



Panduan Pengguna

# Sistem File Elastis Amazon



# Sistem File Elastis Amazon: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara 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 tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

---

# Table of Contents

Apa itu Amazon Elastic File System? .....	1
Apakah Anda pengguna Amazon EFS pertama kali? .....	2
Cara kerjanya .....	3
Bagaimana Amazon EFS bekerja dengan Amazon EC2 .....	5
Bagaimana Amazon EFS bekerja dengan AWS Direct Connect dan AWS Dikelola 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 EC2 instance .....	18
Transfer file ke sistem file Anda .....	19
Prasyarat .....	20
Pembersihan sumber daya .....	20
Membuat dan mengelola sumber daya .....	22
Ringkasan implementasi .....	23
Sumber Daya IDs .....	24
Token pembuatan dan idempotensi .....	25
Membuat sistem file .....	25
IAM Izin yang diperlukan untuk membuat sistem file .....	26
Opsi konfigurasi .....	26
Menghapus sistem file .....	38
Membuat grup keamanan .....	39
Membuat kebijakan sistem file .....	41
Membuat titik akses .....	43
Menghapus titik akses .....	46
Penandaan pada sumber daya .....	47
Dasar-dasar tag .....	47
Batasan tag .....	47
Menggunakan tag untuk kontrol akses .....	48
Tandai sumber daya Anda .....	48

Tutorial: Membuat subdirektori per pengguna yang dapat ditulis .....	49
Menginstal EFS klien .....	52
Dependensi untuk alat EFS .....	52
Distribusi yang didukung .....	53
Menginstal EFS klien secara manual .....	55
Menginstal EFS klien Amazon di instans Amazon EC2 Linux .....	55
Menginstal EFS klien Amazon di distribusi Linux lainnya .....	56
Menginstal EFS klien pada instance EC2 Mac .....	56
Menginstal EFS klien secara otomatis .....	57
Apa yang dilakukan EFS klien Amazon selama instalasi .....	57
Systems Manager Distributor didukung sistem operasi .....	58
Mengkonfigurasi AWS Systems Manager untuk menginstal klien EFS .....	59
Instalasi dan upgrade botocore .....	61
Upgrade stunnel .....	61
Menyelesaikan masalah dengan menginstal stunnel .....	63
Memasang sistem file .....	66
Pertimbangan pemasangan untuk Linux .....	66
Menggunakan EFS mount helper .....	68
Pengaturan pemasangan yang digunakan oleh EFS mount helper .....	70
Mendapatkan log dukungan .....	71
Prasyarat .....	72
Pemasangan di EC2 Linux .....	73
Pemasangan di EC2 Mac .....	75
Pemasangan dari Wilayah yang berbeda .....	77
Memasang sistem file One Zone .....	78
Pemasangan dengan IAM otorisasi .....	81
Pemasangan dengan titik EFS akses .....	82
Memasang beberapa EC2 instance .....	83
Pemasangan dari akun lain atau VPC .....	84
Menggunakan NFS .....	87
Prasyarat .....	88
Dukungan NFS .....	88
Menginstal NFS klien .....	89
Pengaturan NFS pemasangan yang disarankan .....	91
Memasang EC2 dengan DNS .....	93
Pemasangan dengan alamat IP .....	96

Secara otomatis memasang sistem file .....	98
Instans EC2 Linux baru .....	98
Instans EC2 Linux yang ada .....	100
Instance Linux dan Mac menggunakan NFS .....	104
Melepaskan sistem file .....	106
Tutorial: Membuat dan me-mount sistem file menggunakan AWS CLI .....	107
Prasyarat .....	108
Menyiapkan AWS CLI .....	109
Langkah 1: Buat EC2 sumber daya .....	110
Langkah 2: Buat EFS sumber daya .....	116
Langkah 3: Pasang dan uji sistem file .....	119
Langkah 4: Membersihkan .....	122
Tutorial: Pemasangan dengan klien lokal .....	124
Prasyarat .....	125
Langkah 1: Buat EFS sumber daya Anda .....	127
Langkah 2: Instal NFS klien .....	128
Langkah 3: Pasang sistem EFS file Amazon di Klien lokal Anda .....	129
Langkah 4: Bersihkan sumber daya dan lindungi AWS akun Anda .....	131
Opsional: Mengenkripsi data dalam perjalanan .....	131
Tutorial: Pasang sistem file dari yang berbeda VPC .....	135
Prasyarat .....	135
Langkah 1: Tentukan ID Availability Zone target mount .....	136
Langkah 2: Tentukan alamat IP target mount .....	137
Langkah 3: Tambahkan entri host untuk target pemasangan .....	138
Langkah 4: Pasang sistem file Anda menggunakan EFS mount helper .....	138
Langkah 5: Bersihkan sumber daya dan lindungi Anda Akun AWS .....	140
Memecahkan masalah pemasangan .....	141
Pemasangan sistem file pada instance Windows gagal .....	141
Akses ditolak oleh server .....	141
Pemasangan otomatis gagal dan instans tidak responsif .....	142
Pemasangan beberapa sistem EFS file Amazon di /etc/fstab gagal .....	142
Perintah mount gagal dengan pesan kesalahan “jenis fs yang salah” .....	143
Perintah mount gagal dengan pesan kesalahan “opsi pemasangan yang salah” .....	144
Pemasangan dengan titik akses gagal .....	144
Pemasangan sistem file gagal segera setelah pembuatan sistem file .....	144
Pemasangan sistem file hang dan kemudian gagal dengan kesalahan timeout .....	145

Pemasangan sistem file dengan NFS menggunakan DNS nama gagal .....	145
Pemasangan sistem file gagal dengan “nfs tidak merespons” .....	146
Status siklus hidup target mount macet .....	146
Status siklus hidup target pemasangan menunjukkan kesalahan .....	147
Mount tidak merespons .....	147
Klien yang dipasang akan terputus .....	148
Operasi pada sistem file yang baru dipasang mengembalikan Kesalahan “pegangan file buruk” .....	148
Melepas sistem file gagal .....	149
Mentransfer data .....	150
Menggunakan AWS DataSync .....	150
Menggunakan AWS Transfer Family .....	151
Prasyarat untuk digunakan dengan Amazon AWS Transfer Family EFS .....	152
Mengkonfigurasi sistem EFS file Anda untuk AWS Transfer Family .....	152
Mengelola sistem file .....	158
Mengelola target mount .....	158
Membuat atau menghapus target mount di a VPC .....	166
VPCMengubah target mount Anda .....	168
Memperbarui konfigurasi target mount .....	168
Mengelola throughput .....	170
Mengelola siklus hidup penyimpanan untuk sistem file .....	171
Operasi sistem file untuk manajemen siklus hidup .....	172
Mengkonfigurasi kebijakan siklus hidup .....	173
Mengelola akses ke sistem file terenkripsi .....	175
Mengelola KMS kunci untuk sistem EFS file .....	176
Mengelola biaya sistem file .....	177
Prasyarat .....	178
Membuat anggaran biaya bulanan untuk sistem EFS file .....	178
Memahami status sistem berkas .....	179
Pemantauan .....	180
Alat pemantauan .....	181
Alat otomatis .....	181
Alat pemantauan manual .....	182
Pengukuran sistem file .....	182
Mengukur objek .....	182
Ukuran sistem file terukur .....	183

Throughput pengukuran .....	185
Melihat ukuran kelas penyimpanan .....	186
Memantau metrik dengan CloudWatch .....	187
CloudWatch metrik .....	188
Mengakses metrik CloudWatch .....	194
Menggunakan CloudWatch metrik .....	196
Menggunakan matematika metrik dengan CloudWatch metrik .....	197
Memantau keberhasilan dan kegagalan upaya pemasangan .....	203
Membuat alarm .....	205
Pencatatan API panggilan dengan CloudTrail .....	206
EFSInformasi Amazon di CloudTrail .....	207
Memahami entri file EFS log Amazon .....	208
Entri file EFS log Amazon untuk sistem encrypted-at-rest file .....	215
Kinerja .....	216
Ringkasan kinerja .....	216
Kelas penyimpanan .....	218
Mode kinerja .....	218
Mode throughput .....	219
Memilih mode throughput .....	220
Throughput elastis .....	220
Throughput yang Disediakan .....	221
Pembatasan pada pengalihan throughput dan perubahan jumlah yang disediakan .....	223
Tips performa .....	224
Ukuran I/O rata-rata .....	224
Mengoptimalkan beban kerja yang menuntut throughput tinggi dan IOPS .....	224
Koneksi simultan .....	225
Model permintaan .....	225
NFSpengaturan pemasangan klien .....	225
Mengoptimalkan kinerja file kecil .....	226
Mengoptimalkan kinerja direktori .....	227
Mengoptimalkan ukuran NFS read_ahead_kb .....	227
Memecahkan masalah kinerja .....	228
Tidak dapat membuat sistem EFS berkas .....	229
Akses ditolak ke file yang diizinkan pada sistem NFS file .....	229
Kesalahan saat mengakses konsol Amazon EFS .....	229
EC2Instans Amazon hang .....	230

Aplikasi menulis sejumlah besar data hang .....	230
Kinerja buruk saat membuka banyak file secara paralel .....	231
NFSPengaturan khusus yang menyebabkan penundaan penulisan .....	231
Membuat backup dengan Oracle Recovery Manager lambat .....	232
Pemecahan masalah AMI dan masalah kernel .....	232
Tidak dapat chown .....	233
Sistem file terus melakukan operasi berulang kali karena bug klien .....	233
Klien menemui jalan buntu .....	233
Daftar file dalam direktori besar membutuhkan waktu lama .....	234
Melindungi data .....	235
Mencadangkan sistem file .....	235
Bagaimana AWS Backup bekerja dengan Amazon EFS .....	236
Diperlukan izin IAM .....	238
Kinerja Backup .....	238
Mengelola backup otomatis .....	238
Mereplikasi sistem file .....	240
Biaya .....	241
Performa replikasi .....	241
Diperlukan izin IAM .....	242
Mengkonfigurasi replikasi ke sistem file baru .....	242
Mengkonfigurasi replikasi ke sistem file yang ada .....	247
Melihat detail replikasi .....	250
Menghapus konfigurasi replikasi .....	254
Menggunakan replika .....	255
Mengamankan data .....	257
Mengkripsi data .....	258
AWS KMS .....	259
Mengkripsi data saat istirahat .....	262
Mengkripsi data dalam perjalanan .....	263
Enkripsi pemecahan masalah .....	265
Pengelolaan identitas dan akses .....	267
Audiens .....	268
Mengautentikasi dengan identitas .....	269
Mengelola akses menggunakan kebijakan .....	272
Bagaimana Amazon Elastic File System bekerja dengan IAM .....	275
Contoh kebijakan berbasis identitas .....	282



Contoh kebijakan berbasis sumber daya .....	287
AWS kebijakan terkelola .....	289
Menggunakan tag dengan Amazon EFS .....	296
Menggunakan peran tertaut layanan untuk Amazon EFS .....	300
Pemecahan Masalah .....	305
Mengontrol akses data sistem file .....	307
Kebijakan Sistem File Default .....	308
EFStindakan untuk klien .....	308
EFSkunci kondisi untuk klien .....	308
Contoh kebijakan sistem file .....	309
Mengontrol akses jaringan .....	309
Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target .....	310
Port sumber .....	311
Pertimbangan keamanan untuk akses jaringan .....	311
Bekerja dengan titik VPC akhir .....	312
NFS-level pengguna, grup, dan izin .....	314
Izin file dan direktori .....	315
Contoh kasus penggunaan dan izin sistem EFS file Amazon .....	316
Izin ID pengguna dan grup untuk file dan direktori dalam sistem file .....	317
Tidak ada perencanaan akar .....	318
Caching izin .....	321
Mengubah kepemilikan objek sistem file .....	321
EFStitik akses .....	321
Titik Akses .....	321
Menegakkan identitas pengguna .....	323
Menegakkan direktori root .....	323
Menggunakan titik akses dalam IAM kebijakan .....	325
Memblokir akses publik ke sistem EFS file .....	327
Memblokir akses publik dengan AWS Transfer Family .....	327
Arti “publik” .....	328
Validasi kepatuhan .....	330
Ketangguhan .....	331
Isolasi jaringan .....	332
Kuota .....	333
EFSKuota Amazon yang dapat Anda tingkatkan .....	333
Meminta peningkatan kuota .....	335

Kuota EFS sumber daya Amazon yang tidak dapat Anda ubah .....	335
Kuota untuk klien NFS .....	337
Kuota untuk sistem EFS file Amazon .....	338
Fitur NFSv4 0 dan 4.1 yang tidak didukung .....	339
Pertimbangan tambahan .....	340
Memecahkan masalah kesalahan operasi file yang terkait dengan kuota .....	341
Perintah gagal dengan kesalahan “Kuota disk terlampaui” .....	341
Perintah gagal dengan “kesalahan I/O” .....	342
Perintah gagal dengan kesalahan “Nama file terlalu panjang” .....	342
Perintah gagal dengan kesalahan “File tidak ditemukan” .....	342
Perintah gagal dengan kesalahan “Terlalu banyak tautan” .....	343
Perintah gagal dengan kesalahan “File terlalu besar” .....	343
API Amazon EFS .....	344
Titik akhir API REST .....	344
Versi API .....	345
Topik terkait .....	345
Bekerja dengan tingkat permintaan API kueri untuk Amazon EFS .....	345
Polling .....	346
Pemrosesan coba ulang atau batch .....	346
Menghitung interval tidur .....	346
Tindakan .....	346
CreateAccessPoint .....	348
CreateFileSystem .....	356
CreateMountTarget .....	372
CreateReplicationConfiguration .....	383
CreateTags .....	389
DeleteAccessPoint .....	392
DeleteFileSystem .....	394
DeleteFileSystemPolicy .....	398
DeleteMountTarget .....	401
DeleteReplicationConfiguration .....	404
DeleteTags .....	407
DescribeAccessPoints .....	410
DescribeAccountPreferences .....	415
DescribeBackupPolicy .....	418
DescribeFileSystemPolicy .....	421

DescribeFileSystems .....	425
DescribeLifecycleConfiguration .....	431
DescribeMountTargets .....	435
DescribeMountTargetSecurityGroups .....	441
DescribeReplicationConfigurations .....	445
DescribeTags .....	449
ListTagsForResource .....	454
ModifyMountTargetSecurityGroups .....	458
PutAccountPreferences .....	462
PutBackupPolicy .....	465
PutFileSystemPolicy .....	468
PutLifecycleConfiguration .....	474
TagResource .....	482
UntagResource .....	486
UpdateFileSystem .....	489
UpdateFileSystemProtection .....	497
Tipe Data .....	500
AccessPointDescription .....	502
BackupPolicy .....	505
CreationInfo .....	506
Destination .....	508
DestinationToCreate .....	510
FileSystemDescription .....	512
FileSystemProtectionDescription .....	517
FileSystemSize .....	518
LifecyclePolicy .....	520
MountTargetDescription .....	522
PosixUser .....	525
ReplicationConfigurationDescription .....	527
ResourceIdPreference .....	529
RootDirectory .....	530
Tag .....	532
Riwayat dokumen .....	533
.....	dlvii

# Apa itu Amazon Elastic File System?

Amazon Elastic File System (AmazonEFS) menyediakan penyimpanan file tanpa server dan sepenuhnya elastis sehingga Anda dapat berbagi data file tanpa menyediakan atau mengelola kapasitas dan kinerja 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 mengkonfigurasi 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 dapat diakses di sebagian besar jenis instans komputasi Amazon Web Services, termasuk Amazon, EC2 AmazonEKS, ECS Amazon 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. Wilayah AWS Menyimpan data di beberapa Availability Zone menyediakan ketersediaan berkelanjutan ke data, bahkan ketika satu atau beberapa Availability Zone dalam sebuah tidak Wilayah AWS 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 [EFS jenis sistem file](#).

Amazon EFS menyediakan throughput, IOPS, dan latensi rendah yang diperlukan untuk berbagai beban kerja. EFS sistem file dapat berkembang menjadi skala petabyte, mendorong tingkat throughput yang tinggi, dan memungkinkan akses paralel besar-besaran dari instance 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 EFS performa dan throughput, lihat [EFS Kinerja Amazon](#).

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 EFS file Amazon. Jika Anda melakukannya, semua data dan metadata Anda dienkripsi. Anda dapat mengaktifkan enkripsi saat transit saat Anda memasang sistem file. NFS akses klien dikendalikan EFS oleh AWS Identity and Access Management (IAM) kebijakan dan kebijakan keamanan jaringan, seperti grup keamanan. Lihat informasi selengkapnya di [Mengkripsi data di Amazon EFS](#), [Manajemen identitas dan akses untuk Amazon EFS](#), dan [Mengontrol akses jaringan ke sistem EFS file Amazon untuk NFS klien](#).

#### Note

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

## Apakah Anda pengguna Amazon EFS pertama kali?

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

1. Untuk ikhtisar EFS produk dan harga Amazon, lihat [Amazon EFS](#).
2. Untuk ikhtisar EFS teknis Amazon, lihat [Bagaimana Amazon EFS bekerja](#).
3. Coba [Memulai](#) latihannya.

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

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

## Bagaimana Amazon EFS bekerja

Amazon Elastic File System (EFS) menyediakan sistem file set-and-forget elastis yang sederhana, tanpa server. Dengan AmazonEFS, Anda dapat membuat sistem file, memasang sistem file pada EC2 instance Amazon, dan kemudian membaca dan menulis data ke dan dari sistem file Anda. Anda dapat memasang sistem EFS file Amazon 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, silakan lihat [Menginstal EFS klien Amazon](#).

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

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

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


Untuk mengakses sistem EFS file Amazon Anda di aVPC, Anda membuat satu atau beberapa target mount di fileVPC.

- Untuk sistem file Regional, Anda dapat membuat target mount di setiap Availability Zone di file Wilayah AWS.

- Untuk sistem file One Zone, Anda hanya membuat satu target mount yang berada di Availability Zone yang sama dengan sistem file.

Untuk informasi selengkapnya, lihat [EFS kelas penyimpanan](#).

Target mount menyediakan alamat IP untuk NFSv4 titik akhir di mana Anda dapat memasang sistem EFS file Amazon. Anda memasang sistem file Anda menggunakan nama Domain Name Service (DNS), yang menyelesaikan ke alamat IP target EFS mount di Availability Zone yang sama dengan instance Anda EC2. Anda dapat membuat satu target mount di setiap Availability Zone di file Wilayah AWS. Jika ada beberapa subnet di Availability Zone di Anda VPC, Anda membuat target mount di salah satu subnet. Kemudian semua EC2 instance di Availability Zone tersebut membagikan target mount tersebut.

 Note

Sistem EFS file Amazon dapat memiliki target mount hanya dalam satu VPC per satu.

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 target mount Anda di setiap Availability Zone bersifat statis, mereka adalah komponen redundan yang didukung oleh beberapa sumber daya.

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

Anda dapat memasang sistem EFS file Amazon di server pusat data lokal saat tersambung ke Amazon VPC dengan AWS Direct Connect atau AWS VPN. Anda dapat memasang sistem EFS file di server lokal untuk memigrasikan kumpulan data 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](#)
- [Bagaimana Amazon EFS bekerja dengan AWS Direct Connect dan AWS Dikelola VPN](#)

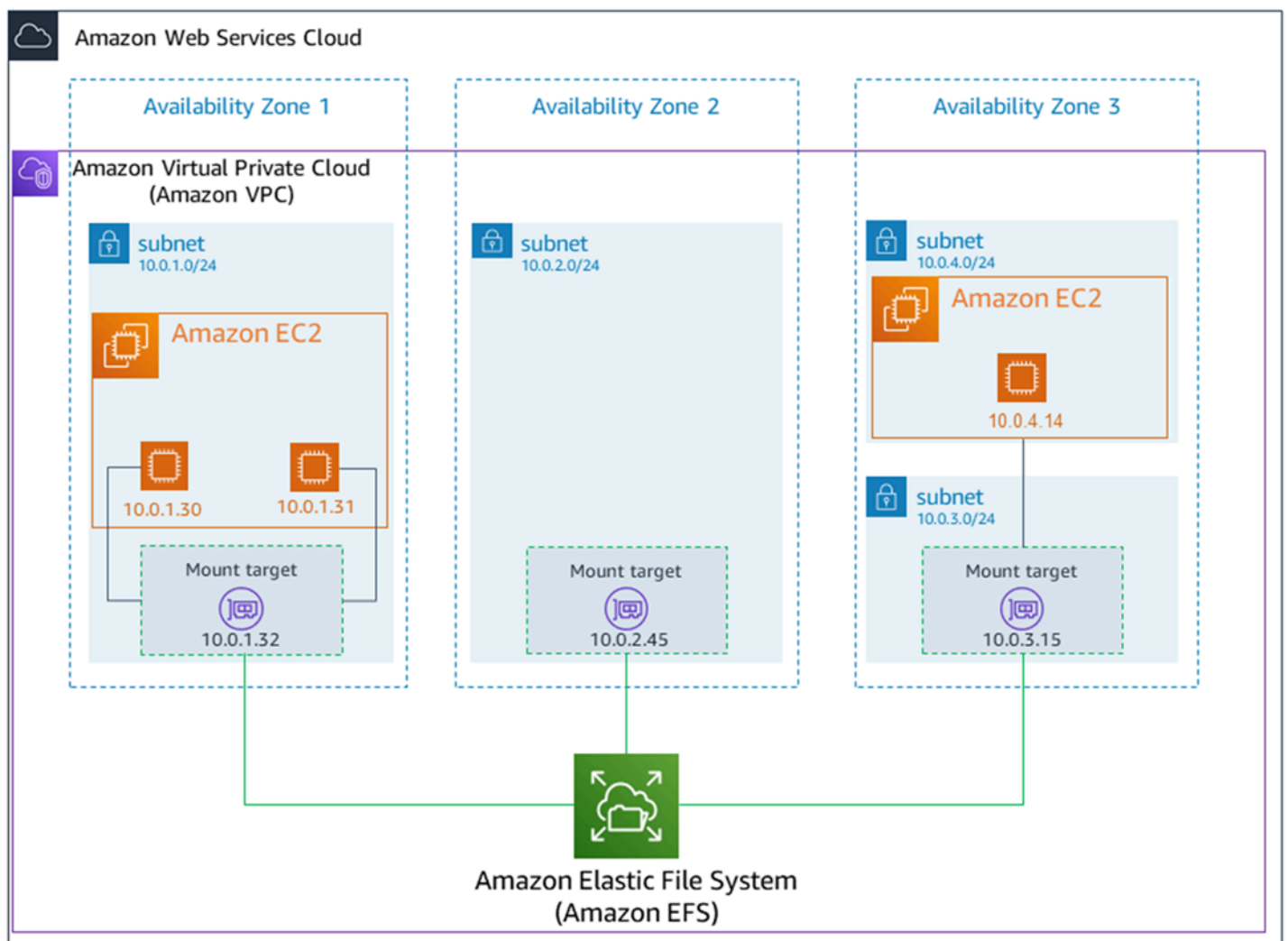
- [Bagaimana Amazon EFS bekerja dengan AWS Backup](#)

## Bagaimana Amazon EFS bekerja dengan Amazon EC2

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

### Sistem EFS file regional

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



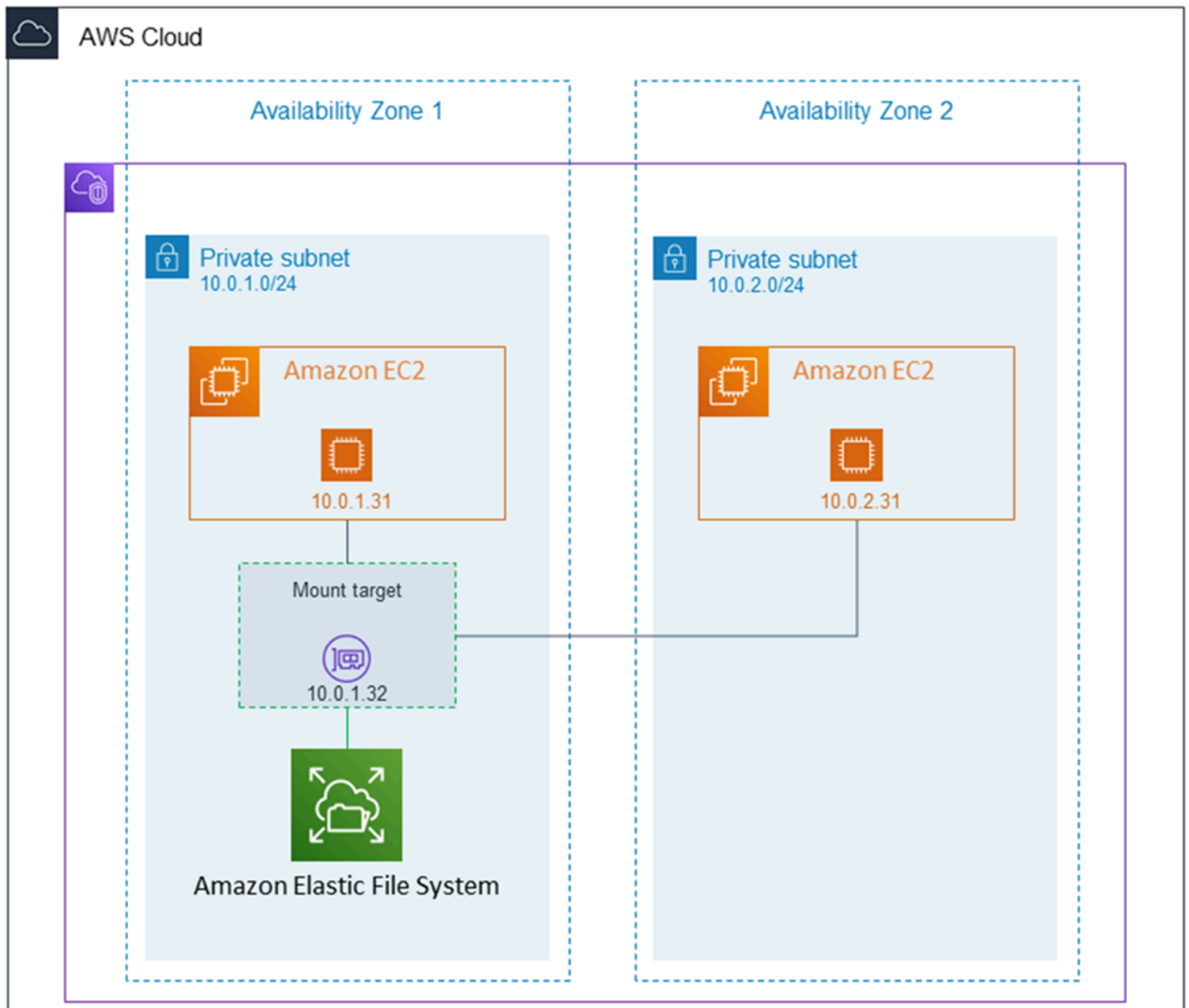
Dalam ilustrasi ini, virtual private cloud (VPC) memiliki tiga Availability Zones. Karena sistem file Regional, target mount dibuat di setiap Availability Zone. Kami menyarankan Anda mengakses sistem file dari target pemasangan dalam Availability Zone yang sama untuk alasan kinerja dan biaya. Salah



satu Availability Zones memiliki dua subnet. Namun, target mount dibuat hanya di salah satu subnet. Untuk informasi selengkapnya, lihat [Memasang sistem EFS file menggunakan EFS mount helper](#).

## Sistem EFS file One Zone

Ilustrasi berikut menunjukkan beberapa EC2 contoh mengakses sistem file One Zone dari Availability Zone yang berbeda dalam satu Wilayah AWS



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

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

## Bagaimana Amazon EFS bekerja dengan AWS Direct Connect dan AWS Dikelola VPN

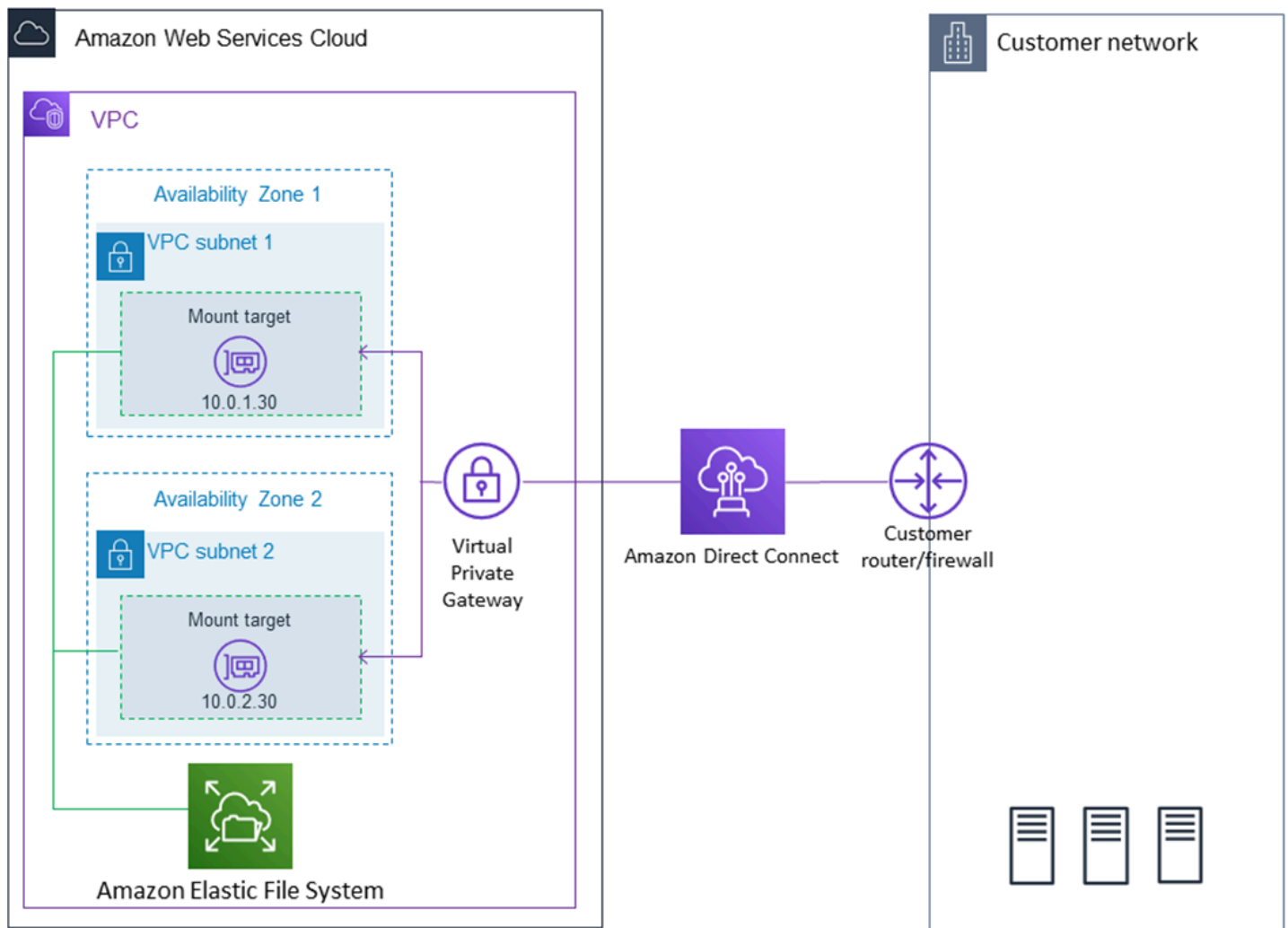
Dengan menggunakan sistem EFS file Amazon yang dipasang di server lokal, Anda dapat memigrasikan data lokal ke AWS Cloud host di sistem file AmazonEFS. Anda juga bisa memanfaatkan meledak. Dengan kata lain, Anda dapat memindahkan data dari server lokal ke Amazon EFS dan menganalisisnya pada armada EC2 instans Amazon di Amazon Anda. VPC 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 EFS file Amazon di server lokal menggunakan alamat IP target mount, bukan DNS nama.

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

Ilustrasi berikut menunjukkan contoh cara mengakses sistem EFS file Amazon 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 AWS 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 NFS port (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 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 EFS file yang digunakan AWS Backup, lihat [Mencadangkan sistem EFS file](#).

## Fitur Amazon EFS

Berikut ini adalah fitur AmazonEFS.

Topik

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

### Kontrol autentikasi dan akses

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

Pengguna dan peran yang Anda buat di 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](#).

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

### Konsistensi data di Amazon EFS

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

Di AmazonEFS, 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.

Tergantung pada pola akses, 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

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

## Ketersediaan dan daya tahan sistem EFS file

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

### EFS jenis sistem file

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. Wilayah AWS Menyimpan data di beberapa Availability Zone menyediakan ketersediaan berkelanjutan ke data, bahkan ketika satu atau beberapa Availability Zone dalam sebuah tidak Wilayah AWS 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 Wilayah AWS, atau Anda dapat mengembalikannya ke yang lain. Wilayah AWS EFSbackup sistem file yang dibuat dan dikelola menggunakan AWS Backup direplikasi ke tiga Availability Zones 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.

Jenis 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

Jenis sistem file	Dirancang untuk	Ketahanan (dirancang untuk)	Ketersediaan	Zona Ketersediaan	Pertimbangan lainnya
Satu Zona	Data yang tidak membutuhkan 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 Wilayah AWS 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

Wilayah AWS Nama	Wilayah AWS 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
US West (Northern California)	us-west-1	usw1-az1, usw1-az3
AS Barat (Oregon)	us-west-2	usw2-az1, usw2-az2, usw2-az3, usw2-az4
Afrika (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

Wilayah AWS Nama	Wilayah AWS 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
Eropa (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



Wilayah AWS Nama	Wilayah AWS 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

## EFS kelas penyimpanan

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

- **EFS Standar** — Kelas penyimpanan EFS Standar 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 Standar dan kemudian dapat berjenjang ke kelas penyimpanan Akses dan EFS Arsip Jarang dengan menggunakan manajemen siklus hidup EFS
- **EFS Akses Jarang (IA)** — Kelas penyimpanan yang dioptimalkan biaya untuk data yang diakses hanya beberapa kali setiap kuartal.
- **EFS Arsip** — Kelas penyimpanan yang dioptimalkan biaya untuk data yang diakses beberapa kali setiap tahun atau kurang.

Kelas penyimpanan EFS Arsip didukung pada sistem EFS file 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 [EFS Kinerja Amazon](#).

