (kelas penyimpanan Standar) untuk memindahkannya ke penyimpanan Arsip. Operasi metadata seperti mendaftar konten direktori tidak dihitung sebagai peristiwa akses file.

Tipe: String

Nilai yang Valid: `AFTER_1_DAY` | `AFTER_7_DAYS` | `AFTER_14_DAYS` | `AFTER_30_DAYS` | `AFTER_60_DAYS` | `AFTER_90_DAYS` | `AFTER_180_DAYS` | `AFTER_270_DAYS` | `AFTER_365_DAYS`

Wajib: Tidak

TransitionToIA

Jumlah hari setelah file terakhir diakses di penyimpanan primer (kelas penyimpanan Standar) untuk memindahkannya ke penyimpanan Akses Jarang (IA). Operasi metadata seperti mendaftar konten direktori tidak dihitung sebagai peristiwa akses file.

Tipe: String

Nilai yang Valid: AFTER_7_DAYS | AFTER_14_DAYS | AFTER_30_DAYS | AFTER_60_DAYS
| AFTER_90_DAYS | AFTER_1_DAY | AFTER_180_DAYS | AFTER_270_DAYS |
AFTER_365_DAYS

Wajib: Tidak

TransitionToPrimaryStorageClass

Apakah akan memindahkan file kembali ke penyimpanan primer (Standar) setelah diakses di penyimpanan IA atau Arsip. Operasi metadata seperti mendaftar konten direktori tidak dihitung sebagai peristiwa akses file.

Tipe: String

Nilai yang Valid: AFTER_1_ACCESS

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

MountTargetDescription

Memberikan deskripsi target mount.

Daftar Isi

FileSystemId

ID sistem file tempat target mount dimaksudkan.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

LifeCycleState

Status siklus hidup target pemasangan.

Tipe: String

Nilai yang Valid: `creating | available | updating | deleting | deleted | error`

Wajib: Ya

MountTargetId

ID target pemasangan yang ditetapkan sistem.

Tipe: String

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsmt-[0-9a-f]{8,40}$`

Wajib: Ya

SubnetId

ID subnet target mount.

Tipe: String

Kendala Panjang: Panjang minimum 15. Panjang maksimum 47.

Pola: `^subnet-[0-9a-f]{8,40}$`

Wajib: Ya

AvailabilityZoneId

Pengidentifikasi unik dan konsisten dari Availability Zone tempat target mount berada. Misalnya, `us-east-1-az1` adalah ID AZ untuk Wilayah `us-east-1` dan memiliki lokasi yang sama di setiap wilayah. Akun AWS

Tipe: String

Wajib: Tidak

AvailabilityZoneName

Nama Availability Zone di mana target mount berada. Availability Zones dipetakan secara independen ke nama masing-masing Akun AWS. Misalnya, Availability Zone `us-east-1a` untuk Anda Akun AWS mungkin bukan lokasi yang sama dengan `us-east-1a` yang lain Akun AWS.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

Wajib: Tidak

IpAddress

IPv4 Alamat untuk target pemasangan.

Tipe: String

Batasan Panjang: Panjang minimum 7. Panjang maksimum 15.

Pola: `^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

Wajib: Tidak

Ipv6Address

IPv6 Alamat untuk target pemasangan.

Tipe: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 39.

Wajib: Tidak

NetworkInterfaceId

ID antarmuka jaringan yang dibuat Amazon EFS saat membuat target pemasangan.

Tipe: String

Wajib: Tidak

OwnerId

Akun AWS ID yang memiliki sumber daya.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

Wajib: Tidak

VpcId

ID virtual private cloud (VPC) tempat target mount dikonfigurasi.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PosixUser

Identitas POSIX lengkap, termasuk ID pengguna, ID grup, dan grup sekunder apa pun IDs, pada titik akses yang digunakan untuk semua operasi sistem file yang dilakukan oleh klien NFS menggunakan titik akses.

Daftar Isi

Gid

ID grup POSIX digunakan untuk semua operasi sistem file menggunakan titik akses ini.

Tipe: Long

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Ya

Uid

ID pengguna POSIX digunakan untuk semua operasi sistem file menggunakan titik akses ini.

Tipe: Long

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Ya

SecondaryGids

Grup POSIX sekunder IDs digunakan untuk semua operasi sistem file menggunakan titik akses 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 bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ReplicationConfigurationDescription

Menjelaskan konfigurasi replikasi untuk sistem file tertentu.

Daftar Isi

CreationTime

Menjelaskan kapan konfigurasi replikasi dibuat.

Tipe: Timestamp

Wajib: Ya

Destinations

Array objek tujuan. Hanya satu objek tujuan yang didukung.

Tipe: Array objek [Destination](#)

Wajib: Ya

OriginalSourceFileSystemArn

Nama Sumber Daya Amazon (ARN) dari sistem file EFS sumber asli dalam konfigurasi replikasi.

Tipe: String

Diperlukan: Ya

SourceFileSystemArn

Nama Sumber Daya Amazon (ARN) dari sistem file sumber saat ini dalam konfigurasi replikasi.