Kelas penyimpanan	Dirancang untuk	Latensi baca byte pertama	Daya tahan (dirancang untuk) <sup>1</sup>	Ketersediaan SLA	Zona ketersediaan	Biaya penagihan minimum per file <sup>2</sup>	Durasi penyimpanan minimum
EFSStandar	Data aktif yang membutuhkan kinerja latensi sub-milidetik yang cepat	Sub-milidetik		99,99% (Regional)		Tidak berlaku	Tidak berlaku
EFSAkses Jarang	Data tidak aktif yang diakses hanya beberapa kali setiap kuartal.	Puluhan milidetik	99,9999999 (11 9)	99,9% (Satu Zona)	=>3 (Regional)	128 KiB	Tidak berlaku
EFSArsip	Data tidak aktif yang diakses beberapa kali setiap tahun atau kurang	Puluhan milidetik		99,9% (Regional)	=>3 (Regional)	128 KiB	90 hari

### Note

<sup>1</sup> 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.

<sup>2</sup> Kebijakan Siklus Hidup yang diperbarui pada atau setelah pukul 12 PM PT, 26 November 2023 akan memberi peringkat file < 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](#)

## Harga 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 [EFS Harga Amazon](#).

Selain itu, kelas penyimpanan Infrequent Access (IA) dan Arsip memiliki biaya penagihan minimum per file 128 KiB. Dukungan untuk file yang lebih kecil dari 128 KiB hanya tersedia untuk kebijakan siklus hidup yang diperbarui pada atau setelah pukul 12:00 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](#)

Harga tambahan berlaku untuk sistem file yang menggunakan throughput Provisioned atau Bursting.

- 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 Standar. EFS
- Untuk sistem file yang menggunakan throughput Bursting, throughput yang diizinkan ditentukan berdasarkan jumlah data yang disimpan di kelas penyimpanan EFS Standar saja.

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

### Note

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

## 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 untuk sistem file EFS](#).

## Replikasi

Anda dapat membuat replika sistem EFS file Amazon Anda sesuai preferensi Anda menggunakan replikasi. Wilayah AWS Replikasi secara otomatis dan transparan mereplikasi data dan metadata pada sistem EFS file Anda ke sistem file tujuan baru yang dibuat dalam EFS file yang Anda pilih. Wilayah AWS EFS secara otomatis menjaga sistem file sumber dan tujuan 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. Untuk informasi selengkapnya, lihat [Mereplikasi sistem EFS file](#).

# Memulai dengan Amazon EFS

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

1. [Tinjau prasyarat untuk memulai](#)
2. [Buat sistem EFS file Anda dan luncurkan EC2 instance Anda](#)
3. [Transfer file ke sistem EFS file 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 dengan peluncuran EC2 instance. Anda memerlukan Akun AWS, pengguna dengan akses administratif, key pair, dan grup keamanan. Untuk informasi selengkapnya, lihat [Mengatur untuk menggunakan Amazon EC2](#).
- Amazon Virtual Private Cloud (AmazonVPC)EC2,, dan EFS sumber daya semuanya sama Wilayah AWS dan Anda memiliki default VPC di Wilayah. Jika Anda tidak memiliki defaultVPC, atau jika Anda ingin me-mount sistem file Anda dari yang baru VPC dengan grup keamanan baru atau yang sudah ada, lihat[Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target](#).
- 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 EFS API panggilan. Untuk informasi selengkapnya, lihat [Tutorial: Buat sistem EFS file dan pasang pada EC2 instance menggunakan AWS CLI](#).

## Buat sistem EFS file Anda dan luncurkan EC2 instance Anda

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

**Note**

Anda tidak dapat menggunakan Amazon EFS dengan berbasis Microsoft Windows.  
EC2instances

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

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

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

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

Atau, di EFS konsol Amazon, 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 EFS file](#).

## Transfer file ke sistem EFS file Anda menggunakan AWS DataSync

Setelah membuat sistem EFS file, 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. AWS 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 EFS file, pastikan Anda memiliki yang berikut:

- Sistem NFS file 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 EFS sistem file.
- Anda diatur untuk digunakan DataSync. Untuk mempelajari lebih lanjut, lihat [Menyiapkan dengan AWS DataSync](#) di Panduan AWS DataSync Pengguna.

Untuk mentransfer file ke sistem EFS file Anda menggunakan AWS DataSync

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

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

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

```
$ sudo umount efs
```

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



# Membuat dan mengelola EFS sumber daya

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

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

Misalnya, anggaplah Anda memiliki satu atau lebih EC2 instance yang diluncurkan di situs AndaVPC. Sekarang Anda ingin membuat dan menggunakan sistem file pada contoh ini. Berikut ini adalah langkah-langkah umum yang harus Anda lakukan untuk menggunakan sistem EFS file Amazon diVPC:

- Buat sistem EFS file Amazon — 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 sistem Anda VPC dan memasang sistem file ke EC2 instans Amazon Anda, Anda harus membuat target mount di VPC subnet.
- Buat grup keamanan — EC2 Instans Amazon 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 EC2 instance tertentu. Kemudian, Anda dapat me-mount sistem file hanya pada EC2 instance itu.

## Topik

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

- [Membuat grup keamanan](#)
- [Membuat kebijakan sistem file](#)
- [Membuat titik akses](#)
- [Menghapus titik akses](#)
- [Sumber daya penandaan EFS](#)
- [Tutorial: Membuat subdirektori per pengguna yang dapat ditulis](#)

## Ringkasan implementasi

Di AmazonEFS, 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 file AndaVPC. Setiap target mount memiliki properti berikut: ID target mount, ID subnet di mana ia dibuat, ID sistem file yang membuatnya, alamat IP di mana sistem file dapat dipasang, grup VPC keamanan, dan status target mount. Anda dapat menggunakan alamat IP atau DNS nama dalam mount perintah Anda.

Setiap sistem file memiliki DNS nama formulir berikut.

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

Anda dapat menentukan DNS nama ini dalam mount perintah Anda untuk me-mount sistem EFS file Amazon. Misalkan Anda membuat `efs-mount-point` subdirektori dari direktori home di server EC2 instans atau lokal. Kemudian, Anda dapat menggunakan perintah mount untuk me-mount sistem file. Misalnya, di Amazon LinuxAMI, 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 diberikan oleh NFS klien. 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 EFS akses Amazon](#).

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

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

- Gunakan EFS konsol Amazon — Sebagai contoh, lihat [Memulai](#).
- Gunakan antarmuka baris EFS perintah Amazon (CLI) — Sebagai contoh, lihat [Tutorial: Buat sistem EFS file dan pasang pada EC2 instance menggunakan AWS CLI](#).
- Anda juga dapat mengelola sumber daya ini secara terprogram sebagai berikut:
  - Gunakan AWS SDKs — AWS SDKs Menyederhanakan tugas pemrograman Anda dengan membungkus Amazon yang mendasarinya. EFS API SDKKlien 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 EFS API panggilan Amazon langsung dari aplikasi Anda. Namun, Anda perlu menulis kode yang diperlukan untuk mengautentikasi permintaan Anda jika Anda menggunakan opsi ini. Untuk informasi lebih lanjut tentang Amazon EFSAPI, lihat [API Amazon EFS](#).

## Sumber Daya IDs

Amazon EFS menetapkan pengidentifikasi sumber daya unik (IDs) ke semua EFS sumber daya saat dibuat. Semua EFS sumber daya IDs terdiri dari pengenal 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 singkatIDs, meskipun sumber daya baru dari jenis ini menerima lebih lamaIDs. ID Sumber Daya tidak pernah berubah.

## Token pembuatan dan idempotensi

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

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 ASCII karakter. Jika Anda menggunakan kembali token permintaan klien dalam waktu satu menit setelah permintaan berhasil, maka akan API mengembalikan rincian 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 EFS file

Berikut ini, Anda dapat mempelajari cara membuat sistem EFS file Amazon dengan menggunakan file AWS Management Console dan file AWS CLI.

Topik

- [IAMizin yang diperlukan untuk membuat sistem file](#)

- [Opsi konfigurasi untuk sistem file](#)

## IAM izin yang diperlukan untuk membuat sistem file

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

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

## Opsi konfigurasi untuk sistem file

Anda dapat membuat sistem file dengan menggunakan EFS konsol Amazon atau dengan menggunakan AWS Command Line Interface (AWS CLI). Anda juga dapat membuat sistem file secara terprogram dengan menggunakan AWS SDKs atau Amazon EFS API secara langsung. Jika Anda menggunakan Amazon EFS API atau Amazon AWS SDK, Anda dapat menggunakan `CreateFileSystem` EFS API tindakan untuk membuat kebijakan sistem file.

Saat membuat sistem EFS file Amazon dengan menggunakan alur pembuatan kustom di konsol atau AWS CLI, Anda dapat memilih pengaturan untuk fitur sistem file berikut dan opsi konfigurasi.

## Jenis sistem file

Jenis sistem file menentukan [ketersediaan dan daya tahan](#) dengan mana sistem EFS file Amazon menyimpan data dalam file file Wilayah AWS. Anda memiliki pilihan berikut untuk jenis sistem file Anda:

- Pilih Regional untuk membuat sistem file yang menyimpan data dan metadata secara berlebihan di semua Availability Zone dalam file. Wilayah AWS Anda juga dapat membuat target mount di setiap Availability Zone di Wilayah AWS. Regional menawarkan tingkat ketersediaan dan daya tahan tertinggi.
- Pilih One Zone untuk membuat sistem file yang menyimpan data dan metadata secara berlebihan dalam satu Availability Zone. Sistem file yang menggunakan jenis 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. 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](#).

## Pencadangan otomatis

Pencadangan otomatis selalu diaktifkan secara default saat Anda membuat sistem file dengan menggunakan konsol. Ketika Anda menggunakan CLI atau API untuk membuat sistem file, backup otomatis diaktifkan secara default hanya ketika Anda membuat sistem file yang menggunakan sistem file One Zone. Untuk informasi selengkapnya, lihat [Mengelola backup otomatis sistem file EFS](#).

## Kebijakan siklus hidup

Manajemen siklus hidup menggunakan kebijakan siklus hidup untuk secara otomatis memindahkan file masuk dan keluar dari kelas penyimpanan Akses Jarang (IA) berbiaya rendah berdasarkan pola akses. Saat Anda membuat sistem file menggunakan AWS Management Console, kebijakan siklus hidup sistem file dikonfigurasi dengan setelan default berikut:

- Transisi ke IA diatur ke 30 hari sejak akses terakhir.
- TransitionToArchived diatur ke 90 hari sejak akses terakhir.
- Transisi ke Standar diatur ke Tidak Ada.

Bila Anda membuat sistem file menggunakan Amazon AWS CLI, atau EFS API AWS SDKs, Anda tidak dapat menetapkan kebijakan siklus hidup secara bersamaan. Anda harus menunggu sampai sistem file dibuat, dan kemudian menggunakan [PutLifecycleConfiguration](#) API operasi untuk memperbarui kebijakan siklus hidup. Untuk informasi selengkapnya, silakan lihat [EFS kelas penyimpanan](#) dan [Mengelola siklus hidup penyimpanan untuk sistem file EFS](#).

## Enkripsi

Anda dapat mengaktifkan enkripsi saat istirahat saat membuat sistem file. Jika Anda mengaktifkan enkripsi saat istirahat untuk sistem file Anda, semua data dan metadata yang tersimpan di dalamnya dienkripsi. Anda dapat mengaktifkan enkripsi saat transit nanti, saat Anda memasang sistem file. Untuk informasi selengkapnya tentang EFS enkripsi Amazon, lihat [Mengenkripsi data di Amazon EFS](#).

Untuk membuat target pemasangan sistem file di AndaVPC, Anda harus menentukan VPC subnet. Konsol mengisi terlebih dahulu daftar VPCs di akun Anda yang ada di yang dipilih. Wilayah AWS Pertama, Anda memilihVPC, dan kemudian konsol mencantumkan Availability Zones diVPC. Untuk setiap Availability Zone, Anda dapat memilih subnet dari daftar, atau menggunakan subnet default jika ada. Setelah memilih subnet, Anda dapat menentukan alamat IP yang tersedia di subnet atau membiarkan Amazon EFS memilih alamat secara otomatis.

## Mode throughput

Ada tiga mode throughput untuk dipilih:

- Elastis (Direkomendasikan) - Menyediakan throughput yang menskalakan naik dan turun secara otomatis secara real time, untuk memenuhi kebutuhan kinerja beban kerja Anda.

### Note

Throughput elastis hanya tersedia untuk sistem file yang memiliki mode kinerja Tujuan Umum.

- Provisioned — Menyediakan tingkat throughput yang Anda tentukan, terlepas dari ukuran sistem file.
- Bursting — Menyediakan throughput yang menskalakan dengan jumlah data dalam penyimpanan Standar.

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

### Note

Biaya tambahan terkait dengan penggunaan throughput Elastic dan Provisioned. Untuk informasi selengkapnya, lihat [EFSharga Amazon](#).

## Mode kinerja

Saat membuat sistem file, Anda juga memilih mode kinerja. Ada dua mode yang bisa dipilih yaitu General Purpose dan Max I/O.

- Mode 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. Mode Max I/O 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](#).

Cepat membuat sistem file yang memiliki pengaturan yang direkomendasikan (konsol)

Pada langkah ini, gunakan EFS konsol Amazon untuk membuat sistem EFS file Amazon yang memiliki pengaturan yang disarankan. Jika Anda ingin membuat sistem file dengan konfigurasi yang disesuaikan, lihat [Buat sistem file dengan pengaturan khusus \(konsol\)](#).

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

1. Masuk ke AWS Management Console dan buka EFS konsol Amazon 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)VPC, pilih, atau tetapkan ke default AndaVPC.
5. Pilih Buat untuk membuat sistem file yang menggunakan pengaturan yang direkomendasikan layanan berikut:
  - Pencadangan otomatis diaktifkan. Untuk informasi selengkapnya, lihat [Mencadangkan sistem EFS file](#).
  - Target pemasangan yang dikonfigurasi dengan pengaturan berikut:
    - Dibuat di setiap Availability Zone Wilayah AWS di mana sistem file dibuat.
    - Terletak di subnet default yang VPC Anda pilih.
    - Menggunakan grup keamanan default — Anda dapat mengelola grup keamanan setelah sistem file dibuat. VPC

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



- Jenis sistem file regional - Untuk informasi lebih lanjut, lihat [EFS jenis sistem file](#).
- Kinerja Tujuan Umum - Untuk informasi lebih lanjut, lihat [Mode kinerja](#).
- Throughput elastis — Untuk informasi lebih lanjut, lihat [Mode throughput](#).
- Enkripsi data saat istirahat diaktifkan menggunakan kunci default untuk Amazon EFS (aws/elasticfilesystem) — Untuk informasi selengkapnya, lihat [Mengenkripsi data saat istirahat](#).
- Manajemen siklus hidup - Amazon EFS membuat sistem file dengan kebijakan siklus hidup berikut:
  - Transisi ke IA diatur ke 30 hari sejak akses terakhir.
  - TransitionToArchived diatur ke 90 hari sejak akses terakhir.
  - Transisi ke Standar diatur ke Tidak Ada.

Untuk informasi selengkapnya, lihat [Mengelola siklus hidup penyimpanan untuk sistem file EFS](#).

Setelah Anda membuat sistem file, Anda dapat menyesuaikan pengaturan sistem file dengan pengecualian ketersediaan dan daya tahan, enkripsi, dan mode kinerja.

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 sistem file dengan pengaturan khusus (konsol)

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

Membuat sistem EFS file 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 NFS klien 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 AWS Management Console dan buka EFS konsol Amazon 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 direkomendasikan layanan. Halaman pengaturan sistem file terbuka.
4. Untuk pengaturan Umum, lakukan hal berikut.
  - a. (Opsional) Masukkan Nama untuk sistem file.
  - b. Untuk jenis sistem File, pilih opsi ketersediaan:
    - Pilih Regional untuk membuat sistem file yang menyimpan data sistem file dan metadata secara berlebihan di semua Availability Zone dalam file. Wilayah AWSRegional menawarkan tingkat ketersediaan dan daya tahan tertinggi.
    - Pilih One Zone untuk 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. Untuk informasi selengkapnya, lihat [EFS kelas penyimpanan](#).
  - c. Pencadangan otomatis dihidupkan secara default. Anda dapat mematikan cadangan otomatis dengan membersihkan kotak centang. Untuk informasi selengkapnya, lihat [Mencadangkan sistem EFS file](#).
  - d. 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 untuk sistem file EFS](#)

- e. Untuk Enkripsi, enkripsi data saat istirahat diaktifkan secara default. Amazon EFS menggunakan AWS Key Management Service (AWS KMS) EFS service key (aws/elasticfilesystem) Anda secara default. Untuk memilih KMS kunci lain yang akan digunakan untuk enkripsi, perluas Sesuaikan pengaturan enkripsi dan pilih kunci dari daftar. Atau, masukkan ID KMS kunci atau Amazon Resource Name (ARN) untuk KMS kunci 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.

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.

Sistem EFS file Amazon mengukur permintaan baca sepertiga tingkat permintaan lainnya. Setelah Anda memasuki 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 target VPC dan mount.

1. Pilih Virtual Private Cloud (VPC) tempat Anda ingin EC2 instance 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, tetapkan properti berikut:
  - Availability Zone — Secara default, target mount dikonfigurasi di setiap Availability Zone dalam file Wilayah AWS. 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.
  - Alamat IP — Secara default, Amazon EFS memilih alamat IP secara otomatis dari alamat yang tersedia di subnet. Atau, Anda dapat memasukkan alamat IP tertentu yang ada di subnet. Meskipun target mount memiliki satu alamat IP, mereka adalah sumber daya jaringan yang berlebihan dan sangat tersedia.
  - Grup keamanan - Anda dapat menentukan satu atau beberapa grup keamanan untuk target pemasangan. Untuk informasi selengkapnya, lihat [Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target](#).

Untuk menambahkan grup keamanan lain, atau mengubah grup keamanan, pilih Pilih grup keamanan dan tambahkan grup keamanan lain dari daftar. Jika Anda tidak ingin menggunakan grup keamanan default, Anda dapat menghapusnya. Untuk informasi selengkapnya, lihat [Membuat grup keamanan](#).

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

1. Di opsi Kebijakan, Anda dapat memilih kombinasi apa pun dari kebijakan yang telah dikonfigurasi sebelumnya:
  - Cegah akses root secara default
  - Menerapkan akses hanya-baca secara default
  - Menerapkan enkripsi in-transit untuk semua klien
2. Gunakan editor Kebijakan untuk menyesuaikan kebijakan yang telah dikonfigurasi sebelumnya atau untuk membuat kebijakan Anda sendiri. Bila Anda memilih salah satu kebijakan yang telah dikonfigurasi sebelumnya, definisi JSON kebijakan akan muncul di editor kebijakan. Anda dapat mengedit JSON untuk membuat kebijakan pilihan Anda. Untuk membatalkan perubahan Anda, pilih Hapus.

Kebijakan yang telah dikonfigurasi menjadi tersedia sekali lagi di opsi Kebijakan.

3. 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 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 sistem file (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 kredensi Anda. Untuk selengkapnya, lihat [Prasyarat untuk menggunakan AWS CLI dalam Panduan Pengguna](#). AWS Command Line Interface

- Untuk membuat sistem file terenkripsi yang menggunakan kelas penyimpanan EFS Arsip, dengan pencadangan otomatis diaktifkan, gunakan `efs create-file-system` CLI perintah Amazon (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 bursting \  
--region aws-region \  
--tags Key=key,Value=value Key=key1,Value=value1 \  
--profile adminuser
```

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

```
aws efs create-file-system \  
--creation-token MyFirstFS \  
--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 sepertiJSON, 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": "bursting",  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "Test File System"  
    }  
  ]  
}
```

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

```
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 sepertiJSON, 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 `describe-file-systems` CLI perintah (API operasi yang sesuai adalah [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 EFS file

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.

## Hapus sistem file (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.

## Hapus sistem file (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 grup keamanan

Baik EC2 instans Amazon dan target mount memiliki grup keamanan terkait. Kelompok keamanan ini bertindak sebagai firewall virtual yang mengontrol lalu lintas di antara mereka. Jika Anda tidak menyediakan grup keamanan saat membuat target pemasangan, Amazon EFS mengaitkan grup keamanan default VPC dengannya.

Terlepas dari itu, untuk mengaktifkan lalu lintas antara EC2 instance dan target mount (dan dengan demikian sistem file), Anda harus mengonfigurasi aturan berikut dalam grup keamanan ini:

- Grup keamanan yang Anda kaitkan dengan target mount harus mengizinkan akses masuk untuk TCP protokol pada NFS port dari semua EC2 instance tempat Anda ingin memasang sistem file.
- Setiap EC2 instance yang memasang sistem file harus memiliki grup keamanan yang memungkinkan akses keluar ke target mount pada NFS port.

Untuk mengubah grup keamanan yang terkait dengan target pemasangan sistem EFS file Anda, lihat [Mengelola target mount](#).

Untuk informasi selengkapnya tentang grup keamanan, lihat [Grup EC2 keamanan Amazon untuk instans Linux](#) di Panduan EC2 Pengguna Amazon.

### Note

Bagian berikut ini khusus untuk Amazon EC2 dan membahas cara membuat grup keamanan sehingga Anda dapat menggunakan Secure Shell (SSH) untuk terhubung ke instans apa pun yang telah memasang sistem EFS file Amazon. Jika Anda tidak menggunakan SSH untuk terhubung ke EC2 instans Amazon Anda, Anda dapat melewati bagian ini.

## Buat grup keamanan (konsol)

Anda dapat menggunakan AWS Management Console untuk membuat grup keamanan di AndaVPC. Untuk menghubungkan sistem EFS file Amazon Anda ke EC2 instans Amazon Anda, Anda harus membuat dua grup keamanan: satu untuk EC2 instans Amazon Anda dan satu lagi untuk target EFS pemasangan Amazon Anda.

1. Buat dua grup keamanan di AndaVPC. Untuk petunjuknya, lihat [Membuat grup keamanan](#) di Panduan VPC Pengguna Amazon.
2. Di VPC konsol, verifikasi aturan default untuk grup keamanan ini. Kedua kelompok keamanan seharusnya hanya memiliki aturan keluar yang memungkinkan lalu lintas pergi.
3. Anda harus mengotorisasi akses tambahan ke grup keamanan sebagai berikut:
  - a. Tambahkan aturan ke grup EC2 keamanan untuk mengizinkan SSH akses ke instance pada port 22 seperti yang ditunjukkan berikut. Ini berguna jika Anda berencana menggunakan SSH klien yang PuTTY ingin terhubung dan mengelola EC2 instance Anda melalui antarmuka terminal. Secara opsional, Anda dapat membatasi alamat Sumber.

Untuk petunjuknya, lihat [Menambahkan aturan ke grup keamanan](#) di Panduan VPC Pengguna Amazon.

- b. Tambahkan aturan ke grup keamanan target mount untuk mengizinkan akses masuk dari grup EC2 keamanan pada TCP port 2049. Grup keamanan yang ditetapkan sebagai Sumber adalah grup keamanan yang terkait dengan EC2 instance.

Untuk melihat grup keamanan yang terkait dengan target pemasangan sistem file Anda, di EFS konsol, pilih tab Jaringan di halaman Detail sistem file. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).

### Note

Anda tidak perlu menambahkan aturan keluar karena aturan keluar default memungkinkan semua lalu lintas untuk pergi. (Jika Anda menghapus aturan keluar default, Anda harus menambahkan aturan keluar untuk membuka TCP koneksi pada NFS port, dan mengidentifikasi grup keamanan target mount sebagai tujuan.)

4. Verifikasi bahwa kedua grup keamanan sekarang mengotorisasi akses masuk dan keluar seperti yang dijelaskan di bagian ini.

## Buat grup keamanan (AWS CLI)

Untuk contoh yang menunjukkan cara membuat grup keamanan menggunakan AWS CLI, lihat [Langkah 1: Buat EC2 sumber daya](#).

## Membuat kebijakan sistem file

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

### Note

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

## Buat kebijakan sistem file (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 EFS tindakan yang diizinkan.
  - Menerapkan akses hanya-baca secara default — Opsi ini menghapus `ClientWriteAccess` dari kumpulan tindakan yang diizinkan. EFS
  - Mencegah akses anonim — Opsi ini menghapus `ClientMount` dari kumpulan EFS tindakan 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, JSON objek kebijakan akan ditampilkan di panel Editor kebijakan.

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

- Setelah Anda menyelesaikan pengeditan kebijakan, pilih Simpan.

## Buat kebijakan sistem file (AWS CLI)

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

```
{
```

```
"FileSystemId": "fs-01234567",
"Policy": "{
"Version" : "2012-10-17",
"Id" : "1",
"Statement" : [
  {
    "Sid" : "efs-statement-7c8d8687-1c94-4fdc-98b7-555555555555",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : [
      "elasticfilesystem:ClientMount"
    ],
    "Resource" : "arn:aws:elasticfilesystem:us-east-2:555555555555:file-system/
fs-01234567"
  }
]
}
}
```

## Membuat titik akses

Anda tidak dapat memodifikasi titik akses setelah dibuat. Sistem file dapat memiliki maksimum 1.000 titik akses. Untuk informasi selengkapnya tentang titik EFS akses, lihat [Bekerja dengan titik EFS akses Amazon](#).

### Buat titik akses (konsol)

Anda dapat membuat dan menghapus EFS jalur akses Amazon menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), dan Amazon EFS API dan SDKs. Anda tidak dapat memodifikasi titik akses setelah dibuat. Sistem file dapat memiliki maksimum 1.000 titik akses.

#### Note

Jika beberapa permintaan untuk membuat titik akses pada sistem file yang sama dikirim secara berurutan, dan sistem file mendekati batas 1.000 titik akses, Anda mungkin mengalami respons pelambatan untuk permintaan ini. Hal ini untuk memastikan bahwa sistem berkas tidak melebihi kuota titik akses yang disebutkan.

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 POSIX pengguna, Anda dapat menentukan POSIX identitas lengkap yang akan digunakan untuk menegakkan informasi pengguna dan grup untuk semua operasi file oleh NFS klien yang menggunakan titik akses. Untuk informasi selengkapnya, lihat [Menegakkan identitas pengguna menggunakan titik akses](#).

- User ID — Masukkan ID POSIX pengguna numerik untuk pengguna.
- ID Grup - Masukkan ID POSIX grup 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, dan direktori root belum ada, tidak EFS akan membuat direktori root. Setiap upaya untuk me-mount sistem file dengan menggunakan titik akses akan gagal.

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

## Buat titik akses (CLI)

Dalam contoh berikut, `create-access-point` CLI perintah membuat titik akses untuk sistem EFS file. API Perintah 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 1.000 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 EFS file Amazon yang dikonfigurasi untuknya.

### Hapus titik akses (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.

### Hapus titik akses (AWS CLI)

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

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

# Sumber daya penandaan EFS

Untuk membantu mengelola EFS sumber daya, 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 tag

Tag adalah label yang Anda tetapkan ke AWS sumber daya. Setiap tag terdiri dari kunci dan nilai opsional, yang keduanya Anda tentukan.

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 EFS file Amazon 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 ke 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.

## Batasan 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 memiliki satu nilai.

- Panjang kunci maksimum - 128 karakter Unicode di 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 diizinkan di seluruh layanan adalah: huruf, angka, dan spasi yang dapat direpresentasikan dalam UTF -8, dan karakter berikut: + - =. \_:/@.
- Kunci dan nilai tag peka 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 pengenal 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 orang lain 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 EFS file Amazon dan sumber daya titik akses.

## Menggunakan tag untuk kontrol akses

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

### Note

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

## Tandai sumber daya Anda

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

## Menandai sistem file atau sumber daya titik akses (konsol)

- Anda dapat menggunakan EFS konsol Amazon untuk menerapkan tag ke sumber daya yang ada dengan menggunakan tab Tag di layar detail sumber daya. Di EFS konsol Amazon, 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

## Menandai sistem file atau sumber daya titik akses (AWS CLI)

- Jika Anda menggunakan Amazon EFSAPI, Amazon, atau an AWS CLI AWS SDK, Anda dapat menggunakan TagResource EFS API tindakan 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 EFS API tindakan Amazon yang setara, tercantum dalam tabel berikut.

CLIperintah	Deskripsi	APIOperasi yang setara
<a href="#">tag-resource</a>	Tambahkan tag baru atau perbarui tag yang ada	<a href="#">TagResource</a>
<a href="#">list-tags-for-resource</a>	Ambil tag yang ada	<a href="#">ListTagsForResource</a>
<a href="#">untag-resource</a>	Hapus tag yang ada	<a href="#">UntagResource</a>

## Tutorial: Membuat subdirektori per pengguna yang dapat ditulis

Setelah Anda membuat sistem EFS file dan memasangnya secara lokal pada EC2 instance Anda, itu mengekspos direktori kosong yang disebut *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 EC2 instance dan memasang subdirektori pada direktori home pengguna. Semua file dan subdirektori yang dibuat pengguna di direktori home mereka kemudian dibuat pada sistem EFS file.

**Note**

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

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

**1. Buat mike pengguna:**

- Masuk ke EC2 instans 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.**

Sebagai contoh:

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

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

Sebagai contoh:

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

Bagian *mount-target-DNS* alamat mengidentifikasi root sistem EFS file jarak jauh.

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

# Menginstal EFS klien Amazon

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

Anda dapat menginstal EFS klien Amazon secara manual di EC2 instans Amazon 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 [Systems Manager Distributor didukung sistem operasi](#).

## Topik

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

## Dependensi untuk alat EFS

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

- NFSklien
  - `nfs-utils` untuk RHEL, CentOS, Amazon Linux, dan distribusi 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)

- Buka SSL 1.0.2 atau yang lebih baru

### Note

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

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


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

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

## Distribusi yang didukung

EFSKlien Amazon telah diverifikasi terhadap distribusi Linux dan Mac berikut:

Distribusi	Jenis Package	<b>initsistem</b>
Amazon Linux 2023 (AL2023)	rpm	systemd
Amazon Linux 2 (AL2)	rpm	systemd
CentOS 8	rpm	systemd
Amazon Linux (AL1) 2017.09	rpm	upstart

 Note

Amazon Linux (AL1) AMI mencapai puncaknya end-of-li



Distribusi	Jenis Package	<b>initsistem</b>
<div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p>fe pada 31 Desember 2023 dan tidak didukung untuk amazon-efs-utils paket yang dirilis pada April 2024 atau yang lebih baru (versi 2.0 dan yang lebih baru).</p> </div>		
Debian 11	deb	systemd
Fedora 29 - 32	rpm	systemd
macOS Big Sur		launchd
macOS Monterey		launchd
macOS Ventura		launchd
macOS Sonoma		launchd
Buka SUSE Lompatan, Tumbleweed	rpm	systemd
Oracle8	rpm	systemd
Perusahaan Topi Merah Linux (RHEL) 7, 8, 9	rpm	systemd
SUSELinux Server Perusahaan (SLES) 12, 15	rpm	systemd
Ubuntu 16.04LTS, 18.04, 20.04LTS, LTS 22.04 LTS	deb	systemd

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

## Menginstal EFS klien Amazon secara manual

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

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

### Topik

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

## Menginstal EFS klien Amazon di instans Amazon EC2 Linux

amazon-efs-utilsPaket untuk menginstal di Amazon EC2 Linux instance dari lokasi berikut:

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

### Note

- Jika Anda menggunakan AWS Direct Connect, Anda dapat menemukan petunjuk instalasi di [Prasyarat](#).
- Amazon Linux (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 to Amazon Linux 2023 (AL2023), yang mencakup dukungan jangka panjang hingga 2028.

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

1. Pastikan Anda telah membuat EC2 instance AL2 023, Amazon Linux 2 (AL2), atau Amazon Linux (AL1). 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 EC2 instans Anda](#) di Panduan EC2 Pengguna Amazon.
3. Jalankan perintah berikut untuk menginstal paket `amazon-efs-utils`.

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

## Menginstal EFS klien Amazon di distribusi Linux lainnya

Jika Anda tidak ingin mendapatkan `amazon-efs-utils` paket dari repositori AMI paket 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 Amazon Linux 2023 (AL2023), Amazon Linux 2 (), Amazon Linux (AL2AL1), 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 Github](#) `amazon-efs-utils` README.

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

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

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

## Langkah selanjutnya

Setelah menginstal `amazon-efs-utils` EC2 instans 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 EFS klien Amazon menggunakan AWS Systems Manager

Anda dapat menggunakan AWS Systems Manager untuk menyederhanakan pengelolaan EFS klien Amazon (`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 EC2 instance Anda. Kemampuan Systems Manager seperti Distributor dan State Manager memungkinkan Anda mengotomatiskan proses berikut:

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

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

## Apa yang dilakukan EFS klien Amazon selama instalasi

Anda menggunakan EFS klien Amazon 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 EFS klien Amazon di EC2 instans Amazon menggunakan Systems Manager, diperlukan tindakan berikut:

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

## Systems Manager Distributor didukung sistem operasi

EC2 Instans Anda harus menjalankan salah satu sistem operasi berikut agar dapat digunakan untuk memperbarui atau menginstal EFS klien Amazon 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 (AL1)	2017.09, 2018.03	x86_64

 **Note**

Amazon Linux (AL1) AMI mencapai puncaknya end-of-life pada 31 Desember

Platform	Versi platform	Arsitektur
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
Perusahaan Topi Merah Linux (RHEL)	7, 8	x86_64, arm64 (RHEL7.6 dan yang lebih baru, jenis instans A1)
SUSE Server Perusahaan 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 instance 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 IAM instance dengan izin yang diperlukan

Secara default, AWS Systems Manager tidak memiliki izin untuk mengelola EFS klien Amazon 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 instance adalah wadah yang meneruskan informasi IAM peran ke EC2 instans Amazon 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 instance ini untuk meluncurkan EC2 instans Amazon 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 EC2 instans terkelola. Untuk melihat versi terbaru `amazon-efs-utils` yang tersedia untuk instalasi, Anda dapat menggunakan AWS Systems Manager konsol atau alat baris AWS perintah pilihan Anda. Untuk mengakses Distributor, buka <https://console.aws.amazon.com/systems-manager/> dan pilih Distributor di panel navigasi kiri. Temukan `AmazonEFSUtils` di bagian Dimiliki oleh Amazon. Pilih `AmazonEFSUtils` untuk melihat detail paket. Untuk informasi selengkapnya, lihat [Melihat paket](#).

Menggunakan State Manager, Anda dapat menginstal atau memperbarui `amazon-efs-utils` paket pada EC2 instans terkelola Anda segera atau sesuai jadwal. Selain itu, Anda dapat memastikan bahwa `amazon-efs-utils` secara otomatis diinstal pada EC2 instance 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 instance untuk mendaftarkan semua EC2 instance baru dan yang sudah ada sebagai target untuk menginstal atau memperbarui A secara otomatis. mazonEFSUtils 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 instance, Anda harus meluncurkan EC2 instance Anda dengan tag untuk memungkinkan AWS Systems Manager menginstal atau memperbarui EFS klien Amazon secara otomatis.
- Untuk Tentukan jadwal pengaturan yang disarankan untuk A mazonEFSUtils adalah setiap 30 hari. Anda dapat menggunakan kontrol untuk membuat cron atau jadwal tarif untuk asosiasi.

Untuk menggunakan AWS Systems Manager untuk me-mount sistem EFS file Amazon ke beberapa EC2 instance, lihat [EFSMemasang ke beberapa EC2 instance](#) .

## Instalasi dan upgrade **botocore**

EFSKlien Amazon menggunakan `botocore` untuk berinteraksi dengan AWS layanan lain. Ini diperlukan jika Anda ingin memantau keberhasilan atau kegagalan upaya pemasangan untuk sistem EFS file 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](#) 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 Online Certificate Status Protocol (OCSP) dan pemeriksaan nama host sertifikat. Pembantu EFS mount menggunakan `stunnel` program untuk TLS fungsinya. Perhatikan bahwa beberapa versi Linux tidak menyertakan versi `stunnel` yang mendukung TLS fitur ini secara default. Saat menggunakan salah satu distribusi Linux tersebut, pemasangan sistem EFS file 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 [SLES12](#))

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. UntukRPM:

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

UntukDEB:

```
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 instance EC2 Mac Anda, dan jalankan perintah berikut untuk meng-upgrade ke versi terbaru stunnel.

```
brew upgrade stunnel
```

Upgrade stunnel untuk 12 SLES

- 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 Anda menggunakan pengaturan TLS yang EFS disarankan Amazon.

## 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 helper mount AmazonEFS. Kami tidak menyarankan Anda menonaktifkan fitur ini di lingkungan produksi. Untuk menonaktifkan pemeriksaan nama host sertifikat, lakukan hal berikut:

1. Menggunakan editor teks pilihan Anda, buka `/etc/amazon/efs/efs-utils.conf` file.
2. Tetapkan `stunnel_check_cert_hostname` nilainya ke `false`.
3. Simpan perubahan ke file dan tutup.

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

## Mengaktifkan Protokol Status Sertifikat Online

Untuk memaksimalkan ketersediaan sistem file jika CA tidak dapat dijangkau dari AndaVPC, 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 TLS sertifikatnya, dan CA menginstruksikan klien untuk menggunakannya OCSP untuk memeriksa sertifikat yang dicabut. OCSP titik akhir harus dapat diakses melalui Internet dari Virtual Private Cloud Anda untuk memeriksa status sertifikat. Di dalam layanan, 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 terhadap penggunaan berbahaya sertifikat yang dicabut, yang tidak mungkin terjadi di dalam AndaVPC. Jika EFS TLS sertifikat dicabut, Amazon akan menerbitkan buletin keamanan dan merilis versi baru EFS mount helper yang menolak sertifikat yang dicabut.

Untuk OCSP mengaktifkan klien Linux Anda untuk semua TLS koneksi 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 **mount** perintah

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

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

# Memasang sistem EFS file

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

Atau, Anda dapat memasang sistem EFS file secara manual menggunakan NFS klien Linux standar. Amazon EFS mendukung protokol Sistem File Jaringan 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 mengonfigurasi EC2 instance untuk memasang sistem EFS file secara otomatis saat instance dimulai.

## Topik

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

## Pertimbangan pemasangan untuk Linux

Kami merekomendasikan nilai berikut untuk opsi pemasangan di Linux:

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

- `wsize=1048576`— Menetapkan jumlah maksimum byte data yang NFS klien dapat mengirim untuk setiap WRITE permintaan jaringan. Nilai ini berlaku saat menulis data ke file pada sistem EFS file. Kami menyarankan Anda menggunakan ukuran sebesar mungkin (hingga `1048576`) untuk menghindari penurunan kinerja.
- `hard`— Mengatur perilaku pemulihan NFS klien setelah waktu NFS permintaan habis, sehingga NFS permintaan 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 NFS klien untuk menunggu respons sebelum mencoba ulang NFS permintaan hingga `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 NFS klien mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.
- `noresvport`— Memberitahu NFS klien untuk menggunakan port sumber Transmission Control Protocol (TCP) non-privileged baru ketika koneksi jaringan dibangun kembali. Melakukan hal ini membantu memastikan bahwa sistem EFS file memiliki ketersediaan tanpa gangguan setelah peristiwa pemulihan jaringan.
- `_netdev`— Ketika hadir/`etc/fstab`, mencegah klien dari mencoba untuk me-mount sistem EFS file sampai jaringan telah 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 EFS sumber Amazon, 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 DNS namanya. Amazon EFS menyelesaikan nama ini ke alamat IP target EFS pemasangan Amazon di Availability Zone yang sama dengan EC2 instans Amazon Anda tanpa memanggil sumber daya eksternal. Jika Anda menggunakan target pemasangan di Availability Zone yang berbeda dengan EC2 instans Amazon, Anda akan dikenakan EC2 biaya standar untuk data yang dikirim melalui 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 EC2 instans Anda perlu dimulai terlepas dari status sistem EFS file yang Anda pasang, tambahkan `nofail` opsi ke entri sistem file Anda di `/etc/fstab` file Anda.

## Memasang sistem EFS file menggunakan EFS mount helper

Setelah menginstal Amazon EFS client (`amazon-efs-utils`), Anda dapat menggunakan EFS mount helper untuk memasang sistem EFS file di instance EC2 Linux dan Mac yang menjalankan distribusi yang [didukung](#).

#### Note

Amazon EFS tidak mendukung pemasangan dari instans Amazon EC2 Windows.

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

#### Warning

Gunakan opsi `_netdev`, yang digunakan untuk mengidentifikasi sistem file jaringan, ketika memasang sistem file Anda secara otomatis. Jika `_netdev` tidak ada, EC2 instance 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:

- DNSNama sistem file — Jika Anda menggunakan DNS nama sistem file, dan mount helper tidak dapat menyelesaikannya, misalnya ketika Anda memasang sistem file di sistem yang berbedaVPC, itu akan kembali menggunakan alamat IP target mount. Untuk informasi selengkapnya, lihat [Memasang sistem EFS file 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 EFS konsol Amazon. DNSNama sistem file ditemukan di layar Lampirkan.

Saat enkripsi data dalam perjalanan dideklarasikan sebagai opsi pemasangan untuk sistem EFS file Amazon Anda, mount helper menginisialisasi `stunnel` proses klien, dan proses pengawas dipanggil. `amazon-efs-mount-watchdog` `amazon-efs-mount-watchdog` Proses ini memantau kesehatan TLS mount, dan dimulai secara otomatis saat pertama kali sistem EFS file dipasang. 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 NFS klien ke port lokal ini.

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

Topik

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



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

## Pengaturan pemasangan yang digunakan oleh EFS mount helper

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

- `nfsvers=4.1`— digunakan saat memasang pada instance EC2 Linux
  - `nfsvers=4.0`— digunakan saat memasang pada instance EC2 Mac yang didukung yang menjalankan macOS Big Sur, Monterey, dan Ventura
- `rsize=1048576`— Menetapkan jumlah maksimum byte data yang dapat diterima NFS klien untuk setiap READ permintaan jaringan ke 1048576, terbesar yang tersedia, untuk menghindari penurunan kinerja.
- `wsize=1048576`— Menetapkan jumlah maksimum byte data yang dapat dikirim NFS klien untuk setiap WRITE permintaan jaringan, yang terbesar yang tersedia1048576, untuk menghindari kinerja yang berkurang.
- `hard`— Mengatur perilaku pemulihan NFS klien setelah waktu NFS permintaan habis, sehingga NFS permintaan dicoba ulang tanpa batas hingga server membalas, untuk memastikan integritas data.
- `timeo=600`— Menetapkan nilai batas waktu yang digunakan NFS klien untuk menunggu respons sebelum mencoba ulang NFS permintaan ke 600 desidetik (60 detik) untuk menghindari penurunan kinerja.
- `retrans=2`— Set ke 2 berapa kali NFS klien mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.

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

## Mendapatkan log dukungan

Pembantu EFS pemasangan memiliki logging bawaan untuk sistem EFS file Amazon 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 EFS file Amazon 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 apa pun berlaku, Anda perlu melepas dan memasang kembali sistem file menggunakan mount helper. EFS 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 mount helper EFS

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

- ID sistem file dari sistem file yang akan dipasang - Helper EFS mount menyelesaikan ID sistem file ke alamat IP lokal dari mount target elastic network interface (ENI) tanpa memanggil sumber daya eksternal.
- Target EFS pemasangan Amazon — Anda membuat target mount di cloud pribadi virtual (VPC). Jika Anda membuat sistem file di konsol menggunakan pengaturan yang direkomendasikan layanan, target pemasangan dibuat di setiap Availability Zone di Wilayah AWS 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 DNS catatan menyebar sepenuhnya di Wilayah AWS tempat sistem file berada.

Jika Anda menggunakan target pemasangan di Availability Zone yang berbeda dengan EC2 instans, Anda akan dikenakan EC2 biaya standar untuk data yang dikirim melalui 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 EC2 instance.
  - Pasang DNS nama target - Atau, Anda dapat menentukan DNS nama target mount alih-alih Availability Zone.
- EC2Instans Amazon 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

EC2Instans Mac yang menjalankan macOS Big Sur NFS mendukung 4.0 saja.

- Helper EFS mount Amazon diinstal pada EC2 instance — Mount helper adalah alat dalam `amazon-efs-utils` paket utilitas. Untuk informasi tentang menginstal `amazon-efs-utils`, lihat [Menginstal EFS klien secara otomatis](#) dan [Instalasi secara manual amazon-efs-utils](#).
- EC2Instans ada di VPC - EC2 Instans penghubung harus berada di cloud pribadi virtual (VPC) berdasarkan VPC layanan Amazon. Itu juga harus dikonfigurasi untuk menggunakan DNS server yang disediakan oleh AWS. Untuk informasi tentang DNS server Amazon, lihat [Set DHCP Opsi](#) di Panduan VPC Pengguna Amazon.
- VPCtelah mengaktifkan DNS nama host - EC2 Instans VPC penghubung harus mengaktifkan DNS nama host. Untuk informasi selengkapnya, lihat [Melihat DNS Nama Host untuk EC2 Instans Anda](#) di Panduan VPC Pengguna Amazon.
- Untuk EC2 contoh dan sistem file yang berbeda Wilayah AWS — Jika EC2 instance dan sistem file yang Anda pasang terletak berbeda Wilayah AWS, Anda perlu mengedit `region` properti dalam `efs-utils.conf` file. Untuk informasi selengkapnya, lihat [Memasang sistem EFS file dari yang berbeda Wilayah AWS](#).

## Pemasangan di instans Amazon EC2 Linux menggunakan EFS mount helper

Prosedur ini membutuhkan yang berikut:

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

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

1. Buka jendela terminal pada EC2 instance Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke EC2 instans Anda](#) di Panduan EC2 Pengguna Amazon.
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 EC2 instance dan sistem file yang Anda pasang terletak di Wilayah AWS s yang berbeda, lihat [Memasang sistem EFS file dari yang berbeda Wilayah AWS](#) 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 DNS nama 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 EFS konsol Amazon, 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 via DNS) Menggunakan DNS nama sistem file dengan EFS mount helper atau NFS klien.
- (Pasang melalui IP) Menggunakan alamat IP target mount di Availability Zone yang dipilih dengan NFS klien.

## Pemasangan di instans Amazon EC2 Mac menggunakan EFS mount helper

Prosedur ini membutuhkan yang berikut:

- Anda telah menginstal `amazon-efs-utils` paket pada instance EC2 Mac. Untuk informasi selengkapnya, lihat [Menginstal EFS klien Amazon di instance 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 instance EC2 Mac yang menjalankan macOS Big Sur, Monterey, atau Ventura. Versi macOS lainnya tidak didukung.

**Note**

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

Untuk me-mount sistem EFS file Amazon Anda menggunakan EFS mount helper pada instance EC2 Mac yang menjalankan macOS Big Sur, Monterey, atau Ventura

1. Buka jendela terminal pada instance EC2 Mac Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke EC2 instans Anda](#) di Panduan EC2 Pengguna Amazon.
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 instance 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 notls 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 EFS konsol Amazon, 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 via DNS) Menggunakan DNS nama sistem file dengan EFS mount helper atau NFS klien.
- (Pasang melalui IP) Menggunakan alamat IP target mount di Availability Zone yang dipilih dengan NFS klien.

## Memasang sistem EFS file dari yang berbeda Wilayah AWS

Untuk me-mount sistem EFS file Anda dari EC2 instance yang berbeda Wilayah AWS 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 EC2 instans Anda melalui Secure Shell (SSH), dan masuk dengan nama pengguna yang sesuai. Untuk informasi selengkapnya, lihat [Connect ke EC2 instans Anda](#) di Panduan EC2 Pengguna Amazon.
2. Temukan `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 tersebut, ganti us-east-1 dengan ID wilayah tempat 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 Amazon 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 Zones dan mencapai kinerja yang lebih baik dengan mengakses sistem EFS file menggunakan instance EC2 komputasi Amazon 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 EC2 instance. Untuk informasi selengkapnya, lihat [Menginstal EFS klien Amazon secara manual](#).
- Anda telah membuat target mount untuk sistem file. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).

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

Jika Anda memasang sistem file One Zone pada EC2 instance yang terletak di Availability Zone yang berbeda, Anda harus menentukan nama Availability Zone sistem file atau DNS nama 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 DNS nama target mount sistem file.

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

Ini adalah perintah dengan contoh DNS nama target mount.

```
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 EC2 instance yang terletak di Availability Zone yang berbeda, Anda harus menentukan nama Availability Zone sistem file atau DNS nama target pemasangan sistem file di `/etc/fstab` entri.

```
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 me-mount sistem EFS file menggunakan penyimpanan One Zone pada EC2 instance yang terletak di Availability Zone yang berbeda, Anda harus menentukan nama Availability Zone sistem file dengan DNS nama sistem file di `/etc/fstab` entri.

```
availability-zone-name.file-system-id.efs.aws-region.amazonaws.com:/ efs-mount-point  
nfs4
```

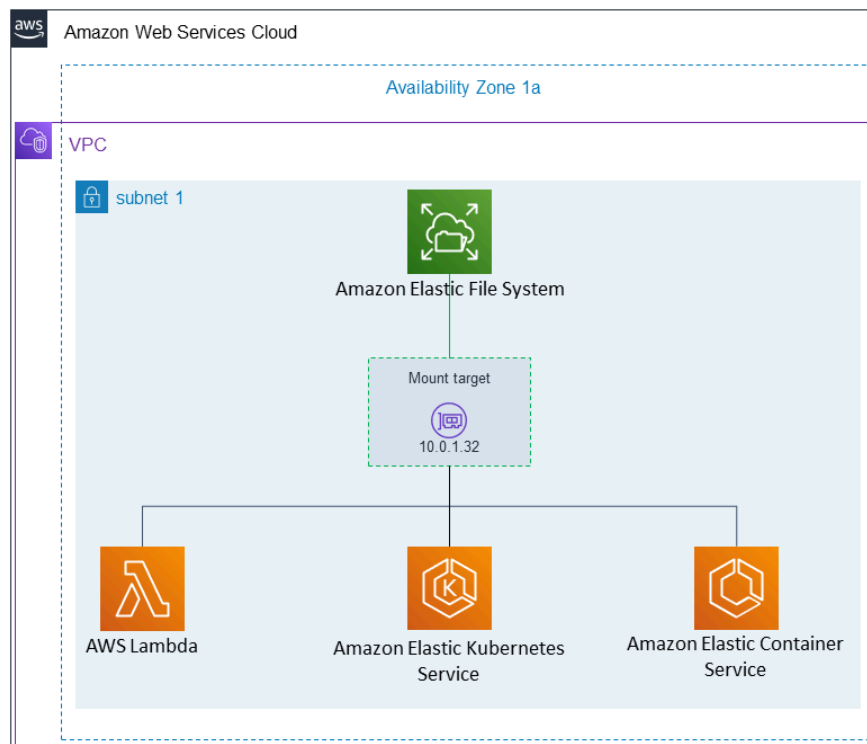
```
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
```

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

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

## 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 lokasi sistem file, diilustrasikan sebagai berikut, dan dijelaskan di EFS bagian berikut.



## Menghubungkan dari Amazon Elastic Container Service

Anda dapat menggunakan sistem EFS file Amazon 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 Amazon 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 [EFSvolume Amazon](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

## Menghubungkan dari Amazon Elastic Kubernetes Service

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

## Menghubungkan dari AWS Lambda

Anda dapat menggunakan Amazon AWS Lambda untuk EFS 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 EFS mount Amazon 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 IAM otorisasi

Untuk me-mount sistem EFS file Anda di instance Linux menggunakan otorisasi AWS Identity and Access Management (IAM), gunakan EFS mount helper. Untuk informasi selengkapnya tentang IAM otorisasi untuk NFS klien, lihat [Menggunakan IAM untuk mengontrol akses data 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 EC2 instance

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

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

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

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

## Pemasangan dengan IAM menggunakan profil bernama

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

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

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

Untuk secara otomatis me-mount dengan IAM otorisasi 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 EFS akses

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

### Note

Anda harus mengkonfigurasi satu atau lebih target mount untuk sistem file Anda saat memasang sistem file menggunakan titik EFS akses.

Saat Anda memasang sistem file menggunakan titik akses, perintah mount menyertakan opsi `access-point-id` dan `tls` mount selain opsi pemasangan biasa. Contoh ditunjukkan sebagai 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 EC2 instance.

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

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

## EFS Memasang ke beberapa EC2 instance

Anda dapat memasang sistem EFS file ke beberapa EC2 instans Amazon dari jarak jauh dan aman tanpa harus masuk ke instance dengan 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 EFS file menggunakan metode ini:

1. EC2Instans diluncurkan dengan profil instance yang menyertakan kebijakan `AmazonElasticFileSystemsUtils` izin. Untuk informasi selengkapnya, lihat [Langkah 1: Konfigurasi profil IAM instance dengan izin yang diperlukan](#).
2. Versi 1.28.1 atau yang lebih baru dari EFS klien Amazon (`amazon-efs-utils` paket) diinstal pada instance. 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 me-mount beberapa sistem EFS file ke beberapa EC2 instance 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 EFS file yang ingin Anda pasang. Sebagai 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 EFS mount menggunakan EFS klien Amazon, lihat [Pemasangan di instans Amazon EC2 Linux menggunakan EFS mount helper](#) atau [Pemasangan di instans Amazon EC2 Mac menggunakan EFS mount helper](#).

7. Pilih EC2 instance 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 EFS file yang ditentukan dalam perintah.

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

## Memasang sistem EFS file dari yang lain Akun AWS atau VPC

Anda dapat memasang sistem EFS file Amazon menggunakan IAM otorisasi untuk NFS klien dan Titik EFS Akses menggunakan EFS mount helper. Secara default, EFS mount helper menggunakan domain name service (DNS) untuk menyelesaikan alamat IP target EFS mount Anda. Jika Anda memasang sistem file dari akun lain atau virtual private cloud (VPC), Anda perlu menyelesaikan target EFS pemasangan secara manual.

Berikut ini, Anda dapat menemukan petunjuk untuk menentukan alamat IP target EFS mount yang benar untuk digunakan untuk NFS klien Anda. Anda juga dapat menemukan instruksi untuk mengkonfigurasi klien untuk me-mount sistem EFS file menggunakan alamat IP tersebut.

Topik

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

## Memasang sistem EFS file dari yang lain Akun AWS

Menggunakan sharedVPCs, Anda dapat me-mount sistem EFS file Amazon yang dimiliki oleh salah satu Akun AWS dari EC2 instans Amazon yang dimiliki oleh yang berbeda Akun AWS. Untuk

informasi selengkapnya tentang menyiapkan berbagiVPC, lihat [Berbagi VPC dengan akun lain](#) di Panduan VPC Peering Amazon.

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

## Memasang sistem EFS file dari yang lain VPC

Saat Anda menggunakan koneksi VPC peering atau gateway transit untuk terhubungVPCs, EC2 instans Amazon yang ada di satu VPC dapat mengakses sistem EFS file di yang lainVPC, bahkan jika itu VPCs milik akun yang berbeda.

### Prasyarat

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

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

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

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

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

Koneksi VPC peering 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 yang sama Wilayah AWS atau di antara Wilayah AWS. Untuk informasi lebih lanjut tentang VPC mengintip, lihat [Apa itu VPC Peering?](#) di Panduan VPC Peering Amazon.

Untuk memastikan ketersediaan sistem file Anda yang tinggi, kami sarankan Anda selalu menggunakan alamat IP target EFS mount yang berada di Availability Zone yang sama dengan NFS klien Anda. Jika Anda memasang sistem EFS file yang ada di akun lain, pastikan NFS klien dan target EFS mount 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 EFS file di tempat lain VPC menggunakan IAM atau titik akses

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

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

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

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

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

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

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

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

## Memasang sistem EFS file dari yang berbeda Wilayah AWS

Jika Anda memasang sistem EFS file Anda dari VPC yang lain yang berbeda Wilayah AWS dari sistem file, Anda perlu mengedit `efs-utils.conf` file. 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 Sistem File Jaringan untuk me-mount sistem EFS file

Berikut ini, pelajari cara menginstal klien Network File System (NFS) dan cara memasang sistem EFS file Amazon Anda pada EC2 instans Amazon. Anda juga dapat menemukan penjelasan tentang mount perintah dan opsi yang tersedia untuk menentukan nama Sistem Nama Domain (DNS) sistem file Anda dalam mount perintah. 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 EFS file Amazon 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 EFS klien Amazon](#).

## Topik

- [Prasyarat](#)
- [Dukungan NFS](#)
- [Menginstal NFS klien](#)
- [Pengaturan NFS pemasangan yang disarankan](#)
- [Memasang di Amazon EC2 dengan DNS nama](#)
- [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 VPC keamanan untuk EC2 instans Amazon Anda dan pasang target dengan akses masuk dan keluar yang diperlukan. Untuk informasi selengkapnya, lihat [Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target](#).

## Dukungan NFS

Amazon EFS mendukung protokol Sistem File Jaringan versi 4.0 dan 4.1 (NFSv4) saat memasang sistem file Anda di instans AmazonEC2. Meskipun NFSv4 .0 didukung, kami menyarankan Anda menggunakan NFSv4 .1. Memasang sistem EFS file Amazon Anda di EC2 instans Amazon Anda juga memerlukan NFS klien yang mendukung NFSv4 protokol yang Anda pilih. Instans Amazon EC2 Mac yang menjalankan macOS Big Sur hanya NFS mendukung v4.0.

Amazon EFS tidak mendukung opsi `nconnect` pemasangan.

#### Note

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

Untuk kinerja optimal dan untuk menghindari berbagai bug NFS klien 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 7 dan yang lebih baru
- Ubuntu 16.04 LTS dan yang lebih baru
- SLES12 Sp2 atau 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 versi tertentu AMI atau kernel saat menggunakan Amazon EFS dari sebuah EC2 instance, lihat [Pemecahan masalah AMI dan masalah kernel](#)

#### Note

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

## Menginstal NFS klien

Untuk me-mount sistem EFS file Anda di EC2 instans Amazon Anda, pertama-tama Anda perlu menginstal NFS klien. Untuk terhubung ke EC2 instans Anda dan menginstal NFS klien, Anda memerlukan DNS nama publik EC2 instance dan nama pengguna untuk masuk. Nama pengguna untuk contoh Anda biasanya `ec2-user`.

Untuk menghubungkan EC2 instans Anda dan menginstal NFS klien

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

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 EC2 instans Amazon Anda](#).

2. (Opsional) Dapatkan pembaruan dan reboot.

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

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

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

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

Jika Anda menggunakan Ubuntu Amazon EC2AMI, instal NFS klien dengan perintah berikut.

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

5. Mulai NFS layanan menggunakan perintah berikut. Untuk RHEL 7:

```
$ sudo service nfs start
```

Untuk RHEL 8:

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

6. Verifikasi bahwa NFS layanan dimulai, sebagai berikut.

```
$ sudo service nfs 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 kustomAMI), 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 EC2 Amazon Anda, Anda tidak perlu `nfs-utils` menginstal karena sudah disertakan secara default. AMI

Berikutnya: Pasang sistem file Anda

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

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

## Pengaturan NFS pemasangan yang disarankan

Kami merekomendasikan nilai berikut untuk opsi pemasangan di Linux:

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

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

**⚠ Important**

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

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

- `rsize=1048576`— Menetapkan jumlah maksimum byte data yang dapat diterima NFS klien untuk setiap READ permintaan jaringan. Nilai ini berlaku saat membaca data dari file pada sistem EFS file. Kami menyarankan Anda menggunakan ukuran sebesar mungkin (hingga `1048576`) untuk menghindari penurunan kinerja.
- `wsiz=1048576`— Menetapkan jumlah maksimum byte data yang NFS klien dapat mengirim untuk setiap WRITE permintaan jaringan. Nilai ini berlaku saat menulis data ke file pada sistem EFS file. Kami menyarankan Anda menggunakan ukuran sebesar mungkin (hingga `1048576`) untuk menghindari penurunan kinerja.
- `hard`— Mengatur perilaku pemulihan NFS klien setelah waktu NFS permintaan habis, sehingga NFS permintaan 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 NFS klien untuk menunggu respons sebelum mencoba ulang NFS permintaan hingga `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 NFS klien mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.
- `_netdev`— Ketika hadir/`etc/fstab`, mencegah klien dari mencoba untuk me-mount sistem EFS file sampai jaringan telah diaktifkan.

- `nofail`— Jika EC2 instans Anda perlu memulai terlepas dari status sistem EFS file 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 EFS sumber Amazon, 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 DNS namanya. Nama ini diselesaikan ke alamat IP target EFS pemasangan Amazon di Availability Zone yang sama dengan instans Amazon EC2 Anda. Jika Anda menggunakan target pemasangan di Availability Zone yang berbeda dengan EC2 instans Amazon, Anda akan dikenakan EC2 biaya standar untuk data yang dikirim melalui Availability Zone. Anda juga mungkin melihat peningkatan latensi untuk operasi sistem file.
- Untuk opsi pemasangan lainnya, dan penjelasan rinci tentang default, lihat dokumentasi Linux.

## Memasang di Amazon EC2 dengan DNS nama

### Note

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

- DNSNama sistem file — Menggunakan nama sistem file adalah opsi pemasangan paling sederhana Anda. DNS DNSNama sistem file secara otomatis menyelesaikan alamat IP



target mount di Availability Zone dari instans Amazon EC2 yang menghubungkan. Anda bisa mendapatkan DNS nama dari konsol, atau jika Anda memiliki ID sistem file, Anda dapat membuatnya menggunakan konvensi berikut.

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

#### Note

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

- Menggunakan DNS nama sistem file, Anda dapat me-mount sistem file pada instance Amazon EC2 Linux Anda dengan perintah berikut.

```
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
```

- Dengan menggunakan DNS nama 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,rsize=65536,wsiz=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 EFS file saat memasang pada instance EC2 Mac yang menjalankan mendukung versi macOS.

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

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

**Note**

Resolusi DNS nama target pemasangan di seluruh Availability Zone didukung.

Dalam beberapa kasus, Anda mungkin menghapus target mount dan kemudian membuat yang baru di Availability Zone yang sama. Dalam kasus seperti itu, DNS nama untuk target mount baru di Availability Zone itu sama dengan DNS nama 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 EFS konsol Amazon, 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. DNSTampilan Mount via default menampilkan perintah untuk me-mount sistem file menggunakan DNS nama sistem file saat memasang dengan EFS mount helper atau NFS klien.

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

Untuk dapat menggunakan DNS nama dalam mount perintah, berikut ini harus benar:

- EC2Instans penghubung harus berada di dalam VPC dan harus dikonfigurasi untuk menggunakan DNS server yang disediakan oleh Amazon. Untuk informasi tentang DNS server Amazon, lihat [set DHCP opsi di Amazon VPC](#) di Panduan VPC Pengguna Amazon.
- EC2Instans VPC penghubung harus mengaktifkan DNSResolusi dan DNSNama Host. Untuk informasi selengkapnya, lihat [Melihat DNS nama host untuk EC2 instans Anda](#) di Panduan VPC Pengguna Amazon.

- EC2Instance penghubung harus berada di dalam VPC sama dengan sistem EFS file. Untuk informasi lebih lanjut tentang mengakses dan memasang sistem file dari lokasi lain atau dari lokasi lainVPC, lihat [Prasyarat](#) dan [Tutorial: Pasang sistem file dari yang berbeda VPC](#) .

#### Note

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

## Pemasangan dengan alamat IP

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

Anda juga dapat mengonfigurasi pemasangan sistem file menggunakan alamat IP target mount sebagai opsi fallback untuk aplikasi yang dikonfigurasi untuk memasang sistem file menggunakan DNS namanya secara default. Saat menghubungkan ke alamat IP target mount, EC2 instance 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 mengizinkan NFS akses masuk dari grup EC2 keamanan. Untuk informasi selengkapnya, lihat [Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target](#).

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

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di EFS konsol Amazon, 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 NFS klien.

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

```
sudo mount -t nfs -o
nfsvers=4.1,rsiz=1048576,wsiz=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,rsiz=65536,wsiz=65536,hard,timeo=600,retrans=2,noresvport,mountport=2049 mount-
target-IP:/ /efs
```

#### Important

Anda harus menggunakan `mountport=2049` agar berhasil terhubung ke sistem EFS file saat memasang pada instance 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 AWS 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 EFS file

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

- Menggunakan EFS mount helper:
  - Lampirkan sistem EFS file saat Anda membuat instance EC2 Linux baru menggunakan EC2 Launch Instance Wizard.
  - Perbarui EC2 `/etc/fstab` file dengan entri untuk sistem EFS file.
- Menggunakan [NFStanpa EFS mount helper](#) untuk memperbarui EC2 `/etc/fstab` file, untuk instance EC2 Linux dan Mac.

### Note

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


### Topik

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


## Mengaktifkan pemasangan otomatis pada instance EC2 Linux baru

Saat membuat instans Amazon EC2 Linux baru menggunakan EC2 Launch Instance Wizard, Anda dapat mengonfigurasinya untuk memasang sistem EFS file Amazon secara otomatis. EC2Instance memasang sistem file secara otomatis instance 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 EFS file Amazon tidak mendukung pemasangan di instans Amazon EC2 Mac yang menjalankan macOS Big Sur atau Monterey saat peluncuran instans.

 Note

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

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

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

1. Buka EC2 konsol Amazon di <https://console.aws.amazon.com/ec2/>.
2. Pilih Luncurkan Instans.
3. Pada Langkah 1: Pilih Gambar Mesin Amazon (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 Jaringan, pilih entri VPC yang sama dengan sistem EFS file yang Anda pasang.
  - Untuk Subnet, pilih subnet default di Availability Zone apa pun.
  - Untuk sistem File, pilih sistem EFS file yang ingin Anda pasang. Jalur yang ditampilkan di sebelah ID sistem file adalah titik pemasangan yang akan digunakan EC2 instance, yang dapat Anda ubah.
  - Di bawah Detail Lanjutan, data Pengguna dibuat secara otomatis, dan menyertakan perintah yang diperlukan untuk memasang sistem EFS file 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: Konfigurasikan Grup Keamanan, atur Tetapkan grup keamanan ke Pilih grup keamanan yang ada. Pilih grup keamanan default untuk memastikannya dapat mengakses sistem EFS file Anda.

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

EC2Instance Anda sekarang dikonfigurasi untuk me-mount sistem EFS file saat peluncuran dan kapan pun itu di-boot ulang.

## Mengaktifkan pemasangan otomatis pada instance 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 instance EC2 Linux secara manual sehingga instance menggunakan EFS mount helper untuk secara otomatis memasang kembali sistem EFS file saat instance restart. `/etc/fstab`

### Note

Sistem EFS file Amazon tidak mendukung pemasangan otomatis menggunakan `/etc/fstab` dengan EFS mount helper di instans Amazon EC2 Mac yang menjalankan macOS Big Sur atau Monterey. Sebagai gantinya, Anda dapat menggunakan [NFSwith /etc/](#)

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

## Prasyarat

Persyaratan berikut harus ada sebelum Anda berhasil menerapkan prosedur ini:

- Anda telah membuat sistem EFS file Amazon yang ingin Anda instal ulang secara otomatis. Untuk informasi selengkapnya, lihat [Cepat membuat sistem file yang memiliki pengaturan yang direkomendasikan \(konsol\)](#).
- Anda telah membuat instance EC2 Linux yang ingin Anda konfigurasi untuk secara otomatis memasang kembali sistem EFS file.
- EFSMount helper diinstal pada instance EC2 Linux. Untuk informasi selengkapnya, lihat [Menginstal EFS klien Amazon](#).

## 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 melakukan remount sistem file saat instance restart. EFS

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

1. Terhubung ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke EC2 instans Anda](#) di Panduan EC2 Pengguna Amazon.
2. Buka `/etc/fstab` file di editor.
3. Untuk pemasangan otomatis menggunakan IAM otorisasi atau titik EFS akses:



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

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

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

```
file-system-id:/ efs-mount-point efs  
_netdev,noresvport,tls,iam,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` tidak ada, EC2 instance 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, silakan lihat [Pemasangan dengan IAM otorisasi](#) dan [Pemasangan dengan titik EFS akses](#).

4. Simpan perubahan pada file.
5. Uji `fstab` entri dengan menggunakan `mount` perintah dengan `'fake'` opsi bersama dengan `'verbose'` opsi `'all'` dan.