Tipe: String

Diperlukan: Ya

SourceFileSystemId

ID dari sistem file Amazon EFS sumber yang sedang direplikasi.

Tipe: String

Batasan Panjang: Panjang maksimum 128.

Pola: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Wajib: Ya

SourceFileSystemRegion

AWS Region Di mana sistem file EFS sumber berada.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

Wajib: Ya

SourceFileSystemOwnerId

ID Akun AWS tempat sistem file sumber berada.

Tipe: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ResourceIdPreference

Menjelaskan jenis sumber daya dan preferensi ID-nya untuk pengguna Akun AWS, saat ini AWS Region.

Daftar Isi

ResourceIdType

Mengidentifikasi preferensi ID sumber daya EFS, baik LONG_ID (17 karakter) atau SHORT_ID (8 karakter).

Tipe: String

Nilai yang Valid: LONG_ID | SHORT_ID

Wajib: Tidak

Resources

Mengidentifikasi resource Amazon EFS yang menerapkan setelan preferensi ID, FILE_SYSTEM dan MOUNT_TARGET.

Tipe: Array string

Nilai yang Valid: FILE_SYSTEM | MOUNT_TARGET

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

RootDirectory

Menentukan direktori pada sistem file Amazon EFS bahwa yang diberikan akses oleh titik akses. Titik akses mengekspos jalur sistem file yang ditentukan selagi direktori root dari sistem file Anda ke aplikasi menggunakan titik akses. Klien NFS yang menggunakan titik akses hanya dapat mengakses data di titik akses `RootDirectory` dan subdirektornya.

Daftar Isi

CreationInfo

(Opsional) Menentukan POSIX IDs dan izin untuk diterapkan ke titik akses. `RootDirectory` Jika `RootDirectory > Path` yang ditentukan tidak ada, EFS membuat direktori root menggunakan pengaturan `CreationInfo` ketika klien terhubung ke titik akses. Ketika menentukan `CreationInfo`, Anda harus menyediakan nilai untuk semua properti.

Important

Jika Anda tidak menyediakan `CreationInfo` dan `RootDirectory > Path` yang ditentukan tidak ada, upaya untuk memasang sistem file menggunakan titik akses akan gagal.

Tipe: Objek [CreationInfo](#)

Wajib: Tidak

Path

Menentukan jalur pada sistem file EFS untuk diekspos sebagai direktori root ke klien NFS menggunakan titik akses untuk mengakses sistem file EFS. Sebuah jalur dapat memiliki hingga empat subdirektori. Jika jalur yang ditentukan tidak ada, Anda diminta untuk menyediakan `CreationInfo`.

Tipe: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 100.

Pola: `^(\\|\\(?!\\.)([^\$#<>;`|&?{}^*\\/\\n]+){1,4})$`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Tag

Tanda merupakan pasangan nilai kunci. Karakter yang diizinkan adalah huruf, spasi putih, dan angka yang dapat direpresentasikan dalam UTF-8, dan karakter berikut: + - = . _ : /

Daftar Isi

Key

Kunci tag (String). Kunci tidak dapat diawali dengan aws :.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: $^{(?![aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_.: /+=\ -@]+)}\$$

Wajib: Ya

Value

Nilai dari kunci tag.

Jenis: String

Batasan Panjang: Panjang maksimum 256.

Pola: $^{([\p{L}\p{Z}\p{N}_.: /+=\ -@]^*)}\$$

Diperlukan: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Riwayat dokumen

- Versi API: 2015-02-01
- Pembaruan dokumentasi terbaru: 9 Juni 2025

Tabel berikut menjelaskan perubahan penting pada Panduan Pengguna Amazon Elastic File System setelah Juli 2018. Untuk notifikasi tentang pembaruan dokumentasi, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
Support untuk IPv6	Internet Protocol Version 6 (IPv6) didukung pada Amazon EFS Service APIs dan mount target. Untuk informasi selengkapnya tentang Amazon EFS API, lihat Amazon EFS API . Untuk informasi selengkapnya tentang target pemasangan, lihat Mengelola target pemasangan .	Juni 9, 2025
Peningkatan throughput elastis pada tertentu Wilayah AWS	Throughput pembacaan maksimum untuk sistem file yang menggunakan throughput Elastic di Wilayah berikut telah meningkat dari 20 GiBps menjadi 60 GiBps: Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Sydney), dan Eropa (London). Untuk informasi selengkapnya, lihat kuota Amazon EFS yang dapat Anda tingkatkan .	2 Mei 2025

Peningkatan throughput elastis pada tertentu Wilayah AWS

Throughput pembacaan maksimum untuk sistem file yang menggunakan throughput Elastis di Wilayah tertentu adalah 60. GiBps. Di semua Wilayah lainnya, batas throughput baca untuk sistem file yang menggunakan throughput Elastic telah ditingkatkan dari 10 menjadi 20. GiBps. Untuk informasi selengkapnya, lihat [kuota Amazon EFS yang dapat Anda tingkatkan](#).