```
$ sudo mount -fav  
home/ec2-user/efs      : successfully mounted
```

EC2Instance Anda sekarang dikonfigurasi untuk me-mount sistem EFS file setiap kali restart.

**Note**

Dalam beberapa kasus, EC2 instans Amazon Anda mungkin perlu memulai terlepas dari status sistem EFS file Amazon 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
<code>file-system-id</code> :/	ID untuk sistem EFS file Amazon Anda. Anda bisa mendapatkan ID ini dari konsol atau secara terprogram dari CLI atau. AWS SDK
<code>efs-mount-point</code>	Titik pemasangan untuk sistem EFS file pada EC2 instance Anda.
<code>efs</code>	Jenis sistem file. Saat Anda menggunakan mount helper, tipe ini selaluefs.
<code>mount options</code>	<p>Opsi pemasangan untuk sistem file. Ini adalah daftar opsi berikut yang dipisahkan koma:</p> <ul style="list-style-type: none"> <li><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.</li> <li><code>noresvport</code> Memberitahu NFS klien untuk menggunakan port sumber Transmission Control Protocol (TCP) baru ketika koneksi jaringan dibangun kembali. Melakukan hal ini membantu memastikan bahwa sistem EFS file memiliki ketersediaan tanpa gangguan setelah peristiwa pemulihan jaringan.</li> <li><code>tls</code>— Memungkinkan enkripsi data dalam perjalanan.</li> <li><code>iam</code>— Gunakan opsi ini untuk memasang dengan IAM otorisasi ke Amazon EC2 yang memiliki profil instance. Menggunakan opsi <code>iam</code> mount juga membutuhkan penggunaan <code>tls</code> opsi. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan IAM untuk mengontrol akses data sistem file</a>.</li> </ul>

Bidang	Deskripsi
	<ul style="list-style-type: none"> <li>• <code>awsprofile= <i>namedprofile</i></code> — Gunakan opsi ini dengan <code>tls</code> opsi <code>iam</code> dan untuk me-mount dengan IAM otorisasi ke instance Linux menggunakan file kredensial. Untuk informasi selengkapnya tentang titik EFS akses, lihat <a href="#">Menggunakan IAM untuk mengontrol akses data sistem file</a>.</li> <li>• <code>accesspoint= <i>access-point-id</i></code> — Gunakan opsi ini dengan <code>tls</code> opsi untuk me-mount menggunakan titik EFS akses. Untuk informasi selengkapnya tentang titik EFS akses, lihat <a href="#">Bekerja dengan titik EFS akses Amazon</a>.</li> </ul>
0	Nilai bukan nol menunjukkan bahwa sistem file harus didukung oleh <code>dump</code> . Karena EFS, nilai ini seharusnya 0.
0	Urutan di mana <code>fsck</code> memeriksa sistem file saat boot. Untuk sistem EFS file, nilai ini 0 harus menunjukkan bahwa tidak <code>fsck</code> boleh berjalan saat start-up.

## Mengaktifkan pemasangan otomatis pada instans EC2 Linux atau Mac menggunakan NFS

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

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

1. Terhubung ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke EC2 instans Anda](#) di Panduan EC2 Pengguna Amazon.
2. Buka `/etc/fstab` file di editor.
3. Untuk secara otomatis me-mount sistem file menggunakan NFS bukan EFS mount helper, tambahkan baris berikut ke `/etc/fstab` file.
  - Ganti `file_system_id` dengan ID sistem file yang Anda pasang.
  - Ganti `aws-region` dengan sistem Wilayah AWS file di dalamnya, seperti `us-east-1`.
  - Ganti `mount_point` dengan titik pemasangan 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
<code>file-system-id</code> :/	ID untuk sistem EFS file Amazon Anda. Anda bisa mendapatkan ID ini dari konsol atau secara terprogram dari CLI atau AWS SDK
<code>efs-mount-point</code>	Titik pemasangan untuk sistem EFS file pada EC2 instance Anda.
<code>nfs4</code>	Menentukan jenis sistem file.
<code>mount options</code>	<p>Daftar opsi mount yang dipisahkan koma untuk sistem file:</p> <ul style="list-style-type: none"> <li><code>nfsvers=4.1</code> - menentukan menggunakan NFS v4.1.</li> <li><code>rsiz=1048576</code> Untuk meningkatkan kinerja, tetapkan jumlah maksimum byte data yang dapat diterima NFS klien untuk setiap READ permintaan jaringan saat membaca data dari file pada sistem EFS file. 1048576 adalah ukuran terbesar yang mungkin.</li> <li><code>wsiz=1048576</code> Untuk meningkatkan kinerja, tetapkan jumlah maksimum byte data yang dapat dikirim NFS klien untuk setiap WRITE permintaan jaringan saat menulis data ke file pada sistem EFS file. 1048576 adalah ukuran terbesar yang mungkin.</li> <li><code>hard</code>— Mengatur perilaku pemulihan NFS klien setelah waktu NFS permintaan habis, sehingga NFS permintaan 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.</li> <li><code>timeo=600</code> — Menetapkan nilai batas waktu yang digunakan NFS klien untuk menunggu respons sebelum mencoba ulang perminta</li> </ul>

Bidang	Deskripsi
	<p>n ke 600 desidetik (60 detik). Jika Anda harus mengubah parameter batas waktu (<code>timeo</code>), kami sarankan Anda menggunakan nilai minimal150, yang setara dengan 15 detik. Melakukannya membantu menghindari penurunan kinerja.</p> <ul style="list-style-type: none"> <li>• <code>retrans=2</code> — Set ke 2 berapa kali NFS klien mencoba ulang permintaan sebelum mencoba tindakan pemulihan lebih lanjut.</li> <li>• <code>noresvport</code> Memberitahu NFS klien untuk menggunakan port sumber Transmission Control Protocol (TCP) baru ketika koneksi jaringan dibangun kembali. Melakukan hal ini membantu memastikan bahwa sistem EFS file memiliki ketersediaan tanpa gangguan setelah peristiwa pemulihan jaringan.</li> <li>• <code>_netdev</code>— Mencegah klien dari mencoba untuk me-mount sistem EFS file sampai jaringan telah diaktifkan.</li> </ul>
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 EC2 instans Amazon yang terhubung dengannya. Anda dapat melepas sistem file pada EC2 instance Amazon Anda dengan menjalankan `umount` perintah pada instance itu sendiri. Anda tidak dapat melepas sistem EFS file Amazon melalui, file AWS CLI AWS Management Console, atau melalui salah satu file. AWS SDKs Untuk melepas sistem EFS file Amazon yang terhubung ke EC2 instance Amazon 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 EFS file Amazon 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 EFS file Amazon yang ingin Anda unmount tidak tercantum dalam output `df` perintah, ini berarti bahwa sistem file dilepas.

Example — Identifikasi status pemasangan sistem EFS file Amazon 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 EFS file dan pasang pada EC2 instance menggunakan AWS CLI

Buat sistem EFS file terenkripsi, pasang pada EC2 instance di AndaVPC, dan uji penyiapannya menggunakan file. AWS CLI

### Note

Dalam [Memulai](#) tutorial, Anda menggunakan konsol untuk membuat Amazon EC2 dan EFS sumber daya. 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:

- EC2 Sumber daya Amazon:
  - Dua grup keamanan (untuk EC2 contoh dan sistem EFS file 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 TCP port NFSv4 .1 standar.

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

Untuk me-mount sistem file Anda pada sebuah EC2 instance, Anda perlu membuat target mount di file AndaVPC. Anda dapat membuat satu target pemasangan di setiap Availability Zone diVPC. Untuk informasi selengkapnya, lihat [Bagaimana Amazon EFS bekerja](#).

Kemudian, Anda menguji sistem file pada EC2 instance 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 Wilayah AWS Anda gunakan, pastikan untuk menggunakannya secara konsisten. Semua sumber daya Anda — sumber daya, sumber EC2 dayaVPC, dan EFS sumber daya Anda — harus sama. Wilayah AWS

## Topik

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

## Prasyarat

- Anda dapat menggunakan kredensi root Anda Akun AWS untuk masuk ke konsol dan mencoba latihan memulai. Namun, AWS Identity and Access Management (IAM) merekomendasikan agar Anda tidak menggunakan kredensi root Anda. Akun AWS Sebagai gantinya, buat pengguna administrator di akun Anda dan gunakan kredensialnya untuk mengelola sumber daya di akun

Anda. Sebagai gantinya, buat pengguna administrator di akun Anda dan gunakan kredensialnya untuk mengelola sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Menetapkan Akun AWS akses untuk pengguna Pusat IAM Identitas](#) di Panduan AWS IAM Identity Center Pengguna.

- Anda dapat menggunakan default VPC atau kustom VPC yang telah Anda buat di akun Anda. Untuk panduan ini, VPC konfigurasi default berfungsi. Namun, jika Anda menggunakan kustomVPC, verifikasi hal berikut:
  - DNSnama host diaktifkan. Untuk informasi selengkapnya, lihat [DNSatribut VPC di](#) Panduan VPC Pengguna Amazon.
  - Gateway internet terpasang ke AndaVPC. Untuk informasi selengkapnya, lihat [Connect to the internet menggunakan gateway internet](#) di Panduan VPC Pengguna Amazon.
  - VPCSubnet dikonfigurasi untuk meminta alamat IP publik untuk instance yang diluncurkan di subnet. VPC Untuk informasi selengkapnya, lihat [Pengalamatan IP untuk subnet Anda VPCs dan subnet](#) di VPCPanduan Pengguna Amazon.
  - Tabel VPC rute 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 konfigurasi AWS CLI. Untuk petunjuk, lihat [Memulai dengan AWS CLI](#) di Panduan AWS Command Line Interface Pengguna.
2. Tetapkan profil.

Anda menyimpan kredensi pengguna dalam file. AWS CLI config Contoh CLI perintah 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. Wilayah AWS Jika Anda tidak menentukan Region dalam CLI perintah, wilayah us-west-2 diasumsikan.

3. Verifikasi pengaturan dengan memasukkan perintah berikut pada prompt perintah. Kedua perintah ini tidak memberikan kredensi secara eksplisit, sehingga kredensi 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 EC2 sumber daya

Dalam langkah ini, Anda melakukan hal berikut:

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

### Langkah 1.1: Buat dua grup keamanan

Di bagian ini, Anda membuat grup keamanan di EC2 instans Anda dan EFS memasang target. VPC Kemudian dalam tutorial, Anda menetapkan grup keamanan ini ke EC2 instance dan target EFS

mount. Untuk informasi tentang grup keamanan, lihat [Grup EC2 keamanan Amazon untuk instans Linux](#).

Untuk membuat grup keamanan

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

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

- b. Buat grup keamanan (efs-walkthrough1-mt-sg) untuk target EFS pemasangan Anda. Anda harus memberikan VPC ID 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 EC2 instans Anda.
- Aktifkan lalu lintas antara EC2 instance dan target EFS mount (yang dengannya Anda mengaitkan grup keamanan ini nanti dalam tutorial ini).

## Langkah 1.2: Tambahkan aturan ke grup keamanan untuk mengotorisasi akses masuk/keluar

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 EC2 instance (`efs-walkthrough1-ec2-sg`) Anda sehingga Anda dapat terhubung ke EC2 instans 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 target EFS mount (`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 sebuah EC2 instance

Pada langkah ini, Anda meluncurkan sebuah EC2 instance.

Untuk meluncurkan sebuah EC2 instance

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

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

#### Note

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

Untuk Amazon Linux HVMAMI, Anda dapat menemukan yang terbaru IDs di [Amazon Linux AMI](#). Anda memilih nilai ID dari AMI IDs tabel 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 HVM64-bit yang EBS didukung (karena dalam CLI perintah Anda menentukan jenis t2.micro instance, yang tidak mendukung penyimpanan instance).
- ID grup keamanan yang Anda buat untuk sebuah EC2 instance.
- Wilayah AWS. Panduan ini menggunakan wilayah us-west-2.
- ID VPC subnet 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 EC2 instance.
- Availability Zone dari subnet — Anda memerlukan nilai ini untuk membangun DNS nama target mount Anda, yang Anda gunakan untuk me-mount sistem file pada EC2 instance.

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

```
$ 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. EC2Instance yang Anda buat harus memiliki DNS nama publik yang Anda gunakan untuk terhubung ke EC2 instance dan memasang sistem file di atasnya. DNSNama publik adalah dari bentuk:

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

Jalankan CLI perintah berikut dan tuliskan DNS nama publik.

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

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

5. (Opsional) Tetapkan nama ke EC2 instance 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 EFS sumber daya

Dalam langkah ini, Anda melakukan hal berikut:

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

### Langkah 2.1: Buat sistem EFS file

Pada langkah ini, Anda membuat sistem EFS file. 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 AWS CLI `create-file-system` perintah 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 Akses EFS Jarang (IA). Untuk mempelajari selengkapnya, lihat [Mengelola siklus hidup penyimpanan untuk sistem file EFS](#) dan [EFS kelas penyimpanan](#).

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 mount untuk sistem file Anda di Availability Zone tempat EC2 instans Anda diluncurkan.

### 1. Pastikan Anda memiliki informasi berikut:

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

Untuk tutorial ini, Anda membuat target mount di subnet yang sama di mana Anda meluncurkan EC2 instance, 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 EC2 instance dan uji

Dalam langkah ini, Anda melakukan hal berikut:

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

Topik

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

### Langkah 3.1: Kumpulkan Informasi

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

- DNSNama publik EC2 instans Anda dalam format berikut:

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

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

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

EC2Contoh di mana Anda memasang sistem file dengan menggunakan target mount dapat menyelesaikan DNS nama sistem file ke alamat IP target mount.

**Note**

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

### Langkah 3.2: Instal NFS klien pada EC2 instans Anda

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

Untuk menginstal NFS klien

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

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

Setelah reboot, sambungkan kembali ke EC2 instans Anda.

- b. Instal NFS klien.

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

**Note**

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

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

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

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

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

2. Pasang sistem EFS file.

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

EC2Instance dapat menyelesaikan DNS nama 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,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-  
target-ip:/ ~/efs-mount-point
```

3. Sekarang setelah Anda memiliki sistem EFS file yang terpasang pada EC2 instance 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 EFS file pada EC2 instance Anda di file AndaVPC.

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 EC2 instance, 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 EC2 sumber daya (EC2 contoh dan dua kelompok keamanan). Amazon EFS menghapus antarmuka jaringan saat Anda menghapus target pemasangan.
- Hapus EFS sumber daya (sistem file, pasang target).

Untuk menghapus AWS sumber daya yang dibuat dalam panduan ini

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

```
$ aws ec2 terminate-instances \
```

```
--instance-ids instance-id \  
--profile adminuser
```

Anda juga dapat menghapus EC2 sumber daya 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 `describe-mount-targets` CLI perintah.

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

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

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

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

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

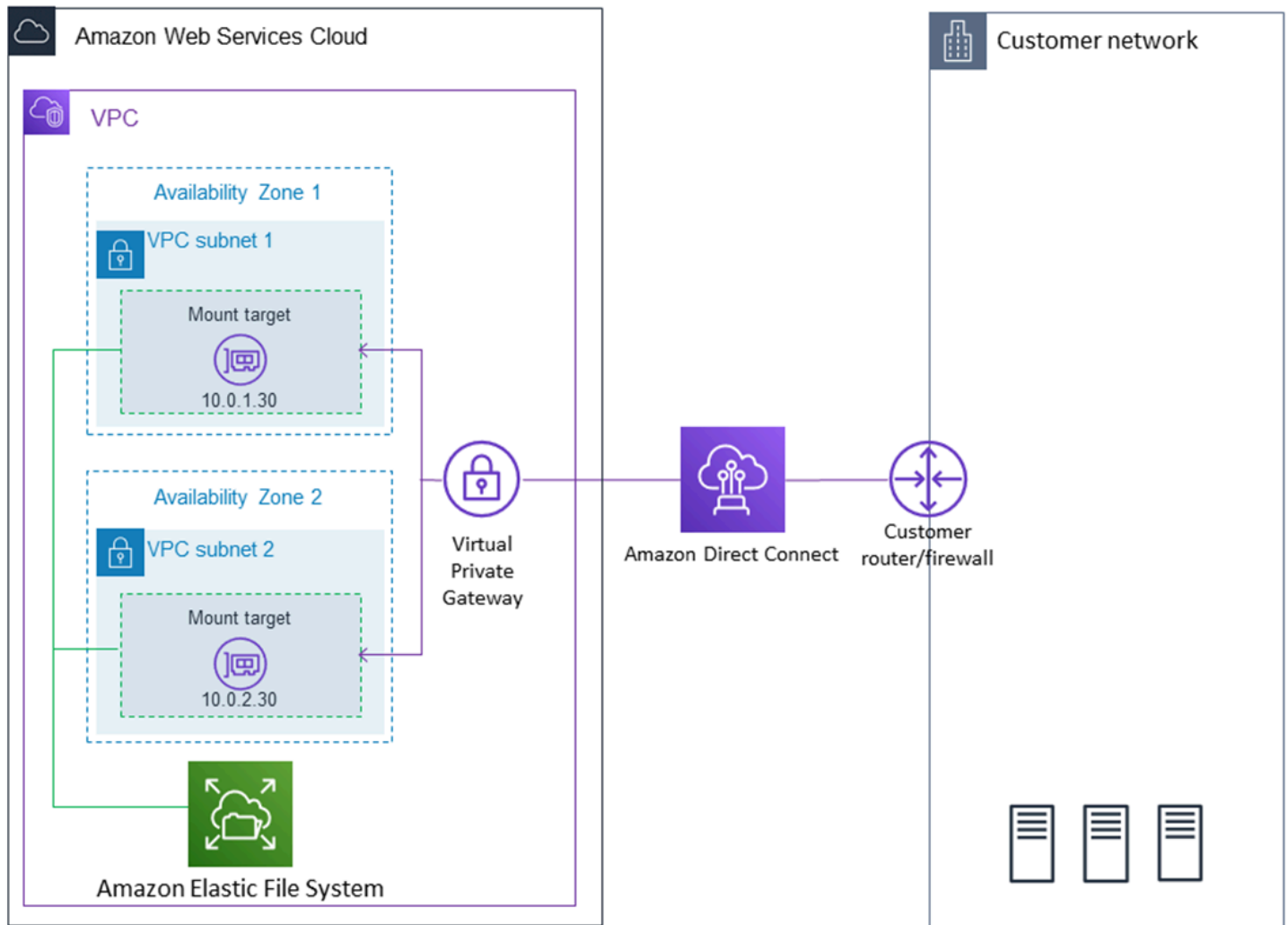
```
aws efs describe-file-systems \  
--profile adminuser \  
--region aws-region
```

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



**Note**

Menggunakan Amazon EFS dengan klien berbasis Microsoft Windows tidak didukung.

## Topik

- [Prasyarat](#)
- [Langkah 1: Buat EFS sumber daya Anda](#)
- [Langkah 2: Instal NFS klien](#)
- [Langkah 3: Pasang sistem EFS file Amazon di Klien lokal Anda](#)
- [Langkah 4: Bersihkan sumber daya dan lindungi AWS akun Anda](#)
- [Opsional: Mengenkripsi data dalam perjalanan](#)

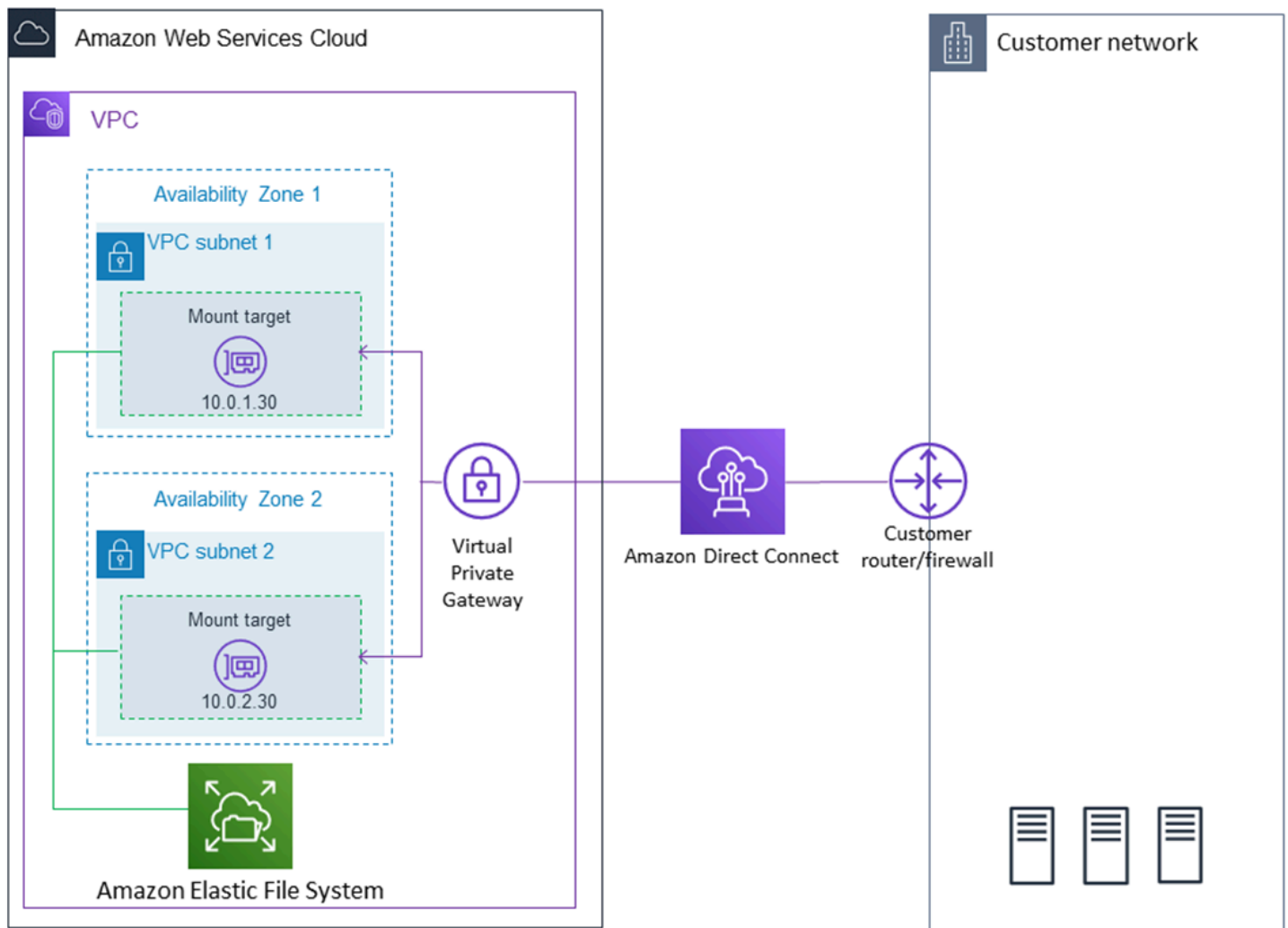
## Prasyarat

Pastikan Anda sudah memiliki koneksi AWS Direct Connect atau VPN koneksi. Untuk informasi selengkapnya tentang AWS Direct Connect, lihat [AWS Direct Connect Panduan Pengguna](#). Untuk informasi selengkapnya tentang menyiapkan VPN sambungan, lihat [VPNKoneksi](#) di Panduan VPC Pengguna Amazon.

Setelah Anda memiliki AWS Direct Connect atau VPN koneksi, buat sistem EFS file dan target mount di Amazon AndaVPC. Setelah itu, Anda mengunduh dan menginstal amazon-efs-utils alat. Kemudian, Anda menguji sistem file dari klien lokal Anda. Akhirnya, langkah pembersihan di akhir panduan 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 Wilayah AWS Anda gunakan, pastikan untuk menggunakannya secara konsisten. Semua sumber daya Anda—target pemasangan AndaVPC, dan sistem EFS file Amazon Anda—harus sama Wilayah AWS, seperti yang ditunjukkan pada diagram berikut.





### Note

Dalam beberapa kasus, aplikasi lokal Anda mungkin perlu mengetahui apakah sistem EFS file 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 kredensi root Anda Akun AWS untuk masuk ke konsol dan mencoba latihan ini. Namun, AWS Identity and Access Management (IAM) praktik terbaik menyarankan agar Anda tidak menggunakan kredensial root Anda. Akun AWS Sebagai gantinya, buat pengguna administrator di akun Anda dan gunakan kredensialnya untuk mengelola sumber daya di akun Anda.

Untuk informasi selengkapnya, lihat [Menetapkan Akun AWS akses untuk pengguna Pusat IAM Identitas](#) di Panduan AWS IAM Identity Center Pengguna.

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

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

## Langkah 1: Buat EFS sumber daya Anda

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

Untuk membuat sistem EFS file Anda

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Pilih Buat Sistem File.
3. Pilih default Anda VPC dari VPC daftar.
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 dengan itu. 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 AWS Management Console untuk menambahkan aturan ke grup keamanan target mount Anda di grup AndaVPC.

Untuk memungkinkan lalu lintas masuk ke port NFS

1. Masuk ke AWS Management Console dan buka EC2 konsol Amazon 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 EFS sumber daya 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 TCP koneksi pada NFS port, mengidentifikasi grup keamanan target mount sebagai tujuan.

## Langkah 2: Instal NFS klien

Pada langkah ini, Anda menginstal NFS klien.

## Untuk menginstal NFS klien di server lokal

### Note

Jika Anda memerlukan data untuk dienkripsi saat transit, gunakan Amazon EFS mount helper `amazon-efs-utils`, bukan klien NFS. Untuk informasi tentang menginstal `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 EFS file Amazon di Klien lokal Anda

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 EC2 instance 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 EFS file Amazon Anda, Anda dapat mengujinya dengan prosedur berikut.

Untuk menguji koneksi sistem EFS file Amazon

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

```
$ cd ~/efs
```

2. Buat subdirektori dan ubah kepemilikan subdirektori tersebut ke pengguna EC2 instans 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` tidak ada, EC2 instance 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 EFS file Amazon dengan perintah berikut.

```
$ sudo umount ~/efs
```

2. Buka EFS konsol Amazon di <https://console.aws.amazon.com/efs/>.
3. Pilih sistem EFS file Amazon 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 EFS file Amazon yang ingin Anda hapus, lalu pilih Hapus Sistem File.
6. Buka EC2 konsol Amazon 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 AndaVPC.

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 Amazon EFS mount helper `amazon-efs-utils`, bukan klien. NFS

amazon-efs-utils Paket ini adalah koleksi sumber terbuka EFS alat Amazon. amazon-efs-utils Koleksinya dilengkapi dengan mount helper dan tooling yang membuatnya lebih mudah untuk mengenkripsi data dalam perjalanan ke Amazon. EFS Untuk informasi lebih lanjut tentang paket ini, lihat [Menginstal EFS klien Amazon](#). 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 7 dan yang lebih baru
- Ubuntu 16.04 LTS dan yang lebih baru

Untuk membangun dan menginstal amazon-efs-utils sebagai RPM paket

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, konfigurasi amazon-efs-utils untuk digunakan Wilayah AWS dengan AWS Direct Connect atauVPN.

Untuk mengkonfigurasi amazon-efs-utils untuk digunakan di Wilayah AWS

1. Menggunakan editor teks pilihan Anda, buka `/etc/amazon/efs/efs-utils.conf` untuk diedit.
2. Temukan garisnya `dns_name_format = {fs_id}.efs.{region}.amazonaws.com`.
3. Ubah `{region}` dengan ID untuk AWS Wilayah Anda, misalnya `us-west-2`.

Untuk me-mount sistem EFS file pada klien lokal Anda, pertama buka terminal pada 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 Wilayah AWS. 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 EC2 instance 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 yang berbeda VPC

Dalam tutorial ini, Anda mengatur EC2 instance untuk me-mount sistem EFS file yang ada di cloud pribadi virtual yang berbeda (VPC). 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 EFS klien Amazon](#).

Klien VPC dan sistem EFS file Anda VPC harus terhubung menggunakan koneksi VPC peering atau gateway VPC transit. Ketika Anda menggunakan koneksi VPC peering atau gateway transit untuk terhubung VPCs, EC2 instance yang ada di satu VPC dapat mengakses sistem EFS file di yang lain VPC, bahkan jika VPCs milik akun yang berbeda.

### 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` Perangkat alat diinstal pada EC2 instance sebelum menggunakan prosedur ini. Untuk petunjuk tentang menginstal `amazon-efs-utils`, lihat [Menginstal EFS klien Amazon](#).
- Salah satu dari yang berikut:
  - Koneksi VPC peering antara VPC tempat sistem EFS file berada dan di VPC mana EC2 instance berada. Koneksi VPC peering 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 yang sama Wilayah AWS atau di antara Wilayah AWS. Untuk informasi selengkapnya, lihat [Membuat dan Menerima Koneksi VPC Peering](#) di Panduan VPC Peering Amazon.
  - Gateway transit yang menghubungkan VPC tempat sistem EFS file berada dan VPC tempat EC2 instance berada. Gateway transit adalah hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. Untuk informasi selengkapnya, lihat [Memulai Gateway Transit di Panduan Gerbang VPC](#) Transit 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 EC2 mount yang berada di Availability Zone yang sama dengan NFS klien Anda. Jika Anda memasang sistem EFS file yang ada di akun lain, pastikan NFS klien dan target EFS mount berada di ID Availability Zone yang sama. Persyaratan ini berlaku karena nama Availability Zone dapat berbeda antar akun.

Untuk menentukan ID Availability Zone dari EC2 instance

1. Terhubung ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Connect ke EC2 instans Anda](#) di Panduan EC2 Pengguna Amazon.
2. Tentukan ID Availability Zone bahwa EC2 instance menggunakan `describe-availability-zones` CLI perintah 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 EC2 instance, 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 describe-mount-targets CLI perintah, 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 EC2 instance yang memetakan alamat IP target mount ke nama host sistem EFS file Anda.

Untuk menambahkan entri host untuk target pemasangan

1. Tambahkan baris untuk alamat IP target mount ke `/etc/hosts` file EC2 instance. 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 VPC keamanan untuk EC2 instance dan target mount memiliki aturan yang memungkinkan akses ke sistem EFS file, sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target](#).

### Langkah 4: Pasang sistem file Anda menggunakan EFS mount helper

Untuk me-mount sistem EFS file Anda, pertama-tama Anda membuat direktori mount pada EC2 instance. Kemudian, menggunakan EFS mount helper, Anda dapat me-mount sistem file dengan otorisasi AWS Identity and Access Management (IAM) atau titik EFS akses. Untuk informasi selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses data sistem file](#) dan [Bekerja dengan titik EFS akses Amazon](#).

Untuk membuat direktori mount

- Buat direktori untuk memasang sistem file menggunakan perintah berikut.

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

Untuk me-mount sistem file menggunakan IAM otorisasi

- Gunakan perintah berikut untuk me-mount sistem file menggunakan IAM otorisasi.

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

Untuk me-mount sistem file menggunakan titik EFS akses

- Gunakan perintah berikut untuk me-mount sistem file menggunakan titik EFS akses.

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

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

Untuk menguji koneksi sistem EFS file

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 EC2 instans 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 instance EC2 Linux yang ada](#).

#### Warning

Gunakan opsi `_netdev`, yang digunakan untuk mengidentifikasi sistem file jaringan, ketika memasang sistem file Anda secara otomatis. Jika `_netdev` tidak ada, EC2 instans 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 pengaturan 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 EFS file dengan perintah berikut.

```
$ sudo umount ~/efs
```

2. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
3. Pilih sistem EFS file yang ingin Anda hapus dari daftar sistem file.
4. Untuk Tindakan, pilih Hapus sistem file.
5. Dalam kotak dialog Hapus sistem file secara permanen, ketik ID sistem EFS file untuk sistem file yang ingin Anda hapus, lalu pilih Hapus Sistem File.
6. Buka EC2 konsol Amazon 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 AndaVPC.

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 EFS file.

### Pemasangan sistem file pada instance Windows gagal

Pemasangan sistem file pada EC2 instance Amazon di Microsoft Windows gagal.

Tindakan yang harus diambil

Jangan gunakan Amazon EFS dengan EC2 instans Windows, 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 NFS klien Anda tidak memiliki izin untuk me-mount sistem file.

Tindakan yang harus diambil

Jika Anda mencoba untuk me-mount sistem file menggunakan IAM, pastikan Anda menggunakan `-o iam` opsi dalam perintah mount Anda. Ini memberitahu EFS mount helper untuk meneruskan kredensialmu ke target mount. EFS Jika Anda masih belum memiliki akses, periksa kebijakan sistem file dan kebijakan identitas Anda untuk memastikan tidak ada DENY klausul yang berlaku untuk koneksi Anda, dan setidaknya ada satu ALLOW klausa yang berlaku untuk koneksi. Untuk informasi



selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses data 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` tidak ada, EC2 instans 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 EFS file Amazon di `/etc/fstab` gagal

Untuk contoh yang menggunakan sistem init `systemd` dengan dua atau lebih EFS entri Amazon 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 EFS mount Amazon.

- 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 EFS mount Amazon, maka ExecStart perintah harus menunjuk NFS4 ke EFS alih-alih 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 EC2 instans Amazon 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 NFS klien](#).

## 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 (AMIs) untuk Amazon Elastic Compute Cloud (AmazonEC2) yang mendukung NFSv4 .1, lihat [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 EFS jalur 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 EFS akses Amazon](#).

Jika Anda tidak menentukan kepemilikan dan izin direktori root, dan direktori root belum ada, tidak EFS 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 Wilayah AWS.

## Tindakan yang harus diambil

Jika Anda membuat dan memasang sistem file secara terprogram, misalnya dengan AWS 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,rsize=1048576,wsize=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 EC2 instans Amazon atau grup keamanan target mount tidak dikonfigurasi dengan benar. Pastikan grup keamanan target mount memiliki aturan masuk yang memungkinkan NFS akses dari grup EC2 keamanan. Untuk informasi selengkapnya, lihat [Membuat grup keamanan](#).

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 DNS nama gagal

Upaya untuk me-mount sistem file menggunakan NFS klien (tidak menggunakan `amazon-efs-utils` klien) menggunakan DNS nama sistem file gagal, seperti yang ditunjukkan dalam contoh berikut:

```
$ sudo mount -t nfs -o
nfsvers=4.1,rsize=1048576,wsize=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 VPC konfigurasi Anda. Jika Anda menggunakan kustomVPC, pastikan DNS pengaturan diaktifkan. Untuk informasi selengkapnya, lihat [DNSatribut untuk Anda VPC](#) di Panduan VPC Pengguna Amazon. Juga, sistem file dan DNS nama target mount tidak dapat diselesaikan dari luar VPC tempat mereka ada.

Sebelum Anda dapat memasang sistem file menggunakan DNS namanya dalam mount perintah, Anda harus melakukan hal berikut:

- Pastikan ada target EFS pemasangan Amazon di Availability Zone yang sama dengan EC2 instans Amazon.
- Pastikan ada target pemasangan yang VPC sama dengan EC2 instans Amazon. Jika tidak, Anda tidak dapat menggunakan resolusi DNS nama untuk target EFS mount yang ada di target lainVPC. Untuk informasi selengkapnya, lihat [Memasang sistem EFS file dari yang lain Akun AWS atau VPC](#).
- Hubungkan EC2 instans Amazon Anda di dalam Amazon yang VPC dikonfigurasi untuk menggunakan DNS server yang disediakan oleh Amazon. Untuk informasi selengkapnya, lihat [set DHCP opsi VPC di Amazon](#) di Panduan VPC Pengguna Amazon.
- Pastikan Amazon VPC dari EC2 instans Amazon yang menghubungkan mengaktifkan DNS nama host. Untuk informasi selengkapnya, lihat [DNSatribut VPC di](#) Panduan VPC Pengguna Amazon.

## Pemasangan sistem file gagal dengan “nfs tidak merespons”

Pemasangan sistem EFS file Amazon gagal pada peristiwa penyambungan ulang Transmission Control Protocol (TCP) dengan "nfs: server\_name still not responding".

### Tindakan yang harus diambil

Gunakan opsi `noresvport` mount untuk memastikan bahwa NFS klien menggunakan port TCP sumber baru saat 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 catatan Sistem Nama Domain (DNS) yang diperlukan untuk target pemasangan sistem file baru jika cloud pribadi virtual (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 `efs.<region>.amazonaws.com` DNS rentang yang bertentangan, buat zona yang dihosting secara terpisah VPC. Untuk informasi selengkapnya tentang DNS pertimbangan VPC, lihat [DNS atribut untuk Anda VPC](#).

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

EFSMount Amazon 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 EFS mount Amazon, 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 EFS file Amazon kadang-kadang dapat terputus karena sejumlah penyebab. NFS klien dirancang untuk terhubung 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 NFS klien yang disertakan dalam versi kernel Linux yang lebih lama (versi v5.4 dan di bawahnya) menyertakan perilaku yang menyebabkan NFS klien, setelah terputus, mencoba menghubungkan kembali pada port sumber yang sama. TCP Perilaku ini tidak sesuai dengan TCPRFC, dan dapat mencegah klien ini dengan cepat membangun kembali koneksi ke NFS server mereka (dalam hal ini, sistem EFS file).

Untuk mengatasi masalah ini, kami sangat menyarankan agar Anda menggunakan Amazon EFS mount helper untuk me-mount sistem EFS file Anda. EFSMount helper menggunakan pengaturan mount yang dioptimalkan untuk sistem EFS file Amazon. Untuk informasi selengkapnya tentang EFS klien dan mount helper, lihat [Menginstal EFS klien Amazon](#).

Jika Anda tidak dapat menggunakan EFS mount helper, kami sangat menyarankan untuk menggunakan opsi `noresvport` NFS mount, yang menginstruksikan NFS klien untuk membangun kembali koneksi menggunakan port TCP sumber baru untuk menghindari masalah ini. Untuk informasi selengkapnya, lihat [Pengaturan NFS pemasangan 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 EC2 instance Amazon terhubung ke satu sistem file dan satu target mount dengan alamat IP tertentu, dan kemudian sistem file dan target mount dihapus. Jika Anda

membuat sistem file baru dan memasang target untuk terhubung ke EC2 instans Amazon tersebut dengan alamat IP target pemasangan 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 instance AmazonEC2. Untuk informasi selengkapnya tentang melepas sistem EFS file Amazon 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 EFS file Amazon Anda. AWS DataSync adalah layanan transfer data online yang dapat menyalin data antara Network File System (NFS), Server Message Block (SMB) server file, penyimpanan objek yang dikelola sendiri, dan juga antar AWS layanan. Untuk informasi selengkapnya tentang penggunaan DataSync dengan AmazonEFS, 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 EFS file Amazon melalui 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 EFS file Amazon Anda untuk kasus penggunaan seperti distribusi data, rantai pasokan, manajemen konten, dan aplikasi penayangan web. Untuk informasi selengkapnya tentang menggunakan Transfer Family dengan AmazonEFS, 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 Sistem File Jaringan (NFS), Server Message Block (SMB) server file, penyimpanan objek yang dikelola sendiri, AWS Snowcone, bucket Amazon S3, sistem file EFS AmazonFSx, dan untuk sistem file Windows File Server.

Anda juga dapat menggunakan DataSync untuk mentransfer file antara dua sistem EFS file, termasuk sistem file dalam Wilayah AWS s yang berbeda dan sistem file yang dimiliki oleh Akun AWS s yang berbeda. Menggunakan DataSync untuk menyalin data antar sistem EFS file, 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 yang dikelola sepenuhnya yang dapat Anda gunakan untuk mentransfer file masuk dan keluar dari sistem EFS file Amazon melalui protokol berikut:

- Secure Shell (SSH) Protokol Transfer File (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 EFS file Anda dari Windows, macOS, dan lingkungan Linux menggunakan SFTP, FTPS, dan FTP klien. Ini membantu memperluas aksesibilitas data Anda di luar NFS klien dan titik akses, kepada pengguna di berbagai lingkungan.

Menggunakan Transfer Family untuk mentransfer data dalam sistem EFS file Amazon diperhitungkan dengan cara yang sama seperti penggunaan klien lainnya. Untuk informasi selengkapnya, lihat [Mode throughput](#) dan [EFSKuota Amazon](#).

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 s yang memiliki sistem EFS file Amazon dengan kebijakan yang memungkinkan akses publik yang dibuat sebelum 6 Januari 2021. Untuk mengaktifkan penggunaan Transfer Family untuk mengakses sistem file Anda, hubungi Support.

### Topik

- [Prasyarat untuk digunakan dengan Amazon AWS Transfer Family EFS](#)

- [Mengkonfigurasi sistem EFS file Anda untuk AWS Transfer Family](#)

## Prasyarat untuk digunakan dengan Amazon AWS Transfer Family EFS

Untuk menggunakan Transfer Family untuk mengakses file sistem EFS file Amazon Anda, konfigurasi Anda harus memenuhi ketentuan berikut:

- Server Transfer Family dan sistem EFS file Amazon Anda berada di tempat yang sama Wilayah AWS.
- IAM kebijakan dikonfigurasi untuk mengaktifkan akses ke IAM peran yang digunakan oleh Transfer Family. Untuk informasi selengkapnya, lihat [Membuat IAM peran dan kebijakan](#) di Panduan AWS Transfer Family 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 EFS file](#).
  - Ubah kebijakan sistem file untuk mengaktifkan akses lintas akun. Untuk informasi selengkapnya, lihat [Mengkonfigurasi akses lintas akun untuk Transfer Family](#).

## Mengkonfigurasi sistem EFS file Anda untuk AWS Transfer Family

Mengonfigurasi sistem EFS file Amazon 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 yang POSIX IDs dialokasikan untuk pengguna Transfer Family.
- Langkah 3. Konfigurasi IAM untuk mengaktifkan akses ke IAM peran 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 EFS file Anda. Tetapkan izin akses ke direktori menggunakan daftar yang POSIX IDs dialokasikan untuk pengguna Transfer Family. Dalam contoh ini, pengguna membuat direktori bernama `transferFam` di bawah titik EFS mount. Membuat direktori adalah opsional, tergantung

pada kasus penggunaan Anda. Jika perlu, Anda dapat memilih nama dan lokasinya pada sistem EFS file.

Untuk menetapkan izin file dan direktori kepada POSIX pengguna untuk Transfer Family

1. Connect ke EC2 instans Amazon Anda. Amazon EFS hanya mendukung pemasangan berdasarkan instans berbasis LinuxEC2.
2. Pasang sistem EFS file Anda jika belum dipasang pada EC2 instance. Untuk informasi selengkapnya, lihat [Memasang sistem EFS file](#).
3. Contoh berikut membuat direktori pada sistem EFS file, dan mengubah grupnya menjadi ID POSIX grup 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 yang POSIX GID ditetapkan `efs/transferFam` untuk pengguna Transfer Family.

```
[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 IAM peran yang digunakan oleh Transfer Family

Di Transfer Family, Anda membuat IAM kebijakan berbasis sumber daya dan IAM peran yang menentukan akses pengguna ke sistem file. EFS Untuk informasi selengkapnya, lihat [Membuat IAM peran dan kebijakan](#) di Panduan AWS Transfer Family Pengguna. Anda harus memberikan akses

IAM peran Transfer Family ke sistem EFS file Anda menggunakan kebijakan IAM identitas atau kebijakan sistem file.

Berikut ini adalah contoh kebijakan sistem file yang memberikan ClientMount (baca) dan ClientWrite akses ke IAM peranEFS-role-for-transfer.

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

Untuk informasi selengkapnya tentang membuat kebijakan sistem file, lihat [Membuat kebijakan sistem file](#). Untuk informasi selengkapnya tentang penggunaan IAM kebijakan berbasis identitas untuk mengelola akses ke EFS sumber daya, lihat. [Kebijakan berbasis identitas untuk Amazon EFS](#)

## Mengkonfigurasi 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 bersifat non-publik. Untuk informasi selengkapnya tentang memblokir akses publik ke sistem file Anda, lihat [Memblokir akses publik ke sistem EFS file](#).

Anda dapat memberikan Akun AWS akses berbeda ke sistem file Anda dalam kebijakan sistem file. Di EFS konsol Amazon, 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 Amazon Resource Name (ARN). Untuk informasi selengkapnya ARNs, lihat [IAM ARNs](#) di Panduan IAM Pengguna.

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 NFS klien 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 ini di akun Anda. EFS

```
{
  "Statement": [
    {
      "Sid": "NFS-client-read-write-via-fsmt",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "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 IAM peran yang digunakan oleh Transfer Family.

```

{
  "Statement": [
    {
      "Sid": "NFS-client-read-write-via-fsmt",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "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 EFS file

Tugas manajemen sistem file termasuk mengelola aksesibilitas jaringan sistem file dengan target mount, mengubah mode throughputnya, memperbarui kebijakan lifecycle, mengelola enkripsi, dan mengelola biaya sistem file menggunakan AWS Budgets

Anda dapat melakukan tugas-tugas manajemen sistem file ini menggunakan AWS Management Console, atau secara terprogram menggunakan AWS Command Line Interface (AWS CLI) atau API, seperti yang dibahas di bagian berikut.

Topik

- [Mengelola target mount](#)
- [Mengelola throughput sistem file](#)
- [Mengelola siklus hidup penyimpanan untuk sistem file EFS](#)
- [Mengelola akses ke sistem file terenkripsi](#)
- [Mengelola biaya sistem EFS file menggunakan AWS Budgets](#)
- [Memahami status sistem berkas](#)

## Mengelola target mount

Anda memasang sistem file di Amazon EC2 atau instance AWS komputasi lainnya di virtual private cloud (VPC) menggunakan target mount yang Anda buat untuk sistem file. Mengelola aksesibilitas jaringan sistem file mengacu pada pengelolaan target mount sistem file.

Setelah Anda membuat sistem EFS file Amazon, Anda dapat membuat target mount. Untuk sistem EFS file Amazon yang menggunakan kelas penyimpanan Regional, Anda dapat membuat target pemasangan di setiap Availability Zone dalam file Wilayah AWS. 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 instance komputasi, termasuk Amazon, EC2 Amazon ECS, dan AWS Lambda di cloud pribadi virtual Anda (VPC).

Diagram berikut menunjukkan sistem file Regional dengan target mount yang dibuat di semua Availability Zone di VPC. Ilustrasi menunjukkan tiga EC2 contoh yang diluncurkan di VPC subnet berbeda yang mengakses sistem file Amazon 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, EC2 instance yang diluncurkan di salah satu subnetnya dapat berbagi target pemasangan yang sama.

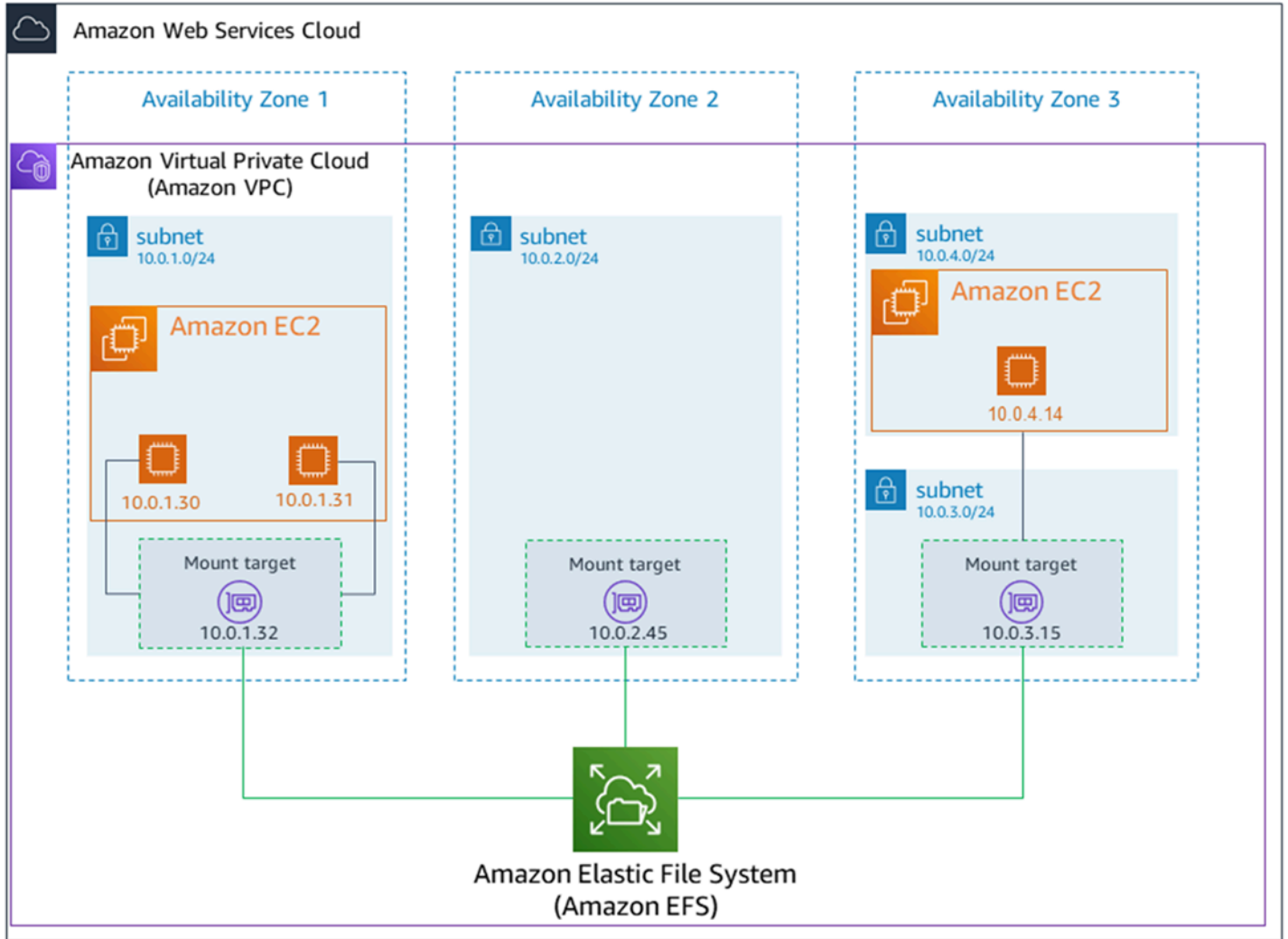
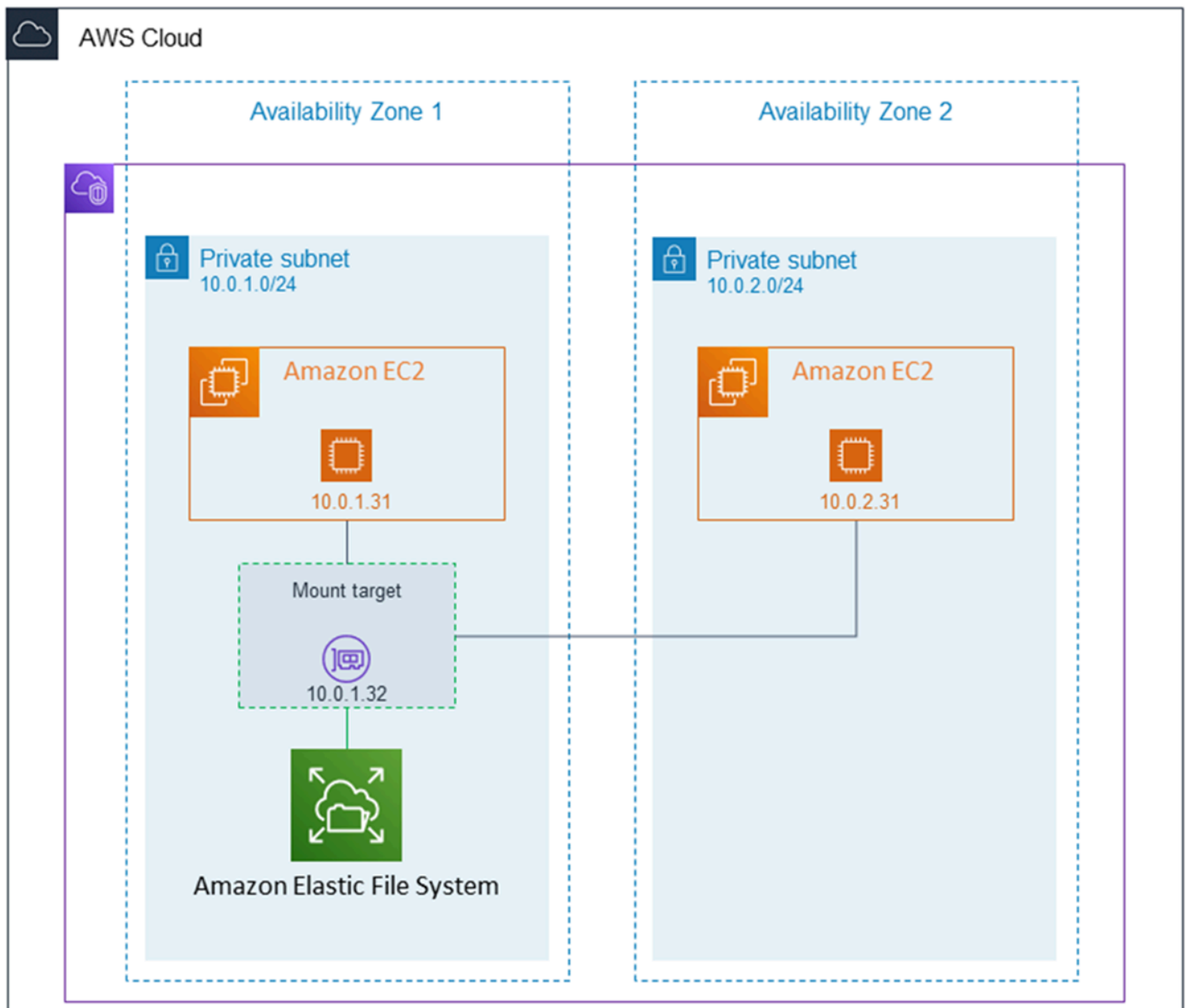


Diagram berikut menunjukkan sistem file One Zone, dengan target mount tunggal yang dibuat di Availability Zone yang sama dengan sistem file. Mengakses sistem file dengan menggunakan EC2 instance di us-west2c Availability Zone menimbulkan biaya akses data karena terletak di Availability Zone yang berbeda 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.
- NFS-tingkat pertimbangan izin.

Awalnya, hanya pengguna root pada EC2 instance Amazon yang memiliki read-write-execute izin pada sistem file. Topik ini membahas izin NFS tingkat -dan memberikan contoh yang menunjukkan

kepada Anda cara memberikan izin dalam skenario umum. Untuk informasi selengkapnya, lihat [Network File System \(NFS\) tingkat pengguna, grup, dan izin](#).

Mengelola target mount mengacu pada kegiatan ini:

- Membuat dan menghapus target mount di VPC - Minimal, Anda harus membuat target mount di setiap Availability Zone tempat Anda ingin mengakses sistem file.
- Memperbarui konfigurasi target mount — Saat membuat target mount, Anda mengaitkan grup keamanan dengan target mount. Grup keamanan bertindak sebagai firewall virtual yang mengontrol lalu lintas ke dan dari target mount. Anda dapat menambahkan aturan masuk untuk mengontrol akses ke target mount, dan dengan demikian sistem file. Setelah membuat target mount, Anda mungkin ingin memodifikasi grup keamanan yang ditetapkan untuk mereka.

Anda dapat membuat target mount untuk sistem file dengan menggunakan AWS Management Console, AWS CLI, atau secara terprogram dengan menggunakan file. AWS SDKs Saat menggunakan konsol, Anda dapat membuat target mount saat pertama kali membuat sistem file atau setelah sistem file dibuat. Untuk petunjuk membuat target mount menggunakan EFS konsol Amazon saat membuat sistem file, lihat [Buat sistem file dengan pengaturan khusus \(konsol\)](#).


## Kelola target pemasangan (konsol)

Gunakan prosedur berikut untuk menambah atau memodifikasi target mount untuk sistem EFS file Amazon yang ada.

Untuk mengelola target pemasangan pada sistem EFS file Amazon


1. Masuk ke AWS Management Console dan buka EFS konsol Amazon di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file. Halaman sistem File menampilkan sistem EFS file di akun Anda.
3. Pilih sistem file yang ingin Anda kelola target mount dengan memilih Namanya atau ID sistem File untuk menampilkan halaman detail sistem file.
4. Pilih Jaringan untuk menampilkan daftar target pemasangan yang ada.
5. Pilih Kelola untuk menampilkan halaman Availability Zone dan membuat modifikasi.

Di halaman ini, untuk target pemasangan yang ada, Anda dapat menambahkan dan menghapus grup keamanan, atau menghapus target pemasangan. Anda juga dapat membuat target pemasangan baru.

 Note

Untuk sistem file One Zone, Anda hanya dapat membuat target mount tunggal yang berada di Availability Zone yang sama dengan sistem file.

- Untuk menghapus grup keamanan dari target pemasangan, pilih X di sebelah ID grup keamanan.
- Untuk menambahkan grup keamanan ke target pemasangan, pilih Pilih grup keamanan untuk menampilkan daftar grup keamanan yang tersedia. Atau, masukkan ID grup keamanan di bidang pencarian di bagian atas daftar.
- Untuk mengantrekan target pemasangan untuk dihapus, pilih Hapus.

 Note

Sebelum menghapus target mount, pertama unmount sistem file.

- Untuk menambahkan target pemasangan, pilih Tambahkan target pemasangan. Opsi ini hanya tersedia untuk sistem file yang menggunakan kelas penyimpanan EFS Regional, dan jika target mount belum ada di setiap Availability Zone untuk Wilayah AWS.
6. Pilih Simpan untuk menyimpan perubahan apa pun.

VPCUntuk mengubah sistem EFS file Amazon (konsol)

VPCUntuk mengubah konfigurasi jaringan sistem file, Anda harus menghapus semua target pemasangan sistem file yang ada.

1. Buka konsol Amazon Elastic File System di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file. Halaman sistem File menunjukkan sistem EFS file di akun Anda.
3. Untuk sistem file yang ingin Anda ubah, pilih Nama atau ID sistem File. VPC Halaman detail sistem file ditampilkan.

4. Pilih Jaringan untuk menampilkan daftar target pemasangan yang ada.
5. Pilih Kelola. Halaman Availability zone muncul.
6. Hapus semua target pemasangan yang ditampilkan di halaman.
7. Pilih Simpan untuk menyimpan perubahan dan menghapus target pemasangan. Tab Jaringan menunjukkan status target pemasangan sebagai penghapusan.
8. Saat semua status target pemasangan ditampilkan sebagai dihapus, pilih Kelola. Halaman Availability Zone muncul.
9. Pilih yang baru VPC dari daftar Virtual Private Cloud (VPC).
10. Pilih Tambahkan target pemasangan untuk menambahkan target pemasangan baru. Untuk setiap target mount yang Anda tambahkan, masukkan yang berikut ini:
  - Zona ketersediaan
  - Sebuah ID Subnet
  - Alamat IP, atau tetap disetel ke Otomatis
  - Satu atau lebih kelompok keamanan
11. Pilih Simpan untuk mengimplementasikan VPC dan memasang perubahan target.

## Mengelola target mount (CLI)

### Note

Untuk sistem file One Zone, Anda hanya dapat membuat target mount tunggal yang berada di Availability Zone yang sama dengan sistem file.

### Untuk membuat target mount (CLI)

- Untuk membuat target mount, gunakan `create-mount-target` CLI perintah (operasi yang sesuai adalah [CreateMountTarget](#)), seperti yang ditunjukkan 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 aws-region \  
--profile adminuser
```

Contoh berikut menunjukkan perintah dengan data sampel.

```
$ aws efs create-mount-target \  
--file-system-id fs-0123467 \  
--subnet-id subnet-b3983dc4 \  
--security-group sg-01234567 \  
--region us-east-2 \  
--profile adminuser
```

Setelah berhasil membuat target mount, Amazon EFS mengembalikan deskripsi target mount JSON seperti yang ditunjukkan pada contoh berikut.

```
{  
  "MountTargetId": "fsmt-f9a14450",  
  "NetworkInterfaceId": "eni-3851ec4e",  
  "FileSystemId": "fs-b6a0451f",  
  "LifeCycleState": "available",  
  "SubnetId": "subnet-b3983dc4",  
  "OwnerId": "23124example",  
  "IpAddress": "10.0.1.24"  
}
```

Untuk mengambil daftar target mount untuk sistem file () CLI

- Anda juga dapat mengambil daftar target mount yang dibuat untuk sistem file dengan menggunakan [describe-mount-targets](#) CLI perintah (operasi yang sesuai adalah [DescribeMountTargets](#)), seperti yang ditunjukkan berikut.

```
$ aws efs describe-mount-targets --file-system-id fs-a576a6dc
```

```
{  
  "MountTargets": [  
    {  
      "OwnerId": "111122223333",  
      "MountTargetId": "fsmt-48518531",  
      "FileSystemId": "fs-a576a6dc",  
      "SubnetId": "subnet-88556633",  
      "LifeCycleState": "available",  
      "IpAddress": "172.31.25.203",
```

```

    "NetworkInterfaceId": "eni-0123456789abcdef1",
    "AvailabilityZoneId": "use2-az2",
    "AvailabilityZoneName": "us-east-2b"
  },
  {
    "OwnerId": "111122223333",
    "MountTargetId": "fsmt-5651852f",
    "FileSystemId": "fs-a576a6dc",
    "SubnetId": "subnet-44223377",
    "LifecycleState": "available",
    "IpAddress": "172.31.46.181",
    "NetworkInterfaceId": "eni-0123456789abcdefa",
    "AvailabilityZoneId": "use2-az3",
    "AvailabilityZoneName": "us-east-2c"
  },
  {
    "OwnerId": "111122223333",
    "MountTargetId": "fsmt-5751852e",
    "FileSystemId": "fs-a576a6dc",
    "SubnetId": "subnet-a3520bcb",
    "LifecycleState": "available",
    "IpAddress": "172.31.12.219",
    "NetworkInterfaceId": "eni-0123456789abcdef0",
    "AvailabilityZoneId": "use2-az1",
    "AvailabilityZoneName": "us-east-2a"
  }
]
}

```

Untuk menghapus target mount (CLI) yang ada

- 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 \  

```

Untuk memodifikasi grup keamanan target mount yang ada

- 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
```

Untuk informasi selengkapnya, lihat [Tutorial: Buat sistem EFS file dan pasang pada EC2 instance menggunakan AWS CLI](#).

## Membuat atau menghapus target mount di a VPC

Untuk mengakses sistem EFS file Amazon di aVPC, Anda perlu memasang target. Untuk sistem EFS file Amazon, berikut ini benar:

- Anda dapat membuat satu target mount di setiap Availability Zone.

- Jika VPC memiliki beberapa subnet di Availability Zone, Anda dapat membuat target mount hanya di salah satu subnet tersebut. Semua EC2 instance di Availability Zone dapat berbagi target pemasangan tunggal.

#### Note

Kami menyarankan Anda membuat target pemasangan di setiap Availability Zone. Ada pertimbangan biaya untuk memasang sistem file pada EC2 instance di Availability Zone melalui target mount yang dibuat di Availability Zone lain. Untuk informasi selengkapnya, lihat [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.

Jika Anda menghapus target mount, 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 [Mengelola target mount](#).

#### Note

Sebelum menghapus target mount, pertama unmount sistem file. Untuk informasi selengkapnya, lihat [Melepaskan sistem file](#).

Anda dapat menggunakan sistem file hanya dalam satu VPC per satu. Artinya, Anda dapat membuat target mount untuk sistem file dalam satu VPC per satu. Jika Anda ingin mengakses sistem file dari yang lain VPC, pertama-tama hapus target mount dari saat ini VPC. Kemudian buat target mount baru di tempat lain VPC.

Dengan menggunakan AWS Management Console, the AWS CLI, and the API, Anda dapat membuat dan mengelola target mount pada sistem file. Untuk target pemasangan yang ada, Anda dapat menambahkan dan menghapus grup keamanan, atau menghapus target pemasangan. Untuk informasi selengkapnya, lihat [Mengelola target mount](#).

## VPC Mengubah target mount Anda

Anda dapat menggunakan sistem EFS file Amazon dalam satu VPC berdasarkan VPC layanan Amazon pada satu waktu. Artinya, Anda membuat target mount di a VPC untuk sistem file Anda, dan menggunakan target mount tersebut untuk menyediakan akses ke sistem file.

Anda dapat memasang sistem EFS file Amazon dari target ini:

- EC2 Contoh Amazon dalam hal yang sama VPC
- EC2 contoh dalam VPC terhubung dengan mengintip VPC
- Server lokal dengan menggunakan AWS Direct Connect
- Server lokal melalui jaringan pribadi AWS virtual (VPN) dengan menggunakan Amazon VPC

Koneksi VPC peering adalah koneksi jaringan antara dua VPCs yang memungkinkan Anda untuk mengarahkan lalu lintas di antara mereka. Koneksi dapat menggunakan alamat Private Internet Protocol versi 4 (IPv4) atau Internet Protocol versi 6 (IPv6). Untuk informasi lebih lanjut tentang cara Amazon EFS bekerja dengan VPC peering, lihat [Memasang sistem EFS file dari yang lain Akun AWS atau VPC](#).

Untuk mengakses sistem file dari EC2 instance lain VPC, Anda harus:

- Hapus target pemasangan saat ini.
- Ubah VPC.
- Buat target mount baru.

Untuk informasi lebih lanjut tentang melakukan langkah-langkah ini di AWS Management Console, lihat [VPC Untuk mengubah sistem EFS file Amazon \(konsol\)](#).

## Menggunakan CLI

Untuk menggunakan sistem file di sistem lain VPC, pertama-tama hapus target mount yang sebelumnya Anda buat di file VPC. Kemudian buat target mount baru di tempat lain VPC. Misalnya AWS CLI perintah, lihat [Mengelola target mount \(CLI\)](#).

## Memperbarui konfigurasi target mount

Setelah Anda membuat target mount untuk sistem file Anda, Anda mungkin ingin memperbarui grup keamanan yang berlaku. Anda tidak dapat mengubah alamat IP dari target mount yang ada. Untuk


mengubah alamat IP, hapus target mount dan buat yang baru dengan alamat baru. Menghapus target mount merusak semua pemasangan sistem file yang ada.

 Note

Sebelum menghapus target mount, pertama unmount sistem file.

Setiap target mount juga memiliki alamat IP. Saat Anda membuat target pemasangan, Anda dapat memilih alamat IP dari subnet tempat Anda menempatkan target pemasangan. Jika Anda menghilangkan nilai, Amazon EFS memilih alamat IP yang tidak digunakan dari subnet tersebut.

Tidak ada EFS operasi Amazon untuk mengubah alamat IP setelah membuat target mount. Dengan demikian, Anda tidak dapat mengubah alamat IP secara terprogram atau dengan menggunakan AWS CLI. Tetapi konsol memungkinkan Anda untuk mengubah alamat IP. Di belakang layar, konsol menghapus target mount dan membuat target mount lagi.

 Warning

Jika Anda mengubah alamat IP dari target mount, Anda merusak pemasangan sistem file yang ada, dan Anda harus memasang kembali sistem file.

Tak satu pun dari perubahan konfigurasi untuk aksesibilitas jaringan sistem file mempengaruhi sistem file itu sendiri. Sistem file dan data Anda tetap tidak berubah.

## Memodifikasi grup keamanan

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 EC2 instance Anda untuk berkomunikasi dengan sistem file.

Untuk informasi selengkapnya tentang grup keamanan, lihat [Grup EC2 keamanan Amazon untuk instans Linux](#) di Panduan EC2 Pengguna Amazon.

Untuk mengubah grup keamanan target mount, lihat [Mengelola target mount](#).

## Mengelola throughput sistem file

Elastis adalah mode throughput default 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.

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 EFS throughput Amazon, 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 EFS konsol Amazon, AWS Command Line Interface (AWS CLI), dan Amazon EFSAPI.

Untuk mengelola throughput sistem file (konsol)

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 EFS file 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 sistem file 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.

Untuk mengelola throughput sistem file () CLI

- Gunakan [update-file-system](#) CLI perintah, atau [UpdateFileSystem](#) API tindakan untuk mengubah mode throughput sistem file.

## Mengelola siklus hidup penyimpanan untuk sistem file EFS

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 Akses Jarang (IA) dan Arsip. EFS EFS 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 POSIX file yang dapat dilihat publik). Setiap kali file dalam Standar diakses, pengatur waktu manajemen siklus hidup diatur ulang.

Kebijakan siklus hidup berlaku untuk seluruh sistem EFS file.

Kebijakan EFS siklus hidup adalah:

- Transisi ke IA - Menginstruksikan manajemen siklus hidup kapan harus memindahkan file ke penyimpanan Akses Jarang, yang dioptimalkan biaya untuk data yang diakses hanya beberapa kali setiap kuartal. Secara default, file yang tidak diakses dalam penyimpanan Standar selama 30 hari dialihkan ke IA.
- Transisi ke Arsip - Menginstruksikan manajemen siklus hidup kapan harus memindahkan file ke kelas penyimpanan Arsip, yang dioptimalkan biaya untuk data yang diakses hanya beberapa kali setiap tahun atau kurang. Secara default, file yang tidak diakses dalam penyimpanan Standar selama 90 hari dialihkan ke Arsip.
- Transisi ke Standar - Menginstruksikan manajemen siklus hidup apakah akan mentransisikan file dari IA atau Arsip dan kembali ke penyimpanan Standar saat file diakses di penyimpanan IA atau Arsip. Secara default, file tidak dipindahkan kembali ke 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 EFS file. Waktu yang diperlukan untuk mentransisikan file ke dalam atau keluar dari penyimpanan IA dan Arsip bervariasi tergantung pada ukuran file dan beban kerja sistem file.

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 EFS file yang memiliki pengaturan yang disarankan menggunakan AWS Management Console, 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 AWS Management Console 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.

Mengonfigurasi kebijakan siklus hidup untuk sistem file yang ada (konsol)

Anda dapat menggunakan AWS Management Console untuk mengatur kebijakan siklus hidup untuk sistem file yang ada.

1. Masuk ke AWS Management Console dan buka EFS konsol Amazon 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.

Mengonfigurasi kebijakan siklus hidup untuk sistem file yang ada () CLI

Anda dapat menggunakan AWS CLI untuk mengatur atau memodifikasi kebijakan siklus hidup sistem file.