April 10, 2025

Peningkatan kuota access point

Jumlah maksimum titik akses yang dapat dimiliki oleh sistem file tunggal telah meningkat dari 1.000 menjadi 10.000. Anda juga dapat meminta kenaikan untuk batas ini. Untuk informasi selengkapnya, lihat [Kuota sumber daya yang dapat Anda tingkatkan](#).

Februari 10, 2025

Tingkat pencadangan dan pemulihan yang ditingkatkan

Kecepatan melakukan pencadangan dan pemulihan telah meningkat. Untuk informasi selengkapnya, lihat [Performa Backup](#).

Januari 8, 2025

[Peningkatan kuota IOPS berdasarkan permintaan](#)

Anda sekarang dapat meminta peningkatan IOPS untuk sistem file EFS menggunakan mode throughput Elastic. Untuk informasi selengkapnya, lihat [kuota Amazon EFS yang dapat Anda tingkatkan](#).

November 26, 2024

[Support replikasi lintas akun](#)

Amazon EFS mendukung replikasi lintas akun. Untuk informasi selengkapnya, lihat [Mereplikasi sistem file EFS di seluruh Akun AWS](#).

November 19, 2024

[Kebijakan AWS terkelola yang ada diperbarui](#)

ReplicationRead, ReplicationWrite, iam:PassRole ditambahkan ke AmazonElasticFileSystemFullAccess kebijakan. Untuk informasi selengkapnya, lihat [AmazonElasticFileSystemFullAccess](#).

November 7, 2024

[Kebijakan AWS terkelola yang ada diperbarui](#)

ReplicationRead dan ReplicationWrite ditambahkan ke AmazonElasticFileSystemServiceRolePolicy. Untuk informasi lebih lanjut, lihat [AmazonElasticFileSystemServiceRolePolicy](#).

November 7, 2024

Kebijakan AWS terkelola yang ada diperbarui	Izin ReplicationRead ditambahkan ke AmazonElasticFileSystemReadOnlyAccess kebijakan yang ada. Untuk informasi selengkapnya, lihat AmazonElasticFileSystemReadOnlyAccess .	November 7, 2024
Batas throughput elastis meningkat	Batas throughput elastis telah meningkat menjadi 60 gibibyte per detik (GiBps) untuk spesifik Wilayah AWS dan menjadi 10 GiBps untuk semua Wilayah lainnya. Untuk informasi selengkapnya, lihat Total throughput Elastis default untuk semua klien yang terhubung di masing-masing AWS Region klien .	Oktober 14, 2024
Kebijakan AWS terkelola yang ada diperbarui	Elemen opsional Sid (ID pernyataan) sekarang disertakan dalam pernyataan AmazonElasticFileSystemReadOnlyAccess kebijakan. Nilai Sid adalah ElasticFileSystemReadOnlyAccess . Untuk informasi selengkapnya tentang elemen Sid kebijakan, lihat elemen kebijakan IAM JSON: Sid .	Agustus 7, 2024

Batas throughput elastis meningkat

Batas throughput elastis telah meningkat untuk spesifik Wilayah AWS. Untuk informasi selengkapnya, lihat [Total throughput Elastis default untuk semua klien yang terhubung di masing-masing AWS Region klien](#).

Juli 31, 2024

Peningkatan kuota untuk target mount

Jumlah maksimum target pemasangan untuk setiap virtual private cloud (VPC) meningkat dari 400 menjadi 1.400. Untuk informasi selengkapnya, lihat [Kuota sumber daya Amazon EFS yang tidak dapat Anda ubah](#).

15 Mei 2024

Peningkatan batas throughput gabungan untuk sistem file Elastic

Throughput baca dan tulis gabungan maksimum adalah 1.500 MiBps untuk sistem file yang menggunakan throughput Elastic dan dipasang menggunakan versi 2.0 atau yang lebih baru dari klien Amazon EFS (amazon-efs-utils versi) atau Amazon EFS CSI Driver (. aws-efs-csi-driver Untuk informasi selengkapnya, lihat tabel ringkasan Kinerja di [performa Amazon EFS](#).

April 30, 2024

Batas throughput elastis meningkat	Batas throughput elastis telah meningkat untuk spesifik Wilayah AWS. Untuk informasi selengkapnya, lihat Total throughput Elastis default untuk semua klien yang terhubung di masing-masing AWS Region klien .	Maret 13, 2024
Peningkatan IOPS	Sistem file yang menggunakan an throughput Elastic dapat mendorong maksimum 90.000 pembacaan untuk data yang jarang diakses. Untuk informasi selengkapnya, lihat Ringkasan kinerja .	Januari 22, 2024
Kebijakan AWS terkelola yang ada diperbarui	Izin elasticfilesystem: UpdateFileSystemProtection ditambahkan ke AmazonElasticFileSystemFull Access kebijakan yang ada untuk mengizinkan prinsipal memperbarui perlindungan pada sistem file. Untuk informasi selengkapnya, lihat Amazon EFS memperbarui kebijakan AWS terkelola .	27 November 2023
Replikasi ke sistem file yang ada	Sistem file sekarang dapat direplikasi ke sistem file yang ada, sehingga lebih mudah untuk menyinkronkan perubahan antara sistem file untuk tujuan failback. Untuk informasi selengkapnya, lihat Sistem file tujuan .	27 November 2023

[Perlindungan sistem file ditambahkan](#)

Perlindungan penyimpanan replikasi telah ditambahkan ke sistem file dan diaktifkan secara default. Perlindungan mencegah sistem file digunakan sebagai tujuan dalam konfigurasi replikasi. Untuk informasi selengkapnya, lihat [Perlindungan sistem file](#).

27 November 2023

[Kelas penyimpanan baru, jenis sistem file, dan kebijakan siklus hidup](#)

Amazon EFS sekarang menawarkan kelas penyimpanan EFS Archive, jenis sistem file, dan kebijakan siklus hidup Transisi ke Arsip. Untuk informasi selengkapnya, lihat [Jenis sistem file dan kelas penyimpanan](#).

26 November 2023

[Peningkatan IOPS](#)

Sistem file throughput elastis sekarang mendukung maksimum 65.000 operasi baca dan 50.000 operasi tulis IOPS untuk data yang jarang diakses, dan sekarang mendukung 250.000 IOPS baca untuk data yang sering diakses. Untuk informasi selengkapnya, lihat [Ringkasan kinerja](#).

26 November 2023

[Hapus konfigurasi replikasi dari sistem file sumber](#)

Konfigurasi replikasi sekarang dapat dihapus dari sistem file sumber. Untuk informasi selengkapnya, lihat [Menghapus konfigurasi replikasi](#).

September 19, 2023

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Wilayah Israel (Tel Aviv).

Agustus 7, 2023

[Peningkatan kinerja untuk sistem file mode Tujuan Umum](#)

Sistem file mode Tujuan Umum Amazon EFS sekarang mendukung hingga 55.000 operasi baca per detik dan 25.000 operasi tulis. Untuk informasi selengkapnya, lihat [Kuota untuk Sistem File Amazon EFS](#).

3 Agustus 2023

[Batas throughput yang disediakan meningkat](#)

Batas throughput yang disediakan telah meningkat untuk spesifik. Wilayah AWS Untuk informasi selengkapnya, lihat [Total throughput standar yang disediakan untuk semua klien yang terhubung di masing-masing klien](#). AWS Region

Juni 21, 2023

[Dukungan Wilayah yang Diperluas untuk replikasi EFS](#)

Replikasi EFS sekarang tersedia Wilayah AWS di semua tempat EFS tersedia. Untuk informasi selengkapnya, lihat [replikasi Amazon EFS](#).

28 April 2023

Peningkatan batas throughput elastis	Batas throughput elastis telah meningkat untuk spesifik Wilayah AWS. Untuk informasi selengkapnya, lihat tabel Total default Elastic throughput untuk semua klien yang terhubung di masing-masing AWS Region klien .	17 April 2023
Elastic menggantikan Bursting sebagai mode throughput default	Mode throughput default (dan direkomendasikan) untuk sistem file sekarang Elastis, bukan Bursting. Untuk informasi selengkapnya, lihat Mode throughput .	13 April 2023
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Wilayah Asia Pasifik (Melbourne).	12 April 2023
Support ditambahkan untuk macOS Ventura	Amazon EFS sekarang dapat diinstal pada instans EC2 Mac yang berjalan di macOS Ventura. Untuk informasi selengkapnya, lihat Distribusi yang didukung .	10 April 2023
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Wilayah Asia Pasifik (Hyderabad).	16 Februari 2023
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Eropa (Spanyol AWS Region).	19 Januari 2023

Batas titik akses untuk sistem file telah meningkat	Jumlah maksimum titik akses yang dapat dimiliki oleh sistem file tunggal telah meningkat dari 120 menjadi 1.000. Untuk informasi selengkapnya, lihat Kuota sumber daya .	Januari 17, 2023
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Eropa (Zurich). AWS Region	15 Desember 2022
Support ditambahkan untuk kebijakan siklus hidup satu hari	Anda sekarang dapat memilih satu hari untuk kebijakan siklus hidup Transisi ke IA. Untuk informasi selengkapnya, lihat Menggunakan kebijakan Siklus Hidup .	27 November 2022
Mengurangi latensi baca dan tulis	Latensi untuk membaca dan menulis data file telah berkurang untuk penyimpanan One Zone dan sistem file penyimpanan Standar. Untuk informasi selengkapnya, lihat Ringkasan kinerja .	27 November 2022
Mode throughput tambahan ditambahkan	Mode throughput elastis ditambahkan sebagai opsi throughput untuk sistem file Amazon EFS. Untuk informasi lebih lanjut, lihat Throughput elastis .	27 November 2022

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Wilayah Timur Tengah (UEA).

Oktober 17, 2022

[Support ditambahkan untuk EFS Replication](#)

Amazon EFS telah menghapus batas sebelumnya di mana replikasi EFS tidak mendukung soket dan pipa bernama, atau FIFOs.