- Jalankan [put-lifecycle-configuration](#) AWS CLI perintah atau [PutLifecycleConfiguration](#) API perintah, 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": []
}
```

## Mengelola akses ke sistem file terenkripsi

Menggunakan AmazonEFS, Anda dapat membuat sistem file terenkripsi. Amazon EFS mendukung dua bentuk enkripsi untuk sistem file, enkripsi dalam perjalanan dan enkripsi saat istirahat. Manajemen kunci apa pun yang perlu Anda lakukan hanya terkait dengan enkripsi saat istirahat. Amazon EFS secara otomatis mengelola kunci untuk enkripsi dalam perjalanan.

Jika Anda membuat sistem file yang menggunakan enkripsi saat istirahat, data dan metadata dienkripsi saat istirahat. Amazon EFS menggunakan AWS Key Management Service (AWS KMS) untuk manajemen kunci. Saat Anda membuat sistem file menggunakan enkripsi saat istirahat, Anda menentukan file AWS KMS key. KMSKuncinya bisa `aws/elasticfilesystem` (Kunci yang dikelola AWS untuk AmazonEFS), atau bisa juga menjadi kunci yang dikelola pelanggan yang Anda kelola.

Data file — isi file Anda — dienkripsi saat istirahat menggunakan KMS kunci yang Anda tentukan saat Anda membuat sistem file Anda. Metadata—nama file, nama direktori, dan konten direktori— dienkripsi menggunakan kunci yang dikelola Amazon. EFS

EFS Kunci yang dikelola AWS Untuk sistem file Anda digunakan sebagai KMS kunci untuk mengenkripsi metadata dalam sistem file Anda, misalnya nama file, nama direktori, dan konten direktori. Anda memiliki kunci terkelola pelanggan yang digunakan untuk mengenkripsi data file (isi file Anda) saat istirahat.

Anda mengelola siapa yang memiliki akses ke KMS kunci Anda dan konten sistem file terenkripsi Anda. Akses ini dikendalikan oleh AWS Identity and Access Management (IAM) kebijakan dan AWS KMS. IAMkebijakan mengontrol akses pengguna ke EFS API tindakan Amazon. AWS KMS kebijakan kunci mengontrol akses pengguna ke KMS kunci yang Anda tentukan saat sistem file dibuat. Untuk informasi selengkapnya, lihat berikut ini:

- [IAMPengguna](#) di Panduan IAM Pengguna
- [Kebijakan utama AWS KMS dalam](#) Panduan AWS Key Management Service Pengembang
- [Hibah AWS KMS di](#) Panduan AWS Key Management Service Pengembang.

Sebagai administrator kunci, Anda dapat mengimpor kunci eksternal. Anda juga dapat memodifikasi kunci dengan mengaktifkannya, menonaktifkannya, atau menghapusnya. Status KMS kunci yang Anda tentukan (saat Anda membuat sistem file dengan enkripsi saat istirahat) memengaruhi akses ke isinya. KMSKunci harus dalam `enabled` keadaan agar pengguna memiliki akses ke konten sistem `encrypted-at-rest` file yang dienkripsi menggunakan kunci itu.

## Mengelola KMS kunci untuk sistem EFS file

Anda dapat menonaktifkan atau menghapus KMS kunci yang dikelola pelanggan, atau Anda dapat mencabut EFS akses Amazon ke kunci AndaKMS. Menonaktifkan dan mencabut akses Amazon ke kunci Anda adalah tindakan yang EFS dapat dibalik. Berhati-hatilah saat menghapus KMS kunci. Menghapus KMS kunci adalah tindakan yang tidak dapat diubah.

Jika Anda menonaktifkan atau menghapus KMS kunci yang digunakan untuk sistem file yang dipasang, berikut ini benar:

- KMSKunci itu tidak dapat digunakan sebagai kunci untuk sistem `encrypted-at-rest` file baru.
- Sistem `encrypted-at-rest` file yang ada yang menggunakan KMS kunci itu berhenti bekerja setelah jangka waktu tertentu.

Jika Anda mencabut EFS akses Amazon ke hibah untuk sistem file terpasang yang ada, perilakunya sama seperti jika Anda menonaktifkan atau menghapus kunci terkait KMS. Dengan kata lain, sistem encrypted-at-rest file terus berfungsi, tetapi berhenti bekerja setelah jangka waktu tertentu.

Anda dapat mencegah akses ke sistem encrypted-at-rest file yang dipasang yang memiliki KMS kunci yang Anda nonaktifkan, hapus, atau cabut EFS akses Amazon. Untuk melakukan ini, lepaskan sistem file dan hapus target EFS pemasangan Amazon Anda.

Anda tidak dapat segera menghapus AWS KMS key, tetapi Anda dapat menjadwalkannya untuk dihapus dalam 7-30 hari. Sementara KMS kunci dijadwalkan untuk dihapus, Anda tidak dapat menggunakannya untuk operasi kriptografi. Anda juga dapat membatalkan penghapusan terjadwal KMS kunci.

Untuk mempelajari cara menonaktifkan dan mengaktifkan kembali KMS kunci terkelola pelanggan, lihat [Mengaktifkan dan menonaktifkan kunci](#) di Panduan Pengembang. AWS Key Management Service Untuk mempelajari cara menjadwalkan penghapusan KMS kunci terkelola pelanggan, lihat [Menghapus KMS kunci](#) di Panduan Pengembang AWS Key Management Service .

## Mengelola biaya sistem EFS file menggunakan AWS Budgets

Anda dapat merencanakan dan mengelola biaya sistem EFS file Amazon Anda dengan menggunakan AWS Anggaran.

Anda dapat bekerja dengan AWS Anggaran dari AWS Billing and Cost Management konsol. Untuk menggunakan AWS Anggaran, Anda membuat anggaran biaya bulanan untuk sistem EFS file Anda. Anda dapat mengatur anggaran Anda untuk memberi tahu Anda jika biaya Anda diperkirakan melebihi jumlah anggaran Anda, dan kemudian membuat penyesuaian untuk mempertahankan anggaran Anda sesuai kebutuhan.

Ada biaya yang terkait dengan penggunaan AWS Anggaran. Untuk reguler Akun AWS, dua anggaran pertama Anda gratis. Untuk informasi selengkapnya tentang AWS Anggaran, termasuk biaya, lihat [Mengelola Biaya Anda dengan Anggaran](#) di AWS Billing Panduan Pengguna.

Anda dapat mengatur anggaran khusus untuk EFS biaya dan penggunaan Amazon Anda di tingkat akun Wilayah AWS, layanan, atau tag dengan menggunakan parameter anggaran. Di bagian berikut, Anda dapat menemukan deskripsi tingkat tinggi tentang cara mengatur anggaran biaya pada sistem EFS file dengan AWS Anggaran. Anda melakukannya dengan menggunakan tag alokasi biaya.

## Prasyarat

Untuk melakukan prosedur yang dirujuk di bagian berikut, pastikan Anda memiliki yang berikut:

- Sebuah sistem EFS file
- Kebijakan AWS Identity and Access Management (IAM) dengan izin berikut:
  - Akses ke AWS Billing and Cost Management konsol.
  - Kemampuan untuk melakukan `elasticfilesystem:CreateTags` dan `elasticfilesystem:DescribeTags` tindakan.

## Membuat anggaran biaya bulanan untuk sistem EFS file

Membuat anggaran biaya bulanan untuk sistem EFS file Amazon Anda menggunakan tag adalah proses tiga langkah.

Untuk membuat anggaran biaya bulanan untuk sistem EFS file Anda menggunakan tag

1. Buat tag yang akan digunakan untuk mengidentifikasi sistem file yang ingin Anda lacak biayanya. Untuk mempelajari caranya, lihat [Sumber daya penandaan EFS](#).
2. Di konsol Billing and Cost Management, aktifkan tag sebagai tag alokasi biaya. Untuk prosedur terperinci, lihat [Mengaktifkan tag alokasi biaya yang ditentukan pengguna](#) di Panduan Pengguna.AWS Billing
3. Di konsol Billing and Cost Management, di bawah Budgets, buat anggaran biaya bulanan dalam Anggaran AWS . Untuk prosedur terperinci, lihat [Membuat anggaran](#) di Panduan AWS Billing Pengguna.

Setelah membuat anggaran biaya EFS bulanan, Anda dapat melihatnya di dasbor Anggaran, yang menampilkan data anggaran berikut:

- Biaya dan penggunaan Anda saat ini dikeluarkan untuk anggaran selama periode anggaran.
- Biaya anggaran Anda untuk periode anggaran.
- Biaya perkiraan Anda untuk periode anggaran.
- Persentase yang menunjukkan biaya Anda dibandingkan dengan jumlah anggaran Anda.
- Persentase yang menunjukkan biaya perkiraan Anda dibandingkan dengan jumlah anggaran Anda.

Untuk informasi selengkapnya tentang melihat anggaran EFS biaya, lihat [Melihat anggaran Anda](#) di Panduan AWS Billing Pengguna.

## Memahami status sistem berkas

Anda dapat melihat status sistem EFS file Amazon menggunakan EFS konsol Amazon atau file AWS CLI. Sistem EFS file Amazon dapat memiliki salah satu nilai status yang dijelaskan dalam tabel berikut.

Status sistem berkas	Deskripsi
AVAILABLE	Sistem file dalam keadaan sehat, dan dapat dijangkau dan tersedia untuk digunakan.
CREATING	Amazon EFS sedang dalam proses membuat sistem file baru.
DELETING	Amazon EFS menghapus sistem file sebagai respons terhadap permintaan penghapusan yang diprakarsai pengguna. Untuk informasi selengkapnya, lihat <a href="#">Menghapus sistem EFS file</a> .
DELETED	Amazon EFS telah menghapus sistem file sebagai tanggapan atas permintaan penghapusan yang diprakarsai pengguna. Untuk informasi selengkapnya, lihat <a href="#">Menghapus sistem EFS file</a> .
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 <a href="#">Melindungi data Anda di Amazon EFS</a>

# Memantau 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 EFS, sebelum Anda mulai memantau Amazon, buat rencana pemantauan yang mencakup jawaban atas pertanyaan-pertanyaan berikut:

- Apa tujuan 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 EFS kinerja Amazon 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 pemantauan](#)
- [Bagaimana Amazon EFS melaporkan sistem file dan ukuran objek](#)
- [Melihat ukuran kelas penyimpanan](#)
- [Memantau metrik dengan Amazon CloudWatch](#)
- [Mencatat EFS API panggilan Amazon dengan AWS CloudTrail](#)

## Alat pemantauan

AWS menyediakan berbagai alat yang dapat Anda gunakan untuk memantau AmazonEFS. Anda dapat mengonfigurasi beberapa alat ini untuk melakukan pemantauan untuk Anda, tetapi beberapa alat memerlukan intervensi manual. Kami menyarankan agar Anda mengotomatiskan 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 (AmazonSNS) 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 [Memantau File Log](#) 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 pemantauan manual

Bagian penting lainnya dari pemantauan Amazon EFS melibatkan pemantauan secara manual item-item yang tidak dicakup oleh CloudWatch alarm Amazon. AmazonEFS, CloudWatch, dan AWS Management Console dasbor lainnya memberikan at-a-glance pandangan tentang keadaan AWS lingkungan Anda. Kami menyarankan Anda juga memeriksa file log pada sistem file.

- Dari EFS konsol Amazon, 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 menemukan tren.
- Cari dan telusuri semua metrik AWS sumber daya Anda.
- Buat dan sunting alarm untuk menerima pemberitahuan tentang 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.

### Mengukur objek sistem EFS file Amazon

Objek yang dapat Anda lihat dalam EFS sistem Amazon termasuk file biasa, direktori, tautan simbolis, dan file khusus (FIFO 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 cara 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 `B`locks 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 EFS file Amazon

Ukuran terukur dari sistem EFS file Amazon mencakup jumlah ukuran semua objek saat ini di semua kelas EFS penyimpanan. 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 EFS file Amazon dengan cara berikut:

- Menggunakan [describe-file-systems](#) AWS CLI perintah dan [DescribeFileSystemAPI](#) operasi, respon meliputi yang berikut:

```
"SizeInBytes":{
  "Timestamp": 1403301078,
  "Value": 29313744866,
  "ValueInIA": 675432,
  "ValueInStandard": 29312741784
  "ValueInArchive": 327650
}
```

[Di mana ukuran terukur juga ValueInStandard digunakan untuk menentukan baseline throughput I/O dan kecepatan burst untuk sistem file menggunakan mode Bursting Throughput.](#)

- 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 sebuah EC2 instance.

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 penyimpanan EFS Infrequent Access (IA) dan Arsip diukur dalam kenaikan 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 I/O sistem 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 EFS file Amazon menggunakan pengukur mode throughput elastis dan metadata tagihan dibaca sebagai operasi baca dan metadata ditulis sebagai operasi tulis. Operasi metadata diukur dalam peningkatan 4 KiB dan operasi data diukur dalam peningkatan 32 KiB.

## Pengukuran Throughput yang disediakan

Untuk sistem file yang menggunakan mode throughput Provisioned, Anda hanya membayar untuk jumlah waktu throughput diaktifkan. Amazon EFS meter sistem file dengan mode throughput Provisioned diaktifkan sekali setiap jam. 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 EFS konsol Amazon, konsol AWS CLI, atau file EFSAPI.

### Melihat ukuran data penyimpanan (EFSkonsol Amazon)

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 Standar.
- Ukuran di IA adalah ukuran (dalam byte biner) data yang disimpan di kelas penyimpanan Akses EFS Jarang. File yang lebih kecil dari 128KiB dibulatkan hingga 128KiB.
- Ukuran dalam Arsip adalah ukuran (dalam byte biner) data yang disimpan di kelas penyimpanan EFS Arsip. File yang lebih kecil dari 128KiB dibulatkan hingga 128KiB.

Anda juga dapat melihat Storage bytes metrik pada tab Monitoring pada halaman rincian sistem File di EFS konsol Amazon. Untuk informasi selengkapnya, lihat [Mengakses CloudWatch metrik untuk Amazon EFS](#).

### Melihat ukuran data penyimpanan (AWS CLI)

Anda dapat melihat berapa banyak data yang disimpan di setiap kelas penyimpanan sistem file Anda menggunakan AWS CLI atau EFSAPI. Lihat detail penyimpanan data dengan memanggil `describe-file-systems` CLI perintah (API operasi 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 [Mengukur objek sistem EFS file Amazon](#).

## Memantau metrik dengan Amazon CloudWatch

Anda dapat memantau sistem file menggunakan Amazon CloudWatch, yang mengumpulkan dan memproses data mentah dari Amazon EFS menjadi metrik yang dapat dibaca, mendekati waktu nyata. 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 EFS metrik Amazon dikirim secara otomatis CloudWatch pada periode 1 menit, kecuali dicatat untuk beberapa metrik individual. EFSKonsol Amazon 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 [Panduan CloudWatch Pengguna Amazon](#).

EFS CloudWatch Metrik Amazon 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

EFSMetrik Amazon menggunakan EFS namespace. Namespace AWS/EFS mencakup metrik berikut. Semua metrik kecuali untuk `TimeSinceLastSync` satu dimensi. `FileSystemId` ID sistem file dapat ditemukan di EFS konsol Amazon, 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` dimensi — ID sistem file sumber dalam konfigurasi replikasi.
- `DestinationFileSystemId` dimensi — ID sistem file tujuan dalam konfigurasi replikasi.

Unit: detik

Statistik yang valid: Minimum, Maximum, Average

### **PercentIOLimit**

Menunjukkan seberapa dekat sistem file untuk mencapai batas I/O dari 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: Bit

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 Standar 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 mode 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 diukur pada sepertiga tingkat operasi lainnya.

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`, `MaximumAverage`, dan `Sum` statistik.
- Jumlah untuk `SampleCount`.


Statistik yang valid: `Minimum`, `Maximum`, `Average`, `Sum`, `SampleCount`

**TotalIOBytes**

Jumlah byte aktual untuk setiap operasi sistem file, termasuk pembacaan data, penulisan data, dan operasi metadata. Ini adalah jumlah aktual yang dikendarai aplikasi Anda, dan bukan throughput yang diukur oleh sistem file. Mungkin lebih tinggi dari angka yang ditunjukkan `PermittedThroughput`.

`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. `AverageStatistic` adalah ukuran rata-rata operasi selama periode tersebut. `SampleCountStatistic` 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.

`SumStatistic` adalah jumlah total byte yang terkait dengan operasi baca. `MinimumStatistic` adalah ukuran operasi baca terkecil selama periode tersebut. `MaximumStatistic` adalah ukuran operasi baca terbesar selama periode tersebut. `AverageStatistic` adalah ukuran rata-rata operasi baca selama periode tersebut. `SampleCountStatistic` 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. `MinimumStatistic` adalah ukuran operasi tulis terkecil selama periode tersebut. `MaximumStatistic` adalah ukuran operasi

penulisan terbesar selama periode tersebut. `AverageStatistic` adalah ukuran rata-rata operasi tulis selama periode tersebut. `SampleCountStatistic` 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.

`SumStatistic` adalah jumlah total byte yang terkait dengan operasi metadata. `MinimumStatistic` adalah ukuran operasi metadata terkecil selama periode tersebut. `MaximumStatistic` adalah ukuran operasi metadata terbesar selama periode tersebut. `AverageStatistic` adalah ukuran operasi metadata rata-rata selama periode tersebut. `SampleCountStatistic` 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.

`SumStatistic` adalah jumlah total byte yang terkait dengan operasi pembacaan metadata. `MinimumStatistic` adalah ukuran operasi pembacaan metadata terkecil selama periode tersebut. `MaximumStatistic` adalah ukuran operasi pembacaan metadata terbesar selama periode tersebut. `AverageStatistic` adalah ukuran rata-rata operasi pembacaan metadata selama periode tersebut. `SampleCountStatistic` 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 EC2 instans Amazon 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 EFS penyimpanan. 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 Standar.
- **IA** adalah ukuran sebenarnya (dalam byte) data yang disimpan di kelas penyimpanan Akses EFS Jarang.
- **IASizeOverhead** adalah perbedaan (dalam byte) antara ukuran aktual data di kelas penyimpanan Akses EFS Jarang (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 Arsip.
- **ArchiveSizeOverhead** adalah perbedaan (dalam byte) antara ukuran aktual data di kelas penyimpanan EFS Arsip (ditunjukkan dalam Archive dimensi) dan ukuran terukur dari kelas penyimpanan, setelah membulatkan file kecil ke 128KiB.

Unit: Bit

Statistik yang valid: Minimum, Maximum, Average

 Note

StorageBytes ditampilkan di halaman metrik sistem File EFS konsol Amazon menggunakan basis 1024 unit (kibibytes, mebibytes, gibibytes, dan tebibytes).

## Mengakses CloudWatch metrik untuk Amazon EFS

Anda dapat melihat EFS metrik Amazon CloudWatch dalam beberapa cara:


- Di EFS konsol Amazon
- Di CloudWatch konsol
- Menggunakan CloudWatch CLI
- Menggunakan CloudWatch API

Untuk melihat CloudWatch metrik dan alarm (konsol Amazon EFS)

1. Masuk ke AWS Management Console dan buka EFS konsol Amazon 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 set 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 kinerja Max I/O, 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. Di panel navigasi, pilih Metrik.
3. Pilih EFSnamespace.
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 CloudWatch API

- Panggil [GetMetricStatistics](#). Untuk informasi selengkapnya, lihat [CloudWatch API Referensi Amazon](#).

## 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 melacak jumlah EC2 instans Amazon yang terhubung ke sistem file?	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 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 <a href="#">Throughput yang melonjak</a> .

## Memantau kinerja throughput

CloudWatch Metrik untuk pemantauan throughput—`TotalIOBytes`, `ReadIOBytes`, `WriteIOBytes`, 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 Pemanfaatan Throughput (%) di bagian Pemantauan EFS konsol Amazon 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 Standar 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 Elastic 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 EFS metrik Amazon 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 Matematika Metrik](#) di Panduan CloudWatch Pengguna Amazon.

Berikut ini, temukan beberapa ekspresi matematika metrik yang berguna untuk AmazonEFS.

### Topik

- [Matematika metrik: Throughput di MiBps](#)
- [Matematika metrik: Persen throughput](#)
- [Matematika metrik: Persentase pemanfaatan throughput yang diizinkan](#)



- [Matematika metrik: Throughput IOPS](#)
- [Matematika metrik: Persentase IOPS](#)
- [Matematika metrik: Ukuran I/O rata-rata di KiB](#)
- [Menggunakan matematika metrik melalui AWS 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`, `DataWriteIOBytesMetadataIOBytes`, 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	<ul style="list-style-type: none"> <li>• <code>DataReadIOBytes</code></li> <li>• <code>DataWriteIOBytes</code></li> <li>• <code>MetadataIOBytes</code></li> <li>• <code>TotalIOBytes</code></li> </ul>	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 persentase throughput keseluruhan yang digunakan untuk tipe I/O yang 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: (jumlah `DataReadIOBytes` x 100 (untuk mengkonversi ke persentase)) ÷ jumlah `TotalIOBytes`

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik atau metrik yang dapat digunakan	Statistik	Periode
m1	• <code>TotalIOBytes</code>	sum	1 menit
m2	• <code>DataReadIOBytes</code>	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	$(m1/1048576)/(m1) \text{ PERIOD}$
e2	$m2/1048576$
e3	$((e1)*100)/(e2)$

## Matematika metrik: Throughput IOPS

Untuk menghitung operasi rata-rata per detik (IOPS) untuk periode waktu, 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

Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	Metrik yang dapat digunakan	Statistik	Periode
m1	<ul style="list-style-type: none"> <li><code>DataReadIOBytes</code></li> <li><code>DataWriteIOBytes</code></li> </ul>	jumlah sampel	1 menit

ID	Metrik yang dapat digunakan	Statistik	Periode
	<ul style="list-style-type: none"> <li>• <code>MetadataIOBytes</code></li> <li>• <code>TotalIOBytes</code></li> </ul>		

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	<code>m1/PERIOD(m1)</code>

## Matematika metrik: Persentase IOPS

Untuk menghitung persentase IOPS per detik dari jenis I/O yang berbeda (`DataReadIOBytes`, `DataWriteIOBytes`, atau `MetadataIOBytes`) untuk periode waktu, 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	<ul style="list-style-type: none"> <li>• <code>TotalIOBytes</code></li> </ul>	jumlah sampel	1 menit
m2	<ul style="list-style-type: none"> <li>• <code>DataReadIOBytes</code></li> <li>• <code>DataWriteIOBytes</code></li> <li>• <code>MetadataIOBytes</code></li> </ul>	jumlah sampel	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	$(m2 * 100) / m1$

### Matematika metrik: Ukuran I/O rata-rata di KiB

Untuk menghitung ukuran I/O rata-rata (dalam KiB) untuk suatu periode, bagilah statistik penjumlahan 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	<ul style="list-style-type: none"> <li>• <code>DataReadIOBytes</code></li> <li>• <code>DataWriteIOBytes</code></li> <li>• <code>MetadataIOBytes</code></li> </ul>	sum	1 menit
m2	<ul style="list-style-type: none"> <li>• <code>DataReadIOBytes</code></li> <li>• <code>DataWriteIOBytes</code></li> <li>• <code>MetadataIOBytes</code></li> </ul>	jumlah sampel	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	(m1/1024)/m2

## Menggunakan matematika metrik melalui AWS CloudFormation template untuk Amazon EFS

Anda juga dapat membuat ekspresi matematika metrik melalui AWS CloudFormation template. Salah satu template tersebut tersedia bagi Anda untuk men-download dan menyesuaikan untuk digunakan dari [Amazon EFS tutorial](#) di GitHub. Untuk informasi selengkapnya tentang penggunaan AWS CloudFormation templat, lihat [Bekerja dengan AWS CloudFormation Template](#) 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 EFS file Anda dari jarak jauh tanpa harus masuk ke klien. Gunakan prosedur berikut untuk mengonfigurasi EC2 instans 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 EC2 instance pemasangan sistem file. Untuk informasi selengkapnya, lihat [Secara otomatis menginstal atau memperbarui EFS klien Amazon menggunakan AWS Systems Manager](#) atau [Menginstal EFS klien Amazon secara manual](#).
2. Instal `botocore` pada EC2 instance 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/utills/{fs_id}
```

5. Lampirkan kebijakan AmazonElasticFileSystemsUtils AWS terkelola ke IAM peran yang telah Anda lampirkan ke EC2 instance, 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 IAM instance 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 log `/aws/efs/utills`. Anda akan melihat aliran log untuk setiap EC2 instance Amazon dan kombinasi sistem EFS file.
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 SNS pesan Amazon 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 pemberitahuan yang dikirim ke SNS topik Amazon atau kebijakan Auto Scaling.

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 EFS file Amazon saat dibuat. Untuk menerapkan encryption-at-rest kebijakan data untuk sistem EFS file Amazon, 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

Untuk mengatur alarm menggunakan konsol CloudWatch

1. Masuk ke AWS Management Console dan buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Pilih Buat Alarm. Proses ini meluncurkan Wizard Buat Alarm.



3. Pilih EFSMetrik dan gulir melalui EFS metrik Amazon untuk menemukan metrik yang ingin Anda gunakan alarm. Untuk hanya menampilkan EFS metrik Amazon di kotak dialog ini, cari ID sistem file dari 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 mengirimi Anda email saat status alarm tercapai, di bidang Setiap kali alarm ini:, pilih Status adalah ALARM. Di bidang Kirim pemberitahuan ke:, pilih SNS topik yang ada. Jika memilih 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 SNS topik Amazon 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.

Untuk mengatur alarm menggunakan AWS CLI

- Panggil [put-metric-alarm](#). Untuk informasi selengkapnya, lihat [Referensi Perintah AWS CLI](#).

Untuk mengatur alarm menggunakan CloudWatch API

- Panggil [PutMetricAlarm](#). Untuk informasi selengkapnya, lihat [CloudWatch API Referensi Amazon](#).

## Mencatat EFS API panggilan Amazon dengan AWS CloudTrail

Amazon EFS terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AmazonEFS. CloudTrail menangkap semua API panggilan untuk Amazon EFS sebagai acara, termasuk panggilan dari EFS konsol Amazon dan dari panggilan kode ke EFS API operasi Amazon.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkalanjutan 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 itu dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

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

## EFSSInformasi Amazon 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 peristiwa terbaru di Akun AWS Anda. Untuk informasi selengkapnya, lihat [Bekerja dengan riwayat CloudTrail Acara](#).

Untuk catatan acara yang sedang berlangsung di Akun AWS Anda, 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 Wilayah AWS. Trail mencatat peristiwa dari semua Wilayah 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 SNS notifikasi Amazon untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa Wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua EFS [API panggilan](#) Amazon 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 siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan dibuat dengan pengguna root atau AWS Identity and Access Management (IAM) kredensial pengguna.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [CloudTrail userIdentityelemen](#) dalam Panduan AWS CloudTrail Pengguna.

## Memahami entri file EFS log Amazon

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 API panggilan publik, sehingga 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 EFS layanan

Peran EFS terkait layanan Amazon membuat API panggilan ke AWS sumber daya. Anda akan melihat entri CloudTrail log dengan panggilan username: `AWSServiceRoleForAmazonElasticFileSystem` yang dibuat oleh peran EFS terkait layanan. Untuk informasi selengkapnya tentang EFS dan peran terkait layanan, lihat [Menggunakan peran tertaut layanan untuk Amazon EFS](#)

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateServiceLinkedRole` tindakan saat Amazon EFS membuat peran `AWSServiceRoleForAmazonElasticFileSystem` 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%22
elasticfilesystem.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 EFS otentikasi

EFS otorisasi Amazon untuk memancarkan `NewClientConnection` dan `UpdateClientConnection` CloudTrail acara NFS klien. 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 EFS otorisasi, lihat [Menggunakan IAM untuk mengontrol akses data 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 EFS log Amazon 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 [Mengenkripsi data di Amazon EFS](#).

Amazon EFS mengirimkan [konteks Enkripsi](#) saat membuat AWS KMS API permintaan untuk menghasilkan kunci data dan mendekripsi data AmazonEFS. 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"
```

# EFS Kinerja Amazon

Bagian berikut memberikan ikhtisar EFS kinerja Amazon, 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 EFS kinerja Amazon](#)
- [Memecahkan masalah kinerja Amazon EFS](#)
- [Pemecahan masalah AMI dan masalah kernel](#)

## Ringkasan kinerja

Kinerja sistem file biasanya diukur dengan menggunakan dimensi latensi, throughput, dan operasi Input/Output per detik (IOPS). IOPS EFS Kinerja Amazon di seluruh dimensi ini bergantung pada konfigurasi sistem file Anda. Konfigurasi berikut memengaruhi kinerja sistem EFS file Amazon:

- Jenis sistem file - Regional atau Satu Zona
- Mode kinerja - Tujuan Umum atau Max I/O

### Important

Mode kinerja Max I/O 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		Maksimum IOPS		Throughput maksimum		
Jenis sistem file	Mode throughput	Baca operasi	Tulis operasi	Baca operasi	Tulis operasi	Per-file-s system baca <sup>1</sup>	Per-file-s system tulis <sup>1</sup>	Baca/tulis per klien
Regional	Elastis	Serendah 250 mikrodetik ( $\mu$ s)	Serendah 2,7 milidetik (ms)	90.000-250.000 <sup>2</sup>	50.000	10—60 gibibyte per detik (GiBps)	1—5 GiBps	1.500 mebibytes per detik (MiBps)
Regional	Disediakan	Serendah 250 $\mu$ s	Serendah 2,7 ms	55.000	25.000	3—10 GiBps	1—3,33 GiBps	500 MiBps
Regional	Meledak	Serendah 250 $\mu$ s	Serendah 2,7 ms	35.000	7.000	3—5 GiBps	1—3 GiBps	500 MiBps
Satu Zona	Elastis, Disediakan, Meledak	Serendah 250 $\mu$ s	Serendah 1,6 ms	35.000	7.000	3 GiBps <sup>4</sup>	1 GiBps <sup>4</sup>	500 MiBps

**Note**

Catatan kaki:

1. Throughput baca dan tulis maksimum tergantung pada Wilayah AWS Throughput yang melebihi throughput maksimum Wilayah AWS seseorang membutuhkan peningkatan kuota throughput. Setiap permintaan untuk throughput tambahan dipertimbangkan case-by-case berdasarkan tim EFS layanan Amazon. Persetujuan mungkin tergantung pada jenis beban kerja Anda. Untuk mempelajari lebih lanjut tentang meminta kenaikan kuota, lihat.

[EFSKuota Amazon](#)

2. Sistem file yang menggunakan throughput Elastic dapat mendorong maksimum 90.000 pembacaan IOPS untuk data yang jarang diakses dan 250.000 bacaan untuk data yang sering diakses. Rekomendasi tambahan berlaku IOPS untuk mencapai maksimum. IOPS Untuk informasi selengkapnya, lihat [the section called “Mengoptimalkan beban kerja yang menuntut throughput tinggi dan IOPS”](#).
3. 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 EFS klien Amazon (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 EFS klien Amazon, lihat [Menginstal EFS klien Amazon](#)
4. 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 EFS penyimpanan Amazon dirancang untuk penyimpanan paling efektif tergantung pada kasus penggunaan.

- EFS Kelas penyimpanan standar menggunakan penyimpanan solid state drive (SSD) untuk memberikan tingkat latensi terendah untuk file yang sering diakses. Kelas penyimpanan ini menyediakan latensi byte pertama serendah 250 mikrodetik untuk pembacaan dan 2,7 milidetik untuk penulisan.
- EFS Kelas penyimpanan Infrequent Access (IA) dan EFS Archive menyimpan data yang jarang diakses yang tidak memerlukan kinerja latensi yang diperlukan data yang sering diakses. Kelas penyimpanan ini menyediakan latensi byte pertama puluhan milidetik.

Untuk informasi selengkapnya tentang kelas EFS penyimpanan, lihat [the section called “EFS kelas penyimpanan”](#).

## 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.
- Mode 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. Mode Max I/O 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 IOPS batas yang tersedia untuk sistem file menggunakan mode kinerja Tujuan Umum, Anda dapat memantau `PercentIOLimit` CloudWatch metrik. Untuk informasi selengkapnya, lihat [CloudWatch metrik untuk Amazon EFS](#).

Aplikasi dapat menskalakannya IOPS secara elastis hingga batas yang terkait dengan mode kinerja. Anda tidak ditagih secara terpisah IOPS; mereka termasuk dalam akuntansi throughput sistem file. Setiap permintaan Network File System (NFS) dihitung sebagai 4 kilobyte (KB) throughput, atau permintaan dan ukuran responsnya yang sebenarnya, 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 wilayah AWS. Untuk informasi selengkapnya tentang throughput sistem file maksimum di berbagai wilayah, lihat [EFS Kuota Amazon](#).

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 yang 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](#).

Anda dapat menggunakan Amazon CloudWatch untuk menentukan average-to-peak rasio beban kerja Anda dengan membandingkan MeteredIOBytes metrik dengan PermittedThroughput metrik. Untuk informasi selengkapnya tentang EFS metrik Amazon, 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**

Throughput elastis hanya tersedia untuk sistem file yang menggunakan mode kinerja Tujuan Umum.

Untuk informasi tentang batas throughput Elastis Per-wilayah, lihat [EFSKuota Amazon 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 Wilayah AWS

Jika throughput baseline 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). Wilayah AWS

Untuk informasi tentang batas per Region Provisioned throughput, lihat [EFSKuota Amazon 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 penyimpanan, hingga Wilayah AWS batasnya, dengan minimal 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 EFS burst Amazon

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 Standar. EFS Tingkat dasar adalah 50 MiBps per tebibyte [TiB] penyimpanan (setara dengan 50 KiBps per GiB penyimpanan). Amazon EFS meteran membaca operasi 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 penyimpanan Standar	<ul style="list-style-type: none"> <li>Burst menjadi 300 (MiBps) read-only hingga 72 menit per hari, atau</li> <li>Burst hingga 100 MiBps write-only hingga 72 menit per hari</li> </ul>	<ul style="list-style-type: none"> <li>Berkendara hingga 15 MiBps read-only terus menerus</li> <li>Berkendara hingga 5 MiBps tulis saja terus menerus</li> </ul>
1 TiB data terukur dalam	<ul style="list-style-type: none"> <li>Burst menjadi 300 MiBps read-only selama 12 jam per hari, atau</li> </ul>	<ul style="list-style-type: none"> <li>Dorong 150 MiBps read-only terus menerus</li> </ul>

Ukuran sistem file	Throughput meledak	Throughput dasar
penyimpanan Standar	<ul style="list-style-type: none"> <li>Burst to 100 MiBps write-only selama 12 jam per hari</li> </ul>	<ul style="list-style-type: none"> <li>Dorong 50 hanya MiBps tulis terus menerus</li> </ul>
10 TiB data terukur dalam penyimpanan Standar	<ul style="list-style-type: none"> <li>Burst menjadi 3 GiBps read-only selama 12 jam per hari, atau</li> <li>Burst to 1 GiBps write-only selama 12 jam per hari</li> </ul>	<ul style="list-style-type: none"> <li>Drive 1.5 GiBps read-only terus menerus</li> <li>Drive 500 hanya MiBps tulis terus menerus</li> </ul>
Umumnya, sistem file yang lebih besar	<ul style="list-style-type: none"> <li>Burst menjadi 300 MiBps read-only per TiB penyimpanan selama 12 jam per hari, atau</li> <li>Burst hingga 100 MiBps write-only per TiB penyimpanan selama 12 jam per hari</li> </ul>	<ul style="list-style-type: none"> <li>Dorong 150 MiBps read-only per TiB penyimpanan terus menerus</li> <li>Dorong 50 MiBps write-only per TiB penyimpanan terus menerus</li> </ul>

### 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. [DescribeFileSystems](#) API. 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 Meledak.
- Mengurangi jumlah throughput yang disediakan.

## Kiat EFS kinerja Amazon

Saat menggunakan AmazonEFS, ingatlah kiat kinerja berikut.

### Ukuran I/O rata-rata

Sifat Amazon yang terdistribusi 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 ukuran I/O 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 maksimum 250.000 pembacaan IOPS 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 NFS klien atau lebih, dengan setidaknya 64 utas per klien dalam satu target pemasangan.

## Koneksi simultan

Anda dapat memasang sistem EFS file Amazon 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 EC2 instance Amazon sebelum ditulis ke Amazon secara asinkron. EFS 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 AmazonEFS.

### Note

Model permintaan yang Anda pilih memiliki konsistensi pengorbanan (jika Anda menggunakan beberapa EC2 instans Amazon) 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.

## NFSPengaturan pemasangan klien

Verifikasi bahwa Anda menggunakan opsi pemasangan yang disarankan seperti yang diuraikan di dalam [Memasang sistem EFS file](#) dan di [Pertimbangan pemasangan untuk Linux](#).

Saat memasang sistem file Anda di EC2 instans Amazon, Amazon EFS mendukung protokol Sistem File Jaringan versi 4.0 dan 4.1 (NFSv4). NFSv4.1 memberikan kinerja yang lebih baik untuk operasi pembacaan file kecil paralel (lebih dari 10.000 file per detik) dibandingkan dengan NFSv4 0,0 (kurang dari 1.000 file per detik). Untuk instans Amazon EC2 macOS 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.
- `fs_c` — Opsi ini memungkinkan caching file lokal, tetapi tidak mengubah koherensi NFS cache, dan tidak mengurangi latensi.

#### Note

Saat Anda memasang sistem file Anda, pertimbangkan untuk meningkatkan ukuran buffer baca dan tulis untuk NFS klien 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. Bundling ini dalam `.zip` file memungkinkan Anda membaca banyak file dengan satu perjalanan pulang-pergi buka-tutup.

`.zip`Format 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, NFS klien Linux 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 NFS `read_ahead_kb`

NFS `read_ahead_kb` atribut 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.\*, NFS klien Linux menggunakan `read_ahead_kb` nilai default 128 KB. Kami merekomendasikan untuk meningkatkan nilai ini menjadi 15 MB.

Pembantu EFS pemasangan Amazon 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 mengalami masalah dalam penyelesaian, 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
- RHEL7.3 atau yang lebih baru
- Semua versi Ubuntu 16.04
- Ubuntu 14.04 dengan kernel 3.13.0-83 atau yang lebih baru
- SLES12 Sp2 atau yang lebih baru

Jika Anda menggunakan distribusi lain atau kernel khusus, kami merekomendasikan kernel versi 4.3 atau yang lebih baru.

### Note

RHEL6.9 mungkin kurang optimal untuk beban kerja tertentu karena. [Kinerja buruk saat membuka banyak file secara paralel](#)

### Topik

- [Tidak dapat membuat sistem EFS berkas](#)

- [Akses ditolak ke file yang diizinkan pada sistem NFS file](#)
- [Kesalahan saat mengakses konsol Amazon EFS](#)
- [EC2Instans Amazon hang](#)
- [Aplikasi menulis sejumlah besar data hang](#)
- [Kinerja buruk saat membuka banyak file secara paralel](#)
- [NFS Pengaturan khusus yang menyebabkan penundaan penulisan](#)
- [Membuat backup dengan Oracle Recovery Manager lambat](#)

## Tidak dapat membuat sistem EFS berkas

Permintaan untuk membuat sistem EFS file 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 untuk membuat sistem EFS file 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 NFS file

Ketika pengguna yang ditugaskan lebih dari 16 grup akses IDs (GIDs) mencoba untuk melakukan operasi pada sistem NFS file, mereka dapat ditolak akses ke file yang diizinkan pada sistem file. [Masalah ini terjadi karena NFS protokol mendukung maksimum 16 GIDs per pengguna, dan tambahan apa pun GIDs dipotong dari permintaan NFS klien, seperti yang didefinisikan dalam RFC 5531.](#)

Tindakan yang harus diambil

Merestrukturisasi pemetaan NFS pengguna dan grup 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 EFS manajemen Amazon.



## Kesalahan mengautentikasi kredensial untuk **ec2:DescribeVPCs**

Pesan galat berikut ditampilkan saat mengakses EFS konsol Amazon:

```
AuthFailure: An error occurred authenticating your credentials for ec2:DescribeVPCs.
```

Kesalahan ini menunjukkan bahwa kredensi login Anda tidak berhasil mengautentikasi dengan layanan Amazon. EC2 EFSKonsol Amazon memanggil EC2 layanan Amazon atas nama Anda saat membuat sistem EFS file VPC yang Anda pilih.

Tindakan yang harus diambil

Pastikan waktu klien mengakses EFS konsol Amazon diatur dengan benar.

## EC2Instans Amazon hang

EC2Instans Amazon 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 EFS file Amazon 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 AmazonEFS, 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 paralelisasi I/O yang diharapkan.

Tindakan yang harus diambil

Masalah ini terjadi pada klien Network File System versi 4 (NFSv4) dan pada RHEL 6 klien yang menggunakan NFSv4 .1 karena NFS klien ini membuat serial NFS OPEN dan CLOSE operasi. Gunakan NFS protokol 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 NFS server 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 EC2 instance Amazon 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 EC2 instans Amazon yang terpisah.

## NFSPengaturan khusus yang menyebabkan penundaan penulisan

Anda memiliki pengaturan NFS klien khusus, dan dibutuhkan hingga tiga detik untuk EC2 instance Amazon untuk melihat operasi tulis yang dilakukan pada sistem file dari EC2 instans Amazon lainnya.

Tindakan yang harus diambil

Jika Anda mengalami masalah ini, Anda dapat menyelesaikannya dengan salah satu cara berikut:

- Jika NFS klien di EC2 instans Amazon 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 NFS prosedur. 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 DirectNFS, seperti yang dijelaskan dalam [Mengaktifkan dan Menonaktifkan Kontrol NFS Klien Langsung NFS di Pusat Bantuan](#) Oracle.

#### Note

Amazon EFS tidak mendukung Oracle DirectNFS.

## Pemecahan masalah AMI dan masalah 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 AmazonEC2.

## 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/direktori menggunakan perintah Linuxchown.

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 EFS root, 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-generik

Tindakan yang harus diambil

Lakukan salah satu hal berikut ini:

- Upgrade ke versi kernel yang lebih baru. Untuk CentOS-7, versi kernel Linux 3.10.0-327 atau yang lebih baru berisi perbaikan.
- Downgrade ke versi kernel yang lebih lama.

## Daftar file dalam direktori besar membutuhkan waktu lama

Hal ini dapat terjadi jika direktori berubah sementara NFS klien Anda iterasi melalui direktori untuk menyelesaikan operasi daftar. Setiap kali NFS klien memperhatikan bahwa isi direktori berubah selama iterasi ini, NFS klien 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

CentOS dan versi RHEL kernel lebih rendah dari 2.6.32-696.el6

Tindakan yang harus diambil

Untuk mengatasi masalah ini, tingkatkan ke versi kernel yang lebih baru.

# Melindungi data Anda di Amazon EFS

Untuk melindungi data Anda, Amazon EFS secara otomatis mencadangkan sistem EFS file Anda. Untuk ketahanan dan perlindungan data lebih lanjut, Anda dapat mereplikasi sistem EFS file Anda dalam file. Wilayah AWS Mencadangkan dan mereplikasi sistem EFS file Anda memastikan bahwa Anda dapat menyediakan operasi atau layanan lanjutan jika terjadi sesuatu pada data sistem EFS file. Misalnya, dalam kasus korupsi data atau kehilangan data.

## Topik

- [Mencadangkan sistem EFS file](#)
- [Mereplikasi sistem EFS file](#)

## Mencadangkan sistem EFS file

Amazon terintegrasi EFS 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

Menggunakan AWS Backup untuk AmazonEFS, 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.
- Kembalikan cadangan data AmazonEFS. 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 <https://docs.aws.amazon.com/aws-backup/latest/devguide/getting-started.html> di Panduan AWS Backup Pengembang.

## Topik

- [Bagaimana AWS Backup bekerja dengan Amazon EFS](#)
- [Diperlukan izin IAM](#)
- [Kinerja Backup](#)

- [Mengelola backup otomatis sistem file EFS](#)

## Bagaimana AWS Backup bekerja dengan Amazon EFS

Sistem file yang Anda buat menggunakan EFS konsol Amazon secara otomatis dicadangkan secara AWS Backup default. Anda dapat mengaktifkan backup otomatis setelah membuat sistem EFS file Anda menggunakan atau. AWS CLI API Paket EFS pencadangan default menggunakan pengaturan yang AWS Backup disarankan untuk pencadangan otomatis—pencadangan harian dengan periode retensi 35 hari. Pencadangan yang dibuat menggunakan paket EFS cadangan default disimpan di brankas EFS cadangan default, yang juga dibuat oleh Amazon EFS atas nama Anda. Paket cadangan default dan brankas cadangan tidak dapat dihapus.

Semua data dalam sistem EFS file dicadangkan, apa pun kelas penyimpanan data tersebut. Anda tidak dikenakan biaya akses data saat mencadangkan sistem EFS file yang telah 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 backup inkremental dari EFS sistem file. 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 EFS file 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.

## 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 file - 2 file = 98 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 Akses vault EFS cadangan default disetel untuk menolak penghapusan titik pemulihan. Untuk menghapus cadangan sistem EFS file yang ada, Anda harus mengubah kebijakan akses vault. Jika Anda mencoba menghapus titik EFS pemulihan 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 IAM tindakan \(akses admin\)](#) di Panduan IAM Pengguna.

## Diperlukan izin IAM

AWS Backup membuat peran terkait layanan atas nama Anda di akun Anda. Peran ini memiliki izin yang diperlukan untuk melakukan EFS backup Amazon.

Anda dapat menggunakan `elasticfilesystem:backup` dan `elasticfilesystem:restore` tindakan untuk mengizinkan atau menolak IAM entitas (seperti pengguna, grup, atau peran) kemampuan untuk membuat atau memulihkan cadangan sistem EFS file. Anda dapat menggunakan tindakan ini dalam kebijakan sistem file atau dalam kebijakan berbasis identitas IAM. Untuk informasi selengkapnya, silakan lihat [Manajemen identitas dan akses untuk Amazon EFS](#) dan [Menggunakan IAM untuk mengontrol akses data 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 1.000 file per detik atau 300 megabyte per detik (MBps), mana yang lebih lambat.
- Tingkat pemulihan 500 file per detik atau 150MBps, 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 EFS file Amazon](#).

## Mengelola backup otomatis sistem file EFS

Saat Anda membuat sistem file menggunakan EFS konsol Amazon, pencadangan otomatis diaktifkan secara default. Anda dapat mengaktifkan backup otomatis setelah membuat sistem file Anda menggunakan atau. AWS CLI API

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 EFS cadangan default menggunakan konsol.AWS Backup](#)

Amazon EFS menerapkan kunci tag `aws:elasticfilesystem:default-backup` sistem dengan nilai `enabled` ke sistem EFS file saat pencadangan otomatis diaktifkan.

Setelah Anda membuat sistem file, Anda dapat mengaktifkan atau menonaktifkan pencadangan otomatis menggunakan konsol, file AWS CLI, atau file. EFS API

Mengaktifkan atau menonaktifkan pencadangan otomatis untuk sistem file (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 backup otomatis, pilih Aktifkan backup otomatis.
  - Untuk mematikan backup otomatis, hapus Aktifkan backup otomatis.
5. Pilih Simpan perubahan.

Mengaktifkan atau menonaktifkan pencadangan otomatis untuk sistem file ()AWS CLI

- Gunakan `put-backup-policy` CLI perintah (API operasi yang sesuai adalah [PutBackupPolicy](#)) aktifkan atau matikan cadangan 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"
```

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"
```

EFS merespons dengan kebijakan cadangan baru.

```
{  
  "BackupPolicy": {  
    "Status": "DISABLING"  
  }  
}
```

## Mereplikasi sistem EFS file

Untuk ketahanan dan perlindungan data yang diperluas, Anda dapat mereplikasi sistem EFS file Anda dalam file. Wilayah AWS Saat Anda mengaktifkan replikasi pada sistem EFS file, 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 menjaga 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 EFS file di Wilayah yang dinonaktifkan secara default, Anda harus terlebih dahulu memilih masuk ke Wilayah. 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 Wilayah AWS lagi.

#### Note

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

## Topik

- [Biaya](#)
- [Performa replikasi](#)
- [Diperlukan izin IAM](#)
- [Mengkonfigurasi replikasi ke sistem file baru EFS](#)
- [Mengkonfigurasi replikasi ke sistem file yang ada EFS](#)
- [Melihat detail replikasi](#)
- [Menghapus konfigurasi replikasi](#)
- [Menggunakan replika](#)

## Biaya

Untuk memfasilitasi replikasi, Amazon EFS membuat direktori tersembunyi dan metadata 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. 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) 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), theAPI, dan Amazon CloudWatch. Dalam CloudWatch, gunakan [TimeSinceLastSync](#)EFSmetrik. Untuk informasi selengkapnya, lihat [Melihat detail replikasi](#).

## Diperlukan izin IAM

Amazon EFS menggunakan peran EFS terkait layanan bernama `AWSServiceRoleForAmazonElasticFileSystem` untuk menyinkronkan status replikasi antara sistem file sumber dan tujuan. Untuk menggunakan replikasi, Anda harus mengonfigurasi izin berikut untuk mengizinkan AWS Identity and Access Management (IAM) entitas (seperti pengguna, grup, atau peran) untuk membuat peran terkait layanan, konfigurasi replikasi, dan sistem file.

- `iam:CreateServiceLinkedRole`— lihat contoh di [Menggunakan peran tertaut layanan untuk Amazon EFS](#).
- `elasticfilesystem:DescribeFileSystem`
- `elasticfilesystem:CreateFileSystem`\*
- `elasticfilesystem:CreateReplicationConfiguration`\*
- `elasticfilesystem>DeleteReplicationConfiguration`\*
- `elasticfilesystem:DescribeReplicationConfigurations`\*

\* Anda dapat menggunakan kebijakan `AmazonElasticFileSystemFullAccess` terkelola untuk secara otomatis mendapatkan semua EFS izin yang diperlukan. Untuk informasi selengkapnya, lihat [AWS kebijakan terkelola: AmazonElasticFileSystemFullAccess](#).

## Mengkonfigurasi replikasi ke sistem file baru EFS

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. Wilayah AWS 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 EFS file Amazon menyimpan data dalam file file Wilayah AWS.
  - Pilih Regional untuk membuat sistem file yang menyimpan data dan metadata secara berlebihan di semua Availability Zone di dalam. Wilayah AWS
  - 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 [EFS jenis sistem file](#).

- 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 KMS kunci, kunci yang dikelola layanan KMS untuk Amazon akan EFS digunakan.

 Important

Setelah sistem file tujuan dibuat, Anda tidak dapat mengubah KMS kuncinya.

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 [EFSKuota Amazon 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 untuk sistem file EFS](#).

## Langkah 1: Buat konfigurasi replikasi


Langkah pertama dalam mereplikasi ke sistem file baru adalah membuat konfigurasi replikasi.

Buat konfigurasi replikasi (konsol)

1. Masuk ke AWS Management Console dan buka EFS konsol Amazon 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 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. 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 Wilayah AWS, pilih Wilayah AWS 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 Wilayah AWS, pilih Regional.
- Untuk membuat sistem file yang menyimpan data secara berlebihan dalam satu Availability Zone dalam satu Wilayah AWS, pilih One Zone, lalu pilih Availability Zone.

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

 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 kunci layanan AWS Key Management Service (AWS KMS) Anda (`aws/elasticfilesystem`). Untuk menggunakan KMS kunci yang berbeda, pilih KMS kunci atau masukkan kunci Amazon Resource Name (ARN).

 Important

Setelah sistem file dibuat, Anda tidak dapat mengubah KMS kuncinya.

Buat konfigurasi replikasi (AWS CLI)

Bagian ini memberikan contoh untuk membuat konfigurasi replikasi dalam AWS CLI menggunakan `create-replication-configuration` perintah. API Perintah 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` Wilayah AWS. `KmsKeyId` parameter menentukan ID KMS kunci yang akan 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 file *fs-0123456789abcdef1*. Contoh menggunakan `AvailabilityZoneName` parameter untuk membuat sistem file tujuan One Zone di *us-west-2a* Availability Zone. Karena tidak ada KMS kunci 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 EFS file](#).

## Mengkonfigurasi replikasi ke sistem file yang ada EFS

Amazon EFS mereplikasi data dan metadata pada sistem file sumber ke sistem file tujuan dan Wilayah AWS yang Anda pilih. Selama replikasi, Amazon EFS mengidentifikasi perbedaan data antara sistem file dan menerapkan perbedaan ke sistem file tujuan.

Untuk mereplikasi ke yang sudah ada, lakukan langkah-langkah berikut.

### Topik

- [Langkah 1: Nonaktifkan perlindungan penimpanan 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 EFS file Amazon, 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 EFS file Amazon dapat memiliki salah satu nilai yang dijelaskan dalam tabel berikut.

Status sistem file	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.

Status sistem file	Deskripsi
REPLICATING	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 penyimpanan replikasi memerlukan izin untuk tindakan tersebut.

`elasticfilesystem:UpdateFileSystemProtection` Untuk informasi selengkapnya, lihat [AWS kebijakan terkelola: AmazonElasticFileSystemFullAccess](#).

Untuk menonaktifkan perlindungan penyimpanan replikasi (konsol)

1. Masuk ke AWS Management Console dan buka EFS konsol Amazon di <https://console.aws.amazon.com/efs/>.
2. Di panel navigasi kiri, pilih Sistem file.
3. Dalam daftar Sistem file, pilih sistem EFS file Amazon yang ingin Anda gunakan sebagai sistem file tujuan dalam konfigurasi replikasi.
4. Di bagian Perlindungan sistem file, matikan Perlindungan Penyimpanan Replikasi.

### Langkah 2: Buat konfigurasi replikasi

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 [Mengekripsi data di Amazon EFS](#).

Untuk membuat konfigurasi replikasi (konsol)

1. Masuk ke AWS Management Console dan buka EFS konsol Amazon 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 EFS file Amazon 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 Browse EFS, lalu pilih sistem file. Jalur ke sistem file tujuan Anda muncul di kotak Tujuan.
7. 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.
8. Pilih Buat replikasi, ketik konfirmasi di kotak masukan pesan konfirmasi, lalu pilih Buat replikasi.
9. Bagian Replikasi menunjukkan detail replikasi.
10. Untuk melihat konfigurasi sistem file tujuan, pilih ID sistem file di atas Sistem file Tujuan.

Untuk membuat konfigurasi replikasi ( )AWS CLI

Bagian ini memberikan contoh untuk membuat konfigurasi replikasi dalam AWS CLI menggunakan `create-replication-configuration` perintah. API Perintah 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` Wilayah AWS. KmsKeyIdParameter menentukan ID KMS kunci yang akan 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 file *fs-0123456789abcdef1*. Contoh menggunakan `AvailabilityZoneName` parameter untuk membuat sistem file tujuan One Zone di *us-west-2a* Availability Zone. Karena tidak ada KMS kunci 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

```

## 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 EFS konsol Amazon, AWS CLI API, atau Amazon CloudWatch.

- Di EFS konsol — Properti terakhir yang disinkronkan di bagian Rincian sistem file > Replikasi menunjukkan waktu ketika sinkronisasi terakhir yang berhasil antara sumber dan tujuan selesai.
- Di AWS CLI or API — `LastReplicatedTimestamp` Properti dalam `Destination` objek menunjukkan waktu bahwa sinkronisasi sukses terakhir selesai. Untuk mengakses

properti ini, gunakan `describe-replication-configurations` CLI perintah.

[DescribeReplicationConfigurations](#) adalah API operasi yang setara.

- Di CloudWatch — `TimeSinceLastSync` CloudWatch Metrik untuk Amazon EFS menunjukkan waktu yang telah berlalu sejak sinkronisasi sukses terakhir selesai. 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 diprakarsai pengguna.
PAUSING	Amazon EFS sedang dalam proses menjeda replikasi, sebagai hasil dari memilih keluar dari Wilayah untuk satu atau kedua sistem file dalam konfigurasi replikasi.
PAUSED	Replikasi dijeda sebagai akibat dari memilih keluar dari Wilayah untuk satu atau kedua sistem file dalam konfigurasi replikasi. Untuk melanjutkan replikasi, Anda perlu kembali memilih untuk. Wilayah AWS Untuk informasi selengkapnya, lihat <a href="#">Menentukan Wilayah AWS akun mana yang dapat digunakan</a> dalam Panduan Referensi AWS Umum.
ERROR	Satu (atau keduanya) sistem file dalam konfigurasi replikasi berada dalam keadaan gagal dan tidak dapat dipulihkan. Untuk mengakses data sistem file, kembalikan cadangan sistem file yang gagal ke sistem file baru. Untuk informasi selengkapnya, lihat <a href="#">Melindungi data Anda di Amazon EFS</a> .

Untuk melihat konfigurasi replikasi (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.

Status Dijeda terjadi sebagai akibat dari memilih keluar dari wilayah sumber atau tujuan setelah konfigurasi replikasi dibuat. Untuk melanjutkan replikasi untuk sistem file, Anda perlu kembali memilih untuk. Wilayah AWS Untuk informasi selengkapnya, lihat [Menentukan Wilayah AWS akun mana yang dapat digunakan](#) dalam Panduan Referensi AWS Umum.

Status replikasi terjadi setelah replikasi dibuat, dengan sistem file sebagai sistem file sumber atau tujuan.

Status Kesalahan terjadi ketika sumber atau sistem file tujuan (atau keduanya) dalam keadaan gagal dan tidak dapat dipulihkan. Untuk informasi selengkapnya, lihat [Melihat detail replikasi](#). Untuk memulihkan, Anda harus menghapus konfigurasi replikasi, dan kemudian mengembalikan cadangan terbaru dari sistem file yang gagal (baik sumber atau tujuan) ke sistem file baru.

- 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), Wilayah AWS di mana ia berada, dan Izinnya. Sistem file sumber memiliki izin Writable, dan sistem file tujuan memiliki izin Read-only.

Untuk melihat konfigurasi replikasi (AWS CLI)

Untuk melihat konfigurasi replikasi, gunakan perintah `describe-replication-configurations`. Anda dapat melihat konfigurasi replikasi baik untuk sistem file tertentu, atau semua konfigurasi replikasi untuk tertentu Akun AWS dalam file. Wilayah AWS API Perintah yang setara adalah [DescribeReplicationConfigurations](#).

Untuk melihat konfigurasi replikasi untuk sistem file, gunakan parameter `file-system-id` URI permintaan. Anda dapat menentukan ID dari sistem file sumber atau tujuan.

```
aws efs describe-replication-configurations --file-system-id fs-0123456789abcdef1
```

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

Untuk melihat semua konfigurasi replikasi akun di akun Wilayah AWS, jangan tentukan parameternya. `file-system-id`

```
aws efs describe-replication-configurations
```

```
{
  "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 EFS konsol Amazon, konsol AWS CLI, atau file. API

Untuk menghapus konfigurasi replikasi (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.

Untuk menghapus konfigurasi replikasi (AWS CLI)

Untuk menghapus konfigurasi replikasi, gunakan `delete-replication-configuration` CLI perintah. API Perintah yang setara adalah [DeleteReplicationConfiguration](#).

Untuk menentukan konfigurasi replikasi yang Anda hapus, gunakan parameter `source-file-system-id`

```
aws efs --region us-west-2 delete-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1
```

## 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. Untuk informasi selengkapnya, lihat [Performa replikasi](#).

# Mengamankan data Anda di Amazon EFS

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di Amazon Elastic File System. 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. Untuk informasi selengkapnya tentang privasi data, lihat [Privasi Data FAQ](#). Untuk informasi tentang perlindungan data di Eropa, lihat [Model Tanggung Jawab AWS Bersama dan](#) posting GDPR blog di Blog AWS Keamanan.

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 otentikasi multi-faktor (MFA) dengan setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan 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 lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan FIPS 140-3 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atau, gunakan titik akhir. API FIPS Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat [Federal Information Processing Standard \(FIPS\) 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 ketika Anda bekerja dengan 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. Jika Anda memberikan URL ke server eksternal, kami sangat menyarankan agar Anda tidak menyertakan informasi kredensial dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

## Topik

- [Mengenkripsi data di Amazon EFS](#)
- [Manajemen identitas dan akses untuk Amazon EFS](#)
- [Menggunakan IAM untuk mengontrol akses data sistem file](#)
- [Mengontrol akses jaringan ke sistem EFS file Amazon untuk NFS klien](#)
- [Network File System \(NFS\) tingkat pengguna, grup, dan izin](#)
- [Bekerja dengan titik EFS akses Amazon](#)
- [Memblokir akses publik ke sistem EFS file](#)
- [Validasi kepatuhan untuk Amazon EFS](#)
- [Ketahanan di Amazon EFS](#)
- [Isolasi jaringan untuk Amazon EFS](#)

## Mengenkripsi data di Amazon EFS

Amazon EFS mendukung dua bentuk enkripsi untuk sistem file, enkripsi data dalam perjalanan dan enkripsi saat istirahat. Anda dapat mengaktifkan enkripsi data saat istirahat saat membuat sistem EFS file Amazon. Anda dapat mengaktifkan enkripsi data dalam perjalanan saat Anda memasang sistem file.

Jika Anda memerlukan FIPS 140-2 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atau, gunakan titik akhir. API FIPS Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat [Federal Information Processing Standard \(FIPS\) 140-2](#).

Jika organisasi Anda tunduk pada kebijakan perusahaan atau peraturan yang memerlukan enkripsi data dan metadata saat istirahat, sebaiknya buat sistem file yang dienkripsi saat istirahat, dan memasang sistem file Anda menggunakan enkripsi data dalam perjalanan.

## Topik

- [AWS KMS](#)
- [Mengenkripsi data saat istirahat](#)

- [Mengenkripsi data dalam perjalanan](#)
- [Enkripsi pemecahan masalah](#)

## AWS KMS

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 menggunakan for Kunci yang dikelola AWS Amazon EFSEFS, `aws/elasticfilesystem`, 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 hibah 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 yang dikelola 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 terkelola pelanggan — Ini adalah KMS kunci yang paling fleksibel untuk digunakan, karena Anda dapat mengonfigurasi kebijakan dan hibah utamanya 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 [Mengelola akses ke sistem file terenkripsi](#).

**⚠ Important**

Amazon hanya EFS menerima kunci yang dikelola pelanggan simetris. Anda tidak dapat menggunakan kunci yang dikelola pelanggan asimetris dengan AmazonEFS.

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 EFS log Amazon untuk sistem encrypted-at-rest file](#).

## Kebijakan EFS utama Amazon 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 lebih lanjut tentang hibah, lihat [Menggunakan Pemberian](#) di Panduan Developer AWS Key Management Service . 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 Pilih 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 EFS KMS kebijakan Amazon

KMSKebijakan JSON Kunci yang dikelola AWS untuk AmazonEFS, `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": "*"
}
]
```

## Mengenkripsi data saat istirahat

Anda dapat membuat sistem file terenkripsi menggunakan AWS Management Console,, atau secara AWS CLI terprogram melalui Amazon EFS API atau salah satu file file. AWS SDKs Organisasi Anda mungkin memerlukan enkripsi semua data yang sesuai dengan klasifikasi tertentu atau yang diasosiasikan dengan aplikasi, beban kerja, atau lingkungan tertentu.

Setelah Anda membuat sistem EFS file, Anda tidak dapat mengubah pengaturan enkripsi. Ini berarti Anda tidak dapat memodifikasi sistem file yang tidak terenkripsi untuk membuatnya terenkripsi. Sebagai gantinya, Anda perlu membuat sistem file baru yang dienkripsi.

### Note

Infrastruktur manajemen AWS kunci menggunakan Standar Pemrosesan Informasi Federal (FIPS) 140-2 algoritma kriptografi yang disetujui. Infrastrukturnya konsisten dengan rekomendasi National Institute of Standards and Technology (NIST) 800-57.

## Menegakkan pembuatan sistem EFS file yang dienkripsi saat istirahat

Anda dapat menggunakan kunci `elasticfilesystem:Encrypted IAM` kondisi dalam kebijakan berbasis identitas AWS Identity and Access Management (IAM) untuk mengontrol apakah pengguna dapat membuat sistem EFS file Amazon yang dienkripsi saat istirahat. 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 EFS enkripsi untuk semua Akun AWS s di organisasi Anda. Untuk informasi selengkapnya tentang kebijakan kontrol [layanan AWS Organizations](#), lihat [Kebijakan kontrol layanan](#) di Panduan AWS Organizations Pengguna.

## Mengenkripsi sistem file saat istirahat menggunakan konsol

Saat Anda membuat sistem file baru menggunakan EFS konsol Amazon, enkripsi saat istirahat diaktifkan secara default.

### Note

Enkripsi saat istirahat tidak diaktifkan secara default saat membuat sistem file baru menggunakan AWS CLI, API, dan SDKs. Untuk informasi selengkapnya, lihat [Buat sistem file \(AWS CLI\)](#).

## Cara kerja enkripsi saat istirahat

Dalam sistem file yang dienkripsi, data dan metadata dienkripsi secara otomatis sebelum ditulis ke sistem file. Demikian pula, ketika data dan metadata terbaca, mereka secara otomatis didekripsi sebelum ditampilkan ke aplikasi. Proses ini ditangani secara transparan oleh Amazon EFS, jadi Anda tidak perlu memodifikasi aplikasi Anda.

Amazon EFS menggunakan algoritme enkripsi AES -256 standar industri untuk mengenkripsi EFS data dan metadata saat istirahat. Untuk informasi selengkapnya, lihat [Dasar-dasar kriptografi](#) di Panduan AWS Key Management Service Pengembang.

## Mengenkripsi data dalam perjalanan

Mengaktifkan enkripsi data dalam perjalanan untuk sistem EFS file Amazon Anda dilakukan dengan mengaktifkan Transport Layer Security (TLS) saat Anda memasang sistem file menggunakan Amazon EFS mount helper. Untuk informasi selengkapnya, lihat [Memasang sistem EFS file menggunakan EFS mount helper](#).

Saat enkripsi data dalam perjalanan dideklarasikan sebagai opsi pemasangan untuk sistem EFS file Amazon Anda, mount helper menginisialisasi proses stunnel klien. Stunnel adalah relay jaringan multiguna open source. Proses stunnel klien mendengarkan pada port lokal untuk lalu lintas masuk, dan mount helper mengarahkan lalu lintas klien Network File System (NFS) ke port lokal ini. Mount helper menggunakan TLS versi 1.2 untuk berkomunikasi dengan sistem file Anda.

## Cara kerja enkripsi dalam perjalanan

Untuk mengaktifkan enkripsi data dalam perjalanan, Anda terhubung ke Amazon EFS menggunakan TLS. Sebaiknya gunakan EFS mount helper untuk memasang sistem file Anda

karena menyederhanakan proses pemasangan dibandingkan dengan pemasangan dengan NFS mount. Pembantu EFS mount mengelola proses menggunakan `stunnel` for TLS. Jika Anda tidak menggunakan mount helper, Anda masih dapat mengaktifkan enkripsi data dalam perjalanan. Pada tingkat tinggi, berikut adalah langkah-langkah untuk melakukannya.

Untuk mengaktifkan enkripsi data dalam perjalanan tanpa menggunakan EFS mount helper

1. Unduh dan instal `stunnel`, dan catat port yang sedang didengarkan aplikasi. Untuk instruksi untuk melakukannya, lihat [Upgrade stunnel](#).
2. Jalankan `stunnel` untuk terhubung ke sistem EFS file Amazon Anda di port 2049 menggunakan TLS.
3. Menggunakan NFS klien, pasang `localhost: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, `stunnel` proses yang digunakan oleh EFS mount helper mendengarkan pada port lokal mulai dari 20049 hingga 21049, dan terhubung ke Amazon pada port 2049. EFS

#### Note

Secara default, saat menggunakan helper EFS mount Amazon dengan TLS, mount helper memberlakukan pemeriksaan nama host sertifikat. Pembantu EFS pemasangan Amazon menggunakan `stunnel` program ini untuk TLS fungsinya. Beberapa versi Linux tidak menyertakan versi `stunnel` yang mendukung TLS fitur ini secara default. Saat menggunakan salah satu versi Linux tersebut, pemasangan sistem EFS file Amazon menggunakan TLS gagal.

Setelah Anda menginstal `amazon-efs-utils` paket, untuk meningkatkan versi `stunnel` lihat [Upgrade stunnel](#) sistem Anda.

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

Saat menggunakan enkripsi data dalam perjalanan, pengaturan NFS klien 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=6
```

Saat memasang dengan TLS dan penolong EFS pemasangan Amazon, Anda mengonfigurasi ulang NFS klien Anda untuk dipasang ke port lokal. EFSMount helper memulai `stunnel` proses klien yang mendengarkan pada port lokal ini, dan `stunnel` membuka koneksi terenkripsi ke sistem EFS file menggunakan TLS. EFSMount helper bertanggung jawab untuk mengatur dan memelihara koneksi terenkripsi ini dan konfigurasi terkait.

Untuk menentukan ID sistem EFS file Amazon mana yang sesuai dengan titik pemasangan lokal mana, Anda dapat menggunakan perintah berikut. Ganti `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
```

Bila Anda menggunakan mount helper untuk enkripsi data dalam perjalanan, itu juga menciptakan proses yang disebut `amazon-efs-mount-watchdog`. Proses ini memastikan bahwa setiap proses `stunnel` mount berjalan, dan menghentikan `stunnel` saat sistem EFS file Amazon dilepas. Jika karena alasan tertentu proses `stunnel` dihentikan secara tak terduga, proses pengawas memulai ulang.

## 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](#)
- [Sistem `ncrypted-at-rest` file E 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 .

## Sistem ncrpted-at-rest file E 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 Wilayah AWS. 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 .

## 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. IAM administrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (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 tertaut layanan untuk Amazon EFS](#)
- [Memecahkan masalah identitas dan akses Amazon Elastic File System](#)

## Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di AmazonEFS.

Pengguna layanan — Jika Anda menggunakan EFS layanan Amazon untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak EFS fitur Amazon untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AmazonEFS, lihat [Memecahkan masalah identitas dan akses Amazon Elastic File System](#).

Administrator layanan - Jika Anda bertanggung jawab atas sumber EFS daya Amazon di perusahaan Anda, Anda mungkin memiliki akses penuh ke AmazonEFS. Tugas Anda adalah menentukan EFS fitur dan sumber daya Amazon mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AmazonEFS, lihat [Bagaimana Amazon Elastic File System bekerja dengan IAM](#).

IAM administrator - Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke AmazonEFS. Untuk melihat contoh kebijakan EFS berbasis identitas Amazon yang dapat Anda gunakan, lihat. IAM [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

## Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai IAM pengguna, atau dengan mengambil peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (Pusat IAM Identitas), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Versi AWS Tanda Tangan 4 untuk API permintaan](#) di Panduan IAM Pengguna.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat [Autentikasi multi-faktor](#) di Panduan AWS IAM Identity Center Pengguna dan [Autentikasi AWS multi-faktor IAM di Panduan Pengguna](#). IAM

### Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas



yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensi pengguna root](#) di IAMPanduan Pengguna.

## Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat IAM Identitas, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat IAM Identitas, lihat [Apa itu Pusat IAM Identitas?](#) dalam AWS IAM Identity Center User Guide.

## Pengguna dan grup IAM

[IAMPengguna](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya mengandalkan kredensi sementara daripada membuat IAM pengguna yang memiliki kredensi jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan IAM pengguna, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat [Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensi jangka panjang](#) di IAMPanduan Pengguna.

[IAMGrup](#) adalah identitas yang menentukan kumpulan IAM pengguna. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup bernama IAMAdmins dan memberikan izin grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara.

Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk IAM pengguna](#) di Panduan IAM Pengguna.

## IAMperan

[IAMPeran](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Ini mirip dengan IAM pengguna, tetapi tidak terkait dengan orang tertentu. Untuk mengambil IAM peran sementara di dalam AWS Management Console, Anda dapat [beralih dari pengguna ke IAM peran \(konsol\)](#). Anda dapat mengambil peran dengan memanggil AWS CLI atau AWS API operasi atau dengan menggunakan kustomURL. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat [Metode untuk mengambil peran](#) dalam Panduan IAM Pengguna.

IAMperan dengan kredensi sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) di Panduan IAM Pengguna. Jika Anda menggunakan Pusat IAM Identitas, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, Pusat IAM Identitas menghubungkan izin yang disetel ke peran. Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin IAM pengguna sementara — IAM Pengguna atau peran dapat mengambil IAM peran untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun — Anda dapat menggunakan IAM peran untuk memungkinkan seseorang (prinsipal tepercaya) di akun lain mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM](#) Panduan Pengguna. IAM
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan

beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. FAS permintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat [Meneruskan sesi akses](#).

- Peran layanan — Peran layanan adalah [IAMperan](#) yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAM Administrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam IAM Panduan Pengguna.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. 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. IAM Administrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS API meminta. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon](#) di IAM Panduan Pengguna.

## Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai JSON dokumen. Untuk informasi selengkapnya tentang struktur dan isi dokumen JSON kebijakan, lihat [Ringkasan JSON kebijakan](#) di Panduan IAM Pengguna.

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

IAMkebijakan menentukan izin untuk tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasi. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan itu bisa mendapatkan informasi peran dari AWS Management Console, AWS CLI, atau AWS API.

## Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. 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 [Menentukan IAM izin khusus dengan kebijakan yang dikelola pelanggan di Panduan Pengguna](#). IAM

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam Akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan sebaris, lihat [Memilih antara kebijakan terkelola dan kebijakan sebaris](#) di IAMPanduan Pengguna.

## Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran 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 prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola IAM dalam kebijakan berbasis sumber daya.

## Daftar kontrol akses (ACLs)

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

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACLs. Untuk mempelajari selengkapnya ACLs, lihat [Ikhtisar daftar kontrol akses \(ACL\)](#) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

## Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- **Batas izin** — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas (pengguna atau peran). IAM Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batas izin, lihat [Batas izin untuk IAM entitas](#) di IAMPanduan Pengguna.
- **Kebijakan kontrol layanan (SCPs)** — SCPs adalah JSON kebijakan yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCPMembatasi izin untuk entitas di akun anggota, termasuk masing-masing Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat [Kebijakan kontrol layanan](#) di Panduan AWS Organizations Pengguna.
- **Kebijakan sesi** – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan

secara tegas dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) di Panduan IAM Pengguna.

## 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 IAM Pengguna.

## Bagaimana Amazon Elastic File System bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke AmazonEFS, pelajari IAM fitur apa saja yang tersedia untuk digunakan dengan AmazonEFS.

IAMfitur yang dapat Anda gunakan dengan Amazon Elastic File System

IAMfitur	EFS Dukungan Amazon
<a href="#">Kebijakan berbasis identitas</a>	Ya
<a href="#">Kebijakan berbasis sumber daya</a>	Ya
<a href="#">Tindakan kebijakan</a>	Ya
<a href="#">Sumber daya kebijakan</a>	Ya
<a href="#">kunci-kunci persyaratan kebijakan (spesifik layanan)</a>	Ya
<a href="#">ACLs</a>	Tidak
<a href="#">ABAC(tag dalam kebijakan)</a>	Parsial
<a href="#">Kredensial sementara</a>	Ya
<a href="#">Izin prinsipal</a>	Ya
<a href="#">Peran layanan</a>	Ya
<a href="#">Peran terkait layanan</a>	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon EFS dan AWS layanan lainnya dengan sebagian besar IAM fitur, lihat [AWS layanan yang berfungsi IAM](#) di Panduan IAM Pengguna.

## Kebijakan berbasis identitas untuk Amazon EFS

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. 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 [Menentukan IAM izin khusus dengan kebijakan yang dikelola pelanggan di Panduan Pengguna](#). IAM

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam JSON kebijakan, lihat [referensi elemen IAM JSON kebijakan](#) di Panduan IAM Pengguna.

Contoh kebijakan berbasis identitas untuk Amazon EFS

Untuk melihat contoh kebijakan EFS berbasis identitas Amazon, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

## Kebijakan berbasis sumber daya di Amazon EFS

Mendukung kebijakan berbasis sumber daya: Ya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran 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 prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau IAM entitas di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, IAM administrator di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun IAM di](#) Panduan IAM Pengguna.

Untuk mempelajari tentang menggunakan kebijakan sumber daya untuk mengontrol akses data sistem file, lihat [Menggunakan IAM untuk mengontrol akses data sistem file](#). Untuk mempelajari cara melampirkan kebijakan berbasis sumber daya ke sistem file, lihat [Membuat kebijakan sistem file](#)

Contoh kebijakan berbasis sumber daya di Amazon EFS

Untuk melihat contoh kebijakan EFS berbasis sumber daya Amazon, lihat [Contoh kebijakan berbasis sumber daya untuk Amazon EFS](#)

Tindakan kebijakan untuk Amazon EFS

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

ActionElemen JSON kebijakan menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan AWS API operasi terkait. Ada beberapa pengecualian, seperti tindakan khusus izin yang tidak memiliki operasi yang cocok. API Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar EFS tindakan Amazon, 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 EFS berbasis identitas Amazon, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

## Sumber daya kebijakan untuk Amazon EFS

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Resource` JSON kebijakan menentukan objek atau objek yang tindakan tersebut berlaku. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Sebagai praktik terbaik, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (\*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*" 
```

Untuk melihat daftar jenis EFS sumber daya Amazon dan jenisnya ARNs, lihat Sumber [daya yang ditentukan oleh Amazon Elastic File System](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh Amazon Elastic File System](#).

Untuk melihat contoh kebijakan EFS berbasis identitas Amazon, lihat. [Contoh kebijakan berbasis identitas untuk Amazon Elastic File System](#)

## Kunci kondisi kebijakan untuk Amazon EFS

Mendukung kunci kondisi kebijakan khusus layanan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. 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.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin IAM pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama IAM pengguna mereka. Untuk informasi selengkapnya, lihat [elemen IAM kebijakan: variabel dan tag](#) di Panduan IAM Pengguna.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan IAM Pengguna.

Untuk melihat daftar kunci EFS kondisi Amazon, 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 EFS berbasis identitas Amazon, 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

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke IAM entitas (pengguna atau peran) dan ke banyak AWS sumber daya. Menandai entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian Anda merancang ABAC kebijakan untuk mengizinkan operasi ketika tag prinsipal cocok dengan tag pada sumber daya yang mereka coba akses.

ABAC membantu dalam lingkungan yang berkembang pesat dan membantu dengan situasi di mana manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag 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 ABAC, lihat [Menentukan izin dengan ABAC otorisasi](#) di IAM Panduan Pengguna. Untuk melihat tutorial dengan langkah-langkah penyiapan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) di IAM Panduan Pengguna.

## Menggunakan kredensial sementara dengan Amazon EFS

Mendukung kredensi sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensial sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat [Layanan AWS yang berfungsi IAM](#) di IAM Panduan Pengguna.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS

menggunakan link sign-on (SSO) tunggal perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang beralih peran, lihat [Beralih dari pengguna ke IAM peran \(konsol\)](#) di Panduan IAM Pengguna.

Anda dapat secara manual membuat kredensial sementara menggunakan atau. AWS CLI AWS API Anda kemudian dapat menggunakan kredensial sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensi keamanan sementara](#) di IAM

## Izin utama lintas layanan untuk Amazon EFS

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. FAS permintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat [Meneruskan sesi akses](#).

## Peran layanan untuk Amazon EFS

Mendukung peran layanan: Ya

Peran layanan adalah [IAM peran](#) yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAM Administrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam IAM Panduan Pengguna.

### Warning

Mengubah izin untuk peran layanan dapat merusak EFS fungsionalitas Amazon. 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. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran EFS terkait layanan Amazon, lihat.

[Menggunakan peran tertaut 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 EFS sumber daya Amazon. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan IAM berbasis identitas menggunakan contoh dokumen kebijakan ini, lihat [Membuat JSON IAM kebijakan \(konsol\) di Panduan Pengguna](#). IAM

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh AmazonEFS, termasuk format ARNs untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Amazon Elastic File System](#) di Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan EFS konsol Amazon](#)
- [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 EFS sumber daya Amazon 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 AWSAWS terkelola](#) atau [kebijakan terkelola untuk fungsi pekerjaan](#) di Panduan IAM Pengguna.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya 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 penggunaan IAM untuk menerapkan izin, lihat [Kebijakan dan izin IAM di IAM](#) Panduan Pengguna.
- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [elemen IAM JSON kebijakan: Kondisi](#) dalam Panduan IAM Pengguna.
- Gunakan IAM Access Analyzer untuk memvalidasi IAM kebijakan Anda guna memastikan izin yang aman dan fungsional — IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa IAM kebijakan ( ) JSON dan praktik terbaik. IAM IAMAccess 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 [Memvalidasi kebijakan dengan IAM Access Analyzer](#) di IAMPanduan Pengguna.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan IAM pengguna atau pengguna root di Anda Akun AWS, aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA kapan API operasi dipanggil, tambahkan MFA kondisi ke kebijakan Anda. Untuk informasi selengkapnya, lihat [APIAkses aman dengan MFA](#) di Panduan IAM Pengguna.

Untuk informasi selengkapnya tentang praktik terbaik di IAM, lihat [Praktik terbaik keamanan IAM di Panduan IAM Pengguna](#).

## Menggunakan EFS konsol Amazon

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 EFS sumber daya Amazon di 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 cocok dengan API operasi yang mereka coba lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan EFS konsol Amazon, lampirkan juga kebijakan `AmazonElasticFileSystemReadOnlyAccess` AWS terkelola Amazon ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) di Panduan IAM Pengguna.

Anda dapat melihat `AmazonElasticFileSystemReadOnlyAccess` dan kebijakan layanan EFS terkelola Amazon lainnya di [AWS kebijakan terkelola untuk Amazon EFS](#).

### Contoh: Izinkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan IAM pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau secara terprogram menggunakan atau AWS CLI atau AWS API

```
{
  "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 AWS CLI,, atau AWS API atau SDK: AWS Management Console

```
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 AWS CLI,, atau AWS API atau SDK: AWS Management Console

```
User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

Anda juga dapat menerapkan pembuatan sistem EFS file Amazon terenkripsi atau tidak terenkripsi dengan membuat kebijakan kontrol layanan (). AWS Organizations 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 EFS tindakan Amazon. Kebijakan sistem EFS file Amazon memiliki batas 20.000 karakter. Untuk informasi tentang elemen kebijakan berbasis sumber daya, lihat [Kebijakan berbasis sumber daya di Amazon EFS](#)

### Important

Jika Anda memberikan izin kepada IAM pengguna 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 Prinsipal](#) di Panduan IAM Pengguna.

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 EFS akses](#)

### Contoh: Berikan akses baca dan tulis ke AWS peran tertentu

Dalam contoh ini, kebijakan sistem EFS file 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 tersebut. EfsReadOnly IAM

```

{
  "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 IAM prinsipal, kecuali untuk workstation manajemen tertentu, lihat [Aktifkan root squashing menggunakan IAM otorisasi untuk klien NFS](#)

## Contoh: Berikan akses ke titik EFS akses

Anda menggunakan kebijakan EFS akses untuk menyediakan NFS klien 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 akses ARN penuh ke sistem file.

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

```
{
  "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, pemutakhiran 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 API operasi baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola](#) di Panduan IAM Pengguna.

## AWS kebijakan terkelola: AmazonElasticFileSystemFullAccess

Anda dapat melampirkan AmazonElasticFileSystemFullAccess kebijakan ke IAM identitas Anda.

Kebijakan ini memberikan izin administratif yang memungkinkan akses penuh ke Amazon EFS dan akses ke AWS layanan terkait melalui AWS Management Console

### Detail izin

Kebijakan ini mencakup izin berikut.

- `elasticfilesystem`— Memungkinkan kepala sekolah 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 Zones, grup keamanan, subnet, virtual private cloud (VPCs) dan atribut yang VPC terkait dengan EFS sistem file Amazon di konsol Amazon. EFS
- `kms`— Memungkinkan prinsipal untuk mencantumkan alias untuk kunci AWS Key Management Service (AWS KMS) 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:GetMetricData",
        "ec2:CreateNetworkInterface",
        "ec2>DeleteNetworkInterface",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcs",
        "ec2:ModifyNetworkInterfaceAttribute",
        "elasticfilesystem:Backup",
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:CreateMountTarget",
        "elasticfilesystem:CreateTags",
        "elasticfilesystem:CreateAccessPoint",
        "elasticfilesystem:CreateReplicationConfiguration",
        "elasticfilesystem>DeleteFileSystem",
        "elasticfilesystem>DeleteMountTarget",
        "elasticfilesystem>DeleteTags",
        "elasticfilesystem>DeleteAccessPoint",
        "elasticfilesystem>DeleteFileSystemPolicy",
        "elasticfilesystem>DeleteReplicationConfiguration",
        "elasticfilesystem:DescribeAccountPreferences",
        "elasticfilesystem:DescribeBackupPolicy",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:DescribeFileSystemPolicy",
        "elasticfilesystem:DescribeLifecycleConfiguration",
        "elasticfilesystem:DescribeMountTargets",
        "elasticfilesystem:DescribeMountTargetSecurityGroups",
        "elasticfilesystem:DescribeReplicationConfigurations",
        "elasticfilesystem:DescribeTags",
        "elasticfilesystem:DescribeAccessPoints",
        "elasticfilesystem:ModifyMountTargetSecurityGroups",
        "elasticfilesystem:PutAccountPreferences",
        "elasticfilesystem:PutBackupPolicy",
        "elasticfilesystem:PutLifecycleConfiguration",
```

```

        "elasticfilesystem:PutFileSystemPolicy",
        "elasticfilesystem:UpdateFileSystem",
        "elasticfilesystem:UpdateFileSystemProtection",
        "elasticfilesystem:TagResource",
        "elasticfilesystem:UntagResource",
        "elasticfilesystem:ListTagsForResource",
        "elasticfilesystem:Restore",
        "kms:DescribeKey",
        "kms:ListAliases"
    ],

    "Sid": "ElasticFileSystemFullAccess",
    "Effect": "Allow",
    "Resource": "*"
},
{
    "Action": "iam:CreateServiceLinkedRole",
    "Sid": "CreateServiceLinkedRoleForEFS",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:AWSServiceName": [
                "elasticfilesystem.amazonaws.com"
            ]
        }
    }
}
]
}

```

## AWS kebijakan terkelola: AmazonElasticFileSystemReadOnlyAccess

Anda dapat melampirkan AmazonElasticFileSystemReadOnlyAccess kebijakan ke IAM identitas Anda.

Kebijakan ini memberikan akses baca saja ke Amazon EFS melalui AWS Management Console

Detail izin

Kebijakan ini mencakup izin berikut.

- `elasticfilesystem`— Memungkinkan prinsipal untuk mendeskripsikan atribut sistem EFS file Amazon, termasuk preferensi akun, kebijakan pencadangan dan sistem file, konfigurasi siklus hidup, target pemasangan dan grup keamanan, tag, dan titik akses mereka di konsol Amazon. EFS
- `cloudwatch`— Memungkinkan kepala sekolah untuk mengambil CloudWatch metrik dan menjelaskan alarm untuk metrik di konsol Amazon. EFS
- `ec2`— Memungkinkan kepala sekolah untuk melihat Availability Zone, antarmuka jaringan dan atributnya, grup keamanan, subnet, VPCs dan atributnya di konsol Amazon. EFS
- `kms`— Memungkinkan kepala sekolah untuk membuat daftar alias untuk kunci AWS KMS di konsol Amazon. EFS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ElasticFileSystemReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:GetMetricData",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcs",
        "elasticfilesystem:DescribeAccountPreferences",
        "elasticfilesystem:DescribeBackupPolicy",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:DescribeFileSystemPolicy",
        "elasticfilesystem:DescribeLifecycleConfiguration",
        "elasticfilesystem:DescribeMountTargets",
        "elasticfilesystem:DescribeMountTargetSecurityGroups",
        "elasticfilesystem:DescribeTags",
        "elasticfilesystem:DescribeAccessPoints",
        "elasticfilesystem:DescribeReplicationConfigurations",
        "elasticfilesystem:ListTagsForResource",
        "kms:ListAliases"
      ]
    }
  ],
}
```



```

    "Resource": "*"
  }
]
}

```

## AWS kebijakan terkelola: AmazonElasticFileSystemClientReadWriteAccess

Anda dapat melampirkan AmazonElasticFileSystemClientReadWriteAccess kebijakan ke IAM entitas.

Kebijakan ini memberikan akses klien baca dan tulis ke sistem EFS file Amazon. Kebijakan ini memungkinkan NFS klien untuk me-mount, membaca, dan menulis ke sistem EFS file Amazon.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:DescribeMountTargets"
      ],
      "Resource": "*"
    }
  ]
}

```

## 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 RSS umpan di EFS [Riwayat dokumen](#) halaman Amazon.

Perubahan	Deskripsi	Tanggal
Memperbarui ke kebijakan yang ada	<p>Kebijakan: <a href="#">AmazonElasticFileSystemReadOnlyAccess</a></p> <p>Amazon EFS menambahkan elemen Sid (ID pernyataan) ke pernyataan kebijakan. Nilai Sid</p>	Agustus 7, 2024


Perubahan	Deskripsi	Tanggal
	adalah <code>ElasticFileSystemReadOnlyAccess</code> .	
Memperbarui ke kebijakan yang ada	<p>Kebijakan: <a href="#">AmazonElasticFileSystemFullAccess</a></p> <p>Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal menonaktifkan dan mengaktifkan perlindungan pada sistem file. Izin diperlukan untuk memungkinkan Amazon mereplikasi EFS ke sistem file yang ada.</p>	27 November 2023
Memperbarui ke kebijakan yang ada	<p>Kebijakan: <a href="#">AmazonElasticFileSystemServiceRolePolicy</a></p> <p>Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal membuat, mendeskripsikan, dan menghapus EFS replikasi Amazon, dan untuk membuat sistem file Amazon. EFS Izin diperlukan untuk memungkinkan Amazon EFS mengelola konfigurasi replikasi sistem file atas nama pengguna.</p>	Januari 25, 2022
Memperbarui ke kebijakan yang ada	<p>Kebijakan: <a href="#">AmazonElasticFileSystemReadOnlyAccess</a></p> <p>Amazon EFS menambahkan izin baru untuk mengizinkan prinsipal menggambarkan replikasi Amazon. EFS Izin diperlukan untuk memungkinkan pengguna melihat konfigurasi replikasi sistem file.</p>	Januari 25, 2022
Memperbarui ke kebijakan yang ada	<p>Kebijakan: <a href="#">AmazonElasticFileSystemFullAccess</a></p> <p>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.</p>	Januari 25, 2022

Perubahan	Deskripsi	Tanggal
Mulai melacak kebijakan	Kebijakan: <a href="#">AmazonElasticFileSystemClientReadWriteAccess</a>  Memberikan hak baca dan tulis di sistem EFS file Amazon kepada NFS klien.	Januari 3, 2022
Mulai melacak kebijakan	Kebijakan: <a href="#">AmazonElasticFileSystemServiceRolePolicy</a>  Izin peran terkait layanan untuk Amazon. EFS	Oktober 8, 2021
Memperbarui ke kebijakan yang ada	Kebijakan: <a href="#">AmazonElasticFileSystemFullAccess</a>  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 EFS konsol Amazon.	7 Mei 2021
Memperbarui ke kebijakan yang ada	Kebijakan: <a href="#">AmazonElasticFileSystemReadOnlyAccess</a>  Amazon EFS menambahkan izin baru untuk memungkinkan prinsipal menggambarkan preferensi akun Amazon. EFS Izin diperlukan untuk memungkinkan pengguna melihat pengaturan preferensi akun di EFS konsol Amazon.	7 Mei 2021
Amazon EFS mulai melacak perubahan	Amazon EFS mulai melacak perubahan untuk kebijakan yang AWS dikelola.	7 Mei 2021

## Menggunakan tag dengan Amazon EFS

Anda dapat menggunakan tanda untuk mengontrol ke sumber daya Amazon EFS dan menerapkan kontrol akses berbasis atribut (ABAC). Untuk informasi selengkapnya, lihat:

- [Sumber daya penandaan EFS](#)
- [Mengontrol akses berdasarkan tag](#)
- [Apa itu ABACAWS?](#) dalam Panduan Pengguna IAM

 Note

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

Untuk menerapkan tag ke sumber daya Amazon EFS selama pembuatan, pengguna harus memiliki izin AWS Identity and Access Management (IAM) tertentu.

### Pemberian izin untuk memberi tag

Tag pada tindakan Amazon EFS API memungkinkan Anda menentukan tanda saat membuat sumber daya.

- `CreateAccessPoint`
- `CreateFileSystem`

Untuk memungkinkan pengguna menandai sumber daya pada pembuatan, mereka harus memiliki izin untuk menggunakan tindakan yang menciptakan sumber daya, seperti `elasticfilesystem:CreateAccessPoint` atau `elasticfilesystem:CreateFileSystem`. Jika tanda ditentukan dalam aksi pembuatan sumber daya, AWS melakukan otorisasi tambahan pada `elasticfilesystem:TagResource` tindakan untuk memverifikasi apakah pengguna memiliki izin untuk membuat tanda. 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 kebijakan: Izinkan menambahkan tanda ke sistem file hanya pada saat pembuatan

Kebijakan contoh berikut memungkinkan pengguna untuk membuat sistem file dan menerapkan tag kepada mereka hanya selama pembuatan. Para pengguna tidak diizinkan untuk memberi

tanda pada sumber daya yang sudah ada (mereka tidak dapat memerintahkan 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 tanda 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 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

Untuk mengontrol tindakan mana 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 kunci-nilai tag pada sumber daya.

Example policy: Buat sistem file hanya jika tag tertentu digunakan

Kebijakan contoh berikut memungkinkan pengguna untuk membuat sistem file hanya ketika mereka menandainya dengan pasangan kunci-nilai tag tertentu, dalam contoh inikey=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

Kebijakan contoh berikut memungkinkan pengguna untuk menghapus sistem file yang diberi tagDepartment=Finance.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem>DeleteFileSystem"
      ],
      "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*",
      "Condition": {
```

```
        "StringEquals": {
            "aws:ResourceTag/Department": "Finance"
        }
    }
}
]
```

## Menggunakan peran tertaut layanan untuk Amazon EFS

Amazon Elastic File System menggunakan [peran terkait layanan AWS Identity and Access Management](#) (IAM). Peran tertaut layanan Amazon EFS adalah jenis IAM role unik yang tertaut langsung dengan Amazon EFS. Peran tertaut layanan Amazon EFS yang telah ditentukan sebelumnya mencakup izin yang diperlukan layanan untuk memanggil orang lain Layanan AWS atas nama Anda.

Peran tertaut layanan memudahkan pengaturan Amazon EFS lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Amazon EFS menentukan izin peran terkait layanannya, dan hanya Amazon EFS yang dapat mengambil peran tersebut. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran tertaut layanan Amazon EFS hanya setelah pertama kali menghapus sistem file Amazon EFS Anda. Tindakan ini dapat melindungi sumber daya Amazon EFS karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Peran terkait layanan memungkinkan semua panggilan API dapat dilihat melalui AWS CloudTrail. Ini membantu persyaratan pemantauan dan audit karena Anda dapat melacak semua tindakan yang dilakukan Amazon EFS lakukan atas nama Anda. Untuk informasi selengkapnya, lihat [Entri log untuk peran terkait EFS layanan](#).

## Izin peran tertaut layanan untuk Amazon EFS

Amazon EFS menggunakan peran terkait layanan bernama `AWSServiceRoleForAmazonElasticFileSystem` untuk memungkinkan Amazon EFS memanggil dan mengelola AWS sumber daya atas nama sistem file EFS Anda.

`AWSServiceRoleForAmazonElasticFileSystem` peran terkait layanan memercayakan layanan berikut untuk menjalankan peran tersebut:

- `elasticfilesystem.amazonaws.com`

Kebijakan izin peran memungkinkan Amazon EFS menyelesaikan tindakan yang termasuk dalam definisi kebijakan 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"
    ],
    "Resource": "*"
  }
]

```

}

**Note**

Anda harus mengonfigurasi izin IAM secara manual untuk AWS KMS saat membuat sistem file Amazon EFS baru yang dienkripsi saat istirahat. Untuk mempelajari selengkapnya, lihat [Mengkripsi data saat istirahat](#).

## Membuat peran tertaut layanan untuk Amazon EFS

Anda harus mengkonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat peran terkait layanan. Lakukan ini dengan menambahkan `iam:CreateServiceLinkedRole` izin untuk entitas IAM seperti yang ditunjukkan pada contoh berikut.

```
{
  "Action": "iam:CreateServiceLinkedRole",
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "iam:AWSServiceName": [
        "elasticfilesystem.amazonaws.com"
      ]
    }
  }
}
```

Untuk informasi selengkapnya, lihat [Izin Peran Terkait Layanan](#) di Panduan Pengguna IAM.

Anda tidak perlu membuat peran terkait layanan secara manual. Ketika Anda membuat target kait atau konfigurasi replikasi untuk sistem file EFS Anda di AWS Management Console, AWS API, atau CLI, atau Amazon EFS membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Ketika Anda membuat target kait atau konfigurasi replikasi untuk sistem file EFS membuat peran terkait layanan untuk Anda kembali.

## Mengedit peran tertaut layanan untuk Amazon EFS

Amazon EFS tidak mengizinkan Anda untuk mengedit peran `AWSServiceRoleForAmazonElasticFileSystem` terkait layanan. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi lebih lanjut, lihat [Mengedit Peran Tertaut Layanan](#) di Panduan Pengguna IAM.

## Menghapus peran tertaut layanan untuk Amazon EFS

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami menyarankan 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.

### Note

Jika layanan Amazon EFS menggunakan peran saat Anda mencoba untuk menghapus sumber daya, maka penghapusan tersebut kemungkinan gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus sumber daya Amazon EFS yang digunakan oleh `AWSServiceRoleForAmazonElasticFileSystem`

Selesaikan langkah-langkah berikut untuk menghapus sumber daya Amazon EFS yang digunakan oleh `AWSServiceRoleForAmazonElasticFileSystem`. Untuk prosedur rinci, lihat [Bersihkan sumber daya dan lindungi AWS akun Anda](#).

1. Pada instans Amazon EC2 Anda, lepaskan sistem file Amazon EFS.
2. Hapus sistem file Amazon EFS.
3. Hapus grup keamanan khusus untuk sistem file.

### Warning

Jika Anda menggunakan grup keamanan default untuk virtual private cloud (VPC) Anda, jangan hapus grup keamanan tersebut.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, AWS CLI, atau AWS API untuk menghapus peran terkait layanan `AWSServiceRoleForAmazonElasticFileSystem`. Untuk informasi lebih lanjut, lihat [Menghapus Peran Tertaut Layanan](#) dalam Panduan Pengguna IAM.

## 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 untuk mengakses EFS sumber daya Amazon 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 `mateojackson` IAM pengguna mencoba menggunakan konsol untuk melihat detail tentang `my-example-widget` sumber daya fiksi tetapi tidak memiliki izin `elasticfilesystem:GetWidget` fiksi.

```
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 AmazonEFS.

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 IAM pengguna bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di AmazonEFS. 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 untuk mengakses EFS sumber daya Amazon 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 IAM pengguna lain Akun AWS yang Anda miliki](#) di Panduan IAM Pengguna.

- 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 IAM Pengguna.
- Untuk mempelajari cara menyediakan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna yang diautentikasi secara eksternal \(federasi identitas\) di 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](#) Panduan Pengguna. IAM

## Menggunakan IAM untuk mengontrol akses data sistem file

Anda dapat menggunakan kebijakan IAM identitas dan kebijakan sumber daya untuk mengontrol akses klien ke EFS sumber daya Amazon dengan cara yang dapat diskalakan dan dioptimalkan untuk lingkungan cloud. Dengan 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 IAM identitas atau kebijakan sumber daya sistem file memungkinkan akses untuk tindakan tersebut. Izin tidak perlu diberikan baik dalam identitas maupun kebijakan sumber daya.

NFS klien dapat mengidentifikasi diri mereka menggunakan IAM peran saat menghubungkan ke sistem EFS file. Saat klien terhubung ke sistem file, Amazon EFS mengevaluasi kebijakan IAM sumber daya sistem file, yang disebut kebijakan sistem file, bersama dengan IAM kebijakan berbasis identitas apa pun untuk menentukan izin akses sistem file yang sesuai untuk diberikan.

Saat Anda menggunakan IAM otorisasi untuk NFS klien, koneksi klien dan keputusan IAM otorisasi dicatat. AWS CloudTrail Untuk informasi selengkapnya tentang cara mencatat EFS API panggilan Amazon CloudTrail, lihat [Mencatat EFS API panggilan Amazon dengan AWS CloudTrail](#).

### Important

Anda harus menggunakan EFS mount helper untuk me-mount sistem EFS file Amazon Anda agar dapat menggunakan IAM otorisasi untuk mengontrol akses klien. Untuk informasi selengkapnya, lihat [Pemasangan dengan IAM otorisasi](#).

## Kebijakan sistem EFS file default

Kebijakan sistem EFS file default tidak digunakan 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, [DescribeFileSystemPolicy](#) API operasi mengembalikan PolicyNotFound respons.

## EFStindakan 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.

## EFSkunci kondisi untuk klien

Untuk menyatakan kondisi, Anda menggunakan kunci kondisi standar. Amazon EFS memiliki kunci kondisi standar berikut untuk NFS klien. Kunci kondisi lainnya tidak diberlakukan saat menggunakan IAM kontrol untuk mengamankan akses ke sistem EFS file.

EFSKunci Kondisi	Deskripsi	Operator
<code>aws:SecureTransport</code>	Gunakan kunci ini untuk meminta klien untuk menggunakan TLS saat menghubungkan ke sistem EFS file.	Boolean

EFSKunci Kondisi	Deskripsi	Operator
<code>aws:SourceIp</code>	Alamat IP pribadi klien yang mengakses sistem EFS file.	String
<code>elasticfilesystem:AccessPointArn</code>	ARN dari titik EFS akses yang terhubung dengan klien.	String
<code>elasticfilesystem:AccessedViaMountTarget</code>	Gunakan kunci ini untuk mencegah akses ke sistem EFS file oleh klien yang tidak menggunakan target pemasangan sistem file.	Boolean

## Contoh kebijakan sistem file

Untuk melihat contoh kebijakan sistem EFS file Amazon, lihat [Contoh kebijakan berbasis sumber daya untuk Amazon EFS](#).

## Mengontrol akses jaringan ke sistem EFS file Amazon untuk NFS klien

Anda dapat mengontrol akses oleh NFS klien ke sistem EFS file Amazon menggunakan keamanan lapisan jaringan dan kebijakan sistem EFS file. Anda dapat menggunakan mekanisme keamanan lapisan jaringan yang tersedia dengan Amazon EC2, seperti aturan grup VPC keamanan dan jaringan ACLs. Anda juga dapat menggunakan AWS IAM untuk mengontrol NFS akses dengan kebijakan sistem EFS file dan kebijakan berbasis identitas.

### Topik

- [Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target](#)
- [Port sumber untuk bekerja dengan EFS](#)
- [Pertimbangan keamanan untuk akses jaringan](#)
- [Bekerja dengan VPC titik akhir antarmuka di Amazon EFS](#)



## Menggunakan grup VPC keamanan untuk EC2 instans Amazon dan memasang target

Saat menggunakan AmazonEFS, Anda menentukan grup EC2 keamanan Amazon untuk EC2 instans dan grup keamanan untuk target EFS pemasangan 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 EC2 instance. Anda kemudian mengaitkan yang lain dengan target EFS pemasangan (yaitu, grup keamanan default untuk default AndaVPC). Pendekatan itu bekerja untuk latihan Memulai. Namun, untuk sistem produksi, Anda harus mengatur grup keamanan dengan izin minimal untuk digunakan. EFS

Anda dapat mengotorisasi akses masuk dan keluar ke sistem file AndaEFS. Untuk melakukannya, Anda menambahkan aturan yang memungkinkan EC2 instans Anda terhubung ke sistem EFS file Amazon Anda melalui target pemasangan menggunakan port Network File System (NFS). Ambil langkah-langkah berikut untuk membuat dan memperbarui grup keamanan Anda.

Untuk membuat grup keamanan untuk EC2 instans dan memasang target

1. Buat dua grup keamanan di AndaVPC.

Untuk petunjuk, lihat prosedur “Untuk membuat grup keamanan” di [Membuat Grup Keamanan](#) di Panduan VPC Pengguna Amazon.

2. Buka Amazon VPC Management Console di <https://console.aws.amazon.com/vpc/>, dan verifikasi aturan default untuk grup keamanan ini. Kedua kelompok keamanan seharusnya hanya memiliki aturan