15 September 2022

[Batas jumlah kunci file per koneksi telah meningkat](#)

Jumlah kunci file per koneksi telah meningkat dari 8192 menjadi 65.536. Untuk informasi selengkapnya, lihat [Kuota untuk klien NFS](#).

4 Mei, 2022

[Batas untuk proses menggunakan kunci file dihapus](#)

Amazon EFS telah menghapus batas sebelumnya di mana maksimum 256 proses pada satu instance dapat menggunakan kunci file pada saat yang sama. Untuk informasi selengkapnya, lihat [Kuota untuk klien NFS](#).

4 Mei, 2022

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Asia Pasifik (Jakarta) AWS Region.

27 Januari 2022

[Support ditambahkan untuk EFS Replication](#)

Gunakan EFS Replication untuk mereplikasi data dan metadata pada sistem file EFS ke sistem file EFS lain sesuai pilihan Anda. AWS Region Untuk informasi lebih lanjut, lihat [Amazon EFS replikasi](#).

Januari 25, 2022

[Sistem file dan sumber daya target mount menggunakan format ID sumber daya 17 karakter](#)

Sistem file Amazon EFS baru dan sumber daya target pemasangan sekarang ditetapkan 17 karakter IDs. Untuk informasi selengkapnya, lihat [Bekerja dengan sumber daya Amazon EFS](#).

Oktober 22, 2021

[Support ditambahkan untuk EFS Intelligent-Tiering](#)

EFS Intelligent-Tiering menggunakan EFS Lifecycle Management untuk memantau pola akses file dan dirancang untuk secara otomatis mentransisikan file ke dan dari kelas penyimpanan Infrequent Access (IA) yang sesuai. Untuk informasi selengkapnya, lihat [EFS Intelligent-Tiering and Lifecycle Management](#).

2 September 2021

[Support ditambahkan untuk menguji format ID sumber daya 17 karakter](#)

Amazon EFS beralih dari menggunakan 8 karakter IDs ke 17 karakter IDs untuk sistem file dan memasang target pada 1 Oktober 2021. Selama transisi ini, Anda dapat ikut serta dan mulai menggunakan sumber daya IDs 17 karakter per AWS Region basis. Untuk informasi selengkapnya, lihat [Sumber Daya IDs](#).

5 Mei 2021

[Support ditambahkan untuk memasang sistem file One Zone dari Availability Zone yang berbeda menggunakan Amazon EFS mount helper](#)

Anda sekarang dapat menggunakan EFS mount helper untuk memasang sistem file Amazon EFS yang menggunakan kelas penyimpanan One Zone ke instans EC2 yang berada di Availability Zone yang berbeda. Anda dapat menggunakan az opsi baru untuk menentukan Availability Zone dari sistem file Amazon EFS. Untuk informasi selengkapnya, lihat [Memasang sistem file dengan kelas penyimpanan One Zone](#).

6 April 2021

[Support ditambahkan untuk kelas penyimpanan EFS One Zone](#)

Kelas penyimpanan Amazon EFS One Zone menyimpan data secara berlebihan dalam satu Availability Zone dalam file. AWS Region Kelas penyimpanan EFS One Zone dan One Zone-Infr equent Access (One Zone-IA) adalah opsi hemat biaya untuk menyimpan data yang tidak memerlukan ketahanan Multi-AZ dari kelas penyimpanan EFS Standard dan Standard-IA. Untuk informasi selengkapnya, lihat [Menggunakan kelas penyimpanan EFS](#).

9 Maret 2021

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Asia Pasifik (Osaka) AWS Region.

3 Maret 2021

[Support ditambahkan untuk instans macOS Amazon EC2 yang menjalankan macOS Big Sur](#)

Sekarang Anda dapat memasang sistem file Amazon EFS dari instans macOS EC2 yang menjalankan macOS Big Sur dengan menggunakan EFS mount helper atau dengan menggunakan perintah pemasangan NFS. Untuk informasi selengkapnya, lihat [Memasang dengan EFS mount helper](#) atau [Mounting file system tanpa EFS mount helper](#).

23 Februari 2021

[Konsol Amazon EFS baru tersedia di AWS GovCloud \(US\) Wilayah](#)

Konsol Amazon EFS baru sekarang tersedia di AWS GovCloud (US) AWS Region.

10 Februari 2021

[Support ditambahkan untuk CloudWatch metrik Amazon EFS baru MeteredIOBytes](#)

Anda dapat menggunakan `MeteredIOBytes` untuk mengukur jumlah byte terukur untuk setiap operasi sistem file, termasuk pembacaan data, penulisan data, dan operasi metadata. Operasi baca diukur pada sepertiga tingkat operasi lainnya. Untuk informasi selengkapnya, lihat [CloudWatchMetrik Amazon untuk Amazon EFS](#).

28 Januari 2021

[Amazon EFS meningkatkan throughput baca sistem file sebesar 300%](#)

Sistem file Amazon EFS sekarang mengukur permintaan baca pada sepertiga tingkat permintaan lainnya.

28 Januari 2021

[Support ditambahkan untuk CloudWatch metrik Amazon EFS baru StorageBytes](#)

Anda dapat menggunakan `StorageBytes` untuk mengukur dan memantau ukuran sistem file dalam byte, termasuk jumlah data yang disimpan dalam kelas penyimpanan Standard dan Infrequent Access. Untuk informasi selengkapnya, lihat [CloudWatch Metrik Amazon untuk Amazon EFS](#).

11 Januari 2021

[Gunakan AWS Transfer Family untuk mengakses sistem file Amazon EFS](#)

Anda dapat menggunakan an AWS Transfer Family untuk mentransfer file masuk dan keluar dari sistem file Amazon EFS Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS Transfer Family untuk mengakses file di sistem file EFS Anda.](#)

6 Januari 2021

[Gunakan AWS Systems Manager untuk mengelola klien Amazon EFS \(amazon-efs-utils \)](#)

Anda dapat menggunakannya AWS Systems Manager untuk menginstal atau memperbarui klien Amazon EFS (amazon-efs-utils) secara otomatis pada instans EC2 Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS Systems Manager untuk menginstall atau memperbarui klien Amazon EFS secara otomatis.](#)

29 September 2020

[Menegakkan pembuatan sistem file EFS terenkripsi](#)

Anda dapat menggunakan kunci kondisi elasticfilesystem:Encrypted AWS Identity and Access Management (IAM) untuk memaksakan agar pengguna membuat sistem file Amazon EFS yang dienkripsi saat istirahat. Untuk informasi selengkapnya, lihat [Menegakkan Pembuatan Sistem File Amazon EFS Terenkripsi](#) saat Istirahat.

16 September 2020

Throughput Amazon EFS per klien meningkat 100%	EFS sekarang mendukung hingga 500 MB/s throughput per klien, peningkatan 100% dari batas sebelumnya 250 MB/s. Untuk informasi selengkapnya, lihat Kuota untuk sistem file Amazon EFS .	23 Juli 2020
Support ditambahkan untuk backup harian otomatis sistem file Amazon EFS	Pencadangan harian otomatis sekarang diaktifkan secara default saat membuat sistem file menggunakan konsol EFS. Untuk informasi selengkapnya, lihat Menggunakan AWS Backup dengan Amazon EFS .	Juli 16, 2020
Alur kerja Quick Create baru menyederhanakan pembuatan sistem file Amazon EFS	Menggunakan opsi Quick Create di konsol EFS, Anda dapat membuat sistem file EFS menggunakan pengaturan yang direkomendasikan layanan dengan satu tombol. Untuk informasi selengkapnya, lihat Membuat sistem file EFS Anda .	Juli 16, 2020
Konsol Amazon EFS baru sekarang tersedia	Konsol EFS baru memudahkan Anda menggunakan Amazon EFS dan menyederhanakan pengelolaan sistem file EFS Anda.	Juli 16, 2020

[Amazon EFS meningkatkan throughput minimum sistem file](#)

Sistem file Amazon EFS yang menggunakan throughput Bursting sekarang memiliki throughput minimum 1 MiB/s. Untuk informasi selengkapnya, lihat [Mode throughput](#).

30 Juni 2020

[Kinerja sistem file mode Tujuan Umum meningkat](#)

Sistem file mode Tujuan Umum Amazon EFS sekarang mendukung hingga 35.000 operasi baca per detik, meningkat 400% dari batas sebelumnya 7.000. Untuk informasi selengkapnya, lihat [Kuota untuk Sistem File Amazon EFS](#).

1 April 2020

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Beijing dan Ningxia Wilayah AWS.

22 Januari 2020

[Support ditambahkan untuk otorisasi IAM untuk klien NFS](#)

Anda sekarang dapat menggunakan AWS Identity and Access Management (IAM) untuk mengelola akses NFS ke sistem file Amazon EFS. Untuk informasi selengkapnya, lihat [Menggunakan AWS IAM untuk Mengontrol Akses NFS ke Amazon EFS](#).

13 Januari 2020

[Support ditambahkan untuk EFS Access Points](#)

Jalur akses Amazon EFS adalah titik masuk khusus aplikasi ke dalam sistem file Amazon EFS yang memudahkan pengelolaan akses aplikasi ke kumpulan data bersama. Untuk informasi selengkapnya, lihat [Bekerja dengan Amazon EFS Access Points](#).

13 Januari 2020

[Support ditambahkan untuk pemulihan AWS Backup sebagian.](#)

Anda sekarang dapat memulihkan file dan direktori tertentu menggunakan pemulihan sebagian, selain memulihkan titik pemulihan lengkap. Untuk informasi selengkapnya, lihat [Menggunakan AWS Backup dengan Amazon EFS](#).

13 Januari 2020

[Support ditambahkan untuk peran terkait layanan IAM](#)

Amazon EFS sekarang menggunakan peran terkait layanan berdasarkan IAM, sehingga memudahkan penyiapan EFS dengan menambahkan izin yang diperlukan secara otomatis. Untuk informasi selengkapnya, lihat [Menggunakan Peran Tertaut Layanan untuk Amazon EFS](#).

10 Desember 2019

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Eropa (Stockholm) AWS Region.

20 November 2019

AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Asia Pasifik (Hong Kong) AWS Region.	20 November 2019
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Amerika Selatan (São Paulo). AWS Region	20 November 2019
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Timur Tengah (Bahrain). AWS Region	20 November 2019
Kebijakan manajemen Siklus Hidup 7 hari baru ditambahkan	Manajemen siklus hidup sekarang memiliki kebijakan tambahan untuk memindahkan data ke kelas penyimpanan Akses Jarang yang hemat biaya setelah 7 hari. Untuk informasi selengkapnya, lihat EFS Lifecycle Management .	6 November 2019
Support ditambahkan untuk Interface VPC Endpoints	Anda dapat membuat koneksi pribadi antara cloud pribadi virtual Anda dan Amazon EFS untuk memanggil EFS API. Untuk informasi selengkapnya, lihat Bekerja dengan Titik Akhir VPC .	22 Oktober 2019

[Pasang sistem file EFS saat meluncurkan instans EC2 baru.](#)

Sekarang Anda dapat mengonfigurasi instans Amazon EC2 baru untuk memasang sistem file EFS Anda saat diluncurkan di Wizard Instans Peluncuran EC2. Untuk informasi lebih lanjut, lihat [Langkah 2. Buat Sumber Daya EC2 Anda dan Luncurkan Instans EC2 Anda.](#)

17 Oktober 2019

[Support untuk Service Quotas ditambahkan](#)

Anda sekarang dapat melihat semua batas Amazon EFS di konsol Service Quotas. Untuk informasi selengkapnya, lihat [Batas Amazon EFS.](#)

10 September 2019

[Kebijakan manajemen siklus hidup baru ditambahkan](#)

Saat menggunakan Manajemen Siklus Hidup, Anda sekarang dapat memilih salah satu dari empat kebijakan siklus hidup untuk menentukan kapan file dialihkan ke kelas penyimpanan Akses Jarang yang hemat biaya. Untuk informasi selengkapnya, lihat [EFS Lifecycle Management.](#)

9 Juli 2019

EFS Lifecycle Management sekarang tersedia di semua sistem file EFS.	Fitur EFS Lifecycle Management sekarang tersedia di semua sistem file EFS. Pembatasan sebelumnya berdasarkan kapan sistem file dibuat sekarang dihapus. Untuk informasi selengkapnya, lihat EFS Lifecycle Management .	9 Juli 2019
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Eropa (Paris) AWS Region.	12 Juni 2019
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Asia Pasifik (Mumbai) AWS Region.	5 Juni 2019
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Kanada (Tengah AWS Region).	1 Mei 2019
Pembaruan API: Tag sekarang menjadi bagian dari muatan CreateFileSystem operasi	Anda sekarang dapat menyertakan tag saat menggunakan CreateFileSystem operasi AWS API dan CLI untuk membuat sistem file Amazon EFS. Untuk informasi selengkapnya, lihat CreateFileSystem dan Membuat Sistem File Menggunakan AWS CLI .	19 Februari 2019

[Fitur baru: Kelas penyimpanan EFS Infrequent Access dan manajemen siklus hidup EFS](#)

Amazon EFS Infrequent Access adalah kelas penyimpanan yang dioptimalkan biaya untuk file yang jarang diakses. Manajemen siklus hidup EFS secara otomatis mentransisikan file dari penyimpanan Standard ke Infrequent Access. Untuk informasi selengkapnya, lihat [EFS Storage Classes](#).

13 Februari 2019

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Eropa (London) AWS Region.

23 Januari 2019

[AWS Backup Integrasi layanan dengan Amazon EFS](#)

Sistem file Amazon EFS dapat dicadangkan menggunakan AWS Backup, layanan pencadangan otomatis yang dikelola sepenuhnya, terpusat, dan otomatis untuk mencadangkan data di seluruh AWS layanan di cloud dan di tempat. Untuk informasi selengkapnya, lihat [AWS Backup dan Amazon EFS](#).

16 Januari 2019

[Dukungan koneksi Transit Gateway ke sistem penyimpanan lokal ditambahkan.](#)

Sistem file Amazon EFS sekarang dapat diakses menggunakan koneksi Transit Gateway ke sistem penyimpanan lokal. Untuk informasi selengkapnya, lihat [Memasang dari Akun Lain atau VPC dan Panduan: Pasang Sistem File dari VPC yang Berbeda.](#)

6 Desember 2018

[EFS File Sync sekarang menjadi bagian dari AWS DataSync layanan baru.](#)

AWS DataSync adalah layanan transfer data terkelola yang menyederhanakan sinkronisasi sejumlah besar data antara sistem penyimpanan lokal dan AWS layanan penyimpanan. Untuk informasi selengkapnya, lihat [Mentransfer File dari Sistem File Lokal ke Amazon EFS Menggunakan AWS DataSync.](#)

26 November 2018

[VPN dan dukungan koneksi peering VPC Antar wilayah ditambahkan](#)

Amazon EFS sekarang dapat diakses melalui koneksi VPN dan koneksi peering VPC antar wilayah. Untuk informasi selengkapnya, lihat [Mentransfer File dari Sistem File Lokal ke Amazon EFS Menggunakan AWS DataSync.](#)

23 Oktober 2018

[VPN dan dukungan koneksi peering VPC Antar wilayah ditambahkan](#)

Sistem file Amazon EFS sekarang dapat diakses melalui koneksi VPN dan koneksi peering VPC antar wilayah. Untuk informasi selengkapnya, lihat [Memasang dari Akun Lain atau VPC](#) dan Cara [Kerja Amazon EFS dengan Direct Connect](#) dan. VPNs

23 Oktober 2018

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Asia Pasifik (Singapura AWS Region).

13 Juli 2018

[Memperkenalkan mode Throughput yang Disediakan](#)

Anda sekarang dapat menyediakan throughput untuk sistem file baru atau yang sudah ada dengan mode Provisioned Throughput yang baru. Untuk informasi selengkapnya, lihat [Mode throughput](#).

12 Juli 2018

[AWS Region Dukungan tambahan ditambahkan](#)

Amazon EFS sekarang tersedia untuk semua pengguna di Asia Pasifik (Tokyo) AWS Region.

11 Juli 2018

Tabel berikut menjelaskan perubahan penting pada Panduan Pengguna Amazon Elastic File System sebelum Juli 2018.

Ubah	Deskripsi	Tanggal Diubah
AWS Region Dukungan	Amazon EFS sekarang tersedia untuk semua pengguna di AWS Wilayah Asia Pasifik (Seoul).	30 Mei 2018

Ubah	Deskripsi	Tanggal Diubah
tambahan ditambahkan		
Menambahk an dukungan matematika CloudWatch metrik	Matematika metrik memungkinkan Anda untuk menanyakan beberapa CloudWatch metrik dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini. Untuk informasi selengkapnya, lihat Menggunakan matematik a metrik dengan CloudWatch metrik .	4 April 2018
Menambahkan amazon-efs- utils set alat sumber terbuka, dan menambahk an enkripsi dalam perjalanan	amazon-efs-utils Alat ini adalah seperangkat file executable open-source yang menyederhanakan aspek penggunaan Amazon EFS, seperti pemasanga n. Tidak ada biaya tambahan untuk digunakan amazon-efs-utils , dan Anda dapat mengunduh alat-alat ini dari GitHub. Untuk informasi selengkapnya, lihat Menginstal klien Amazon EFS . Juga dalam rilis ini, Amazon EFS sekarang mendukung enkripsi dalam perjalanan melalui tunneling Transport Layer Security (TLS). Untuk informasi selengkapnya, lihat Enkripsi data di Amazon EFS .	4 April 2018
Batas sistem file yang diperbarui per AWS Region	Amazon EFS telah meningkatkan batas jumlah sistem file untuk semua akun di semua AWS Region s. Untuk informasi selengkapnya, lihat Kuota sumber daya Amazon EFS yang tidak dapat Anda ubah .	15 Maret 2018
AWS Region Dukungan tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di AS Barat (California Utara) AWS Region.	14 Maret 2018

Ubah	Deskripsi	Tanggal Diubah
Enkripsi data saat istirahat	Amazon EFS sekarang mendukung enkripsi data saat istirahat. Untuk informasi selengkapnya, lihat Enkripsi data di Amazon EFS .	14 Agustus 2017
Dukungan Wilayah tambahan ditambahkan	Amazon EFS sekarang tersedia untuk semua pengguna di Wilayah Eropa (Frankfurt).	Juli 20, 2017
Nama sistem file menggunakan Domain Name System (DNS)	Amazon EFS sekarang mendukung nama DNS untuk sistem file. Nama DNS sistem file secara otomatis menyelesaikan alamat IP target mount di Availability Zone untuk menghubungkan instans Amazon EC2. Untuk informasi selengkapnya, lihat Pemasangan di Amazon EC2 dengan nama DNS .	20 Desember 2016
Peningkatan dukungan tag untuk sistem file	Amazon EFS sekarang mendukung 50 tag per sistem file. Untuk informasi selengkapnya tentang tag di Amazon EFS, lihat Menandai sumber daya EFS .	Selasa, 29 Agustus 2016
Ketersediaan umum	Amazon EFS sekarang umumnya tersedia untuk semua pengguna di Wilayah AS Timur (Virginia N.), AS Barat (Oregon), dan Eropa (Irlandia).	28 Juni 2016
Peningkatan batas sistem file	Jumlah sistem file Amazon EFS yang dapat dibuat per akun untuk masing-masing AWS Region meningkat dari 5 menjadi 10.	Agustus 21, 2015
Latihan Memulai yang Diperbarui	Latihan Memulai telah diperbarui untuk menyederhanakan proses memulai.	17 Agustus 2015
Panduan baru	Ini adalah rilis pertama dari Panduan Pengguna Amazon Elastic File System.	26 Mei 2015

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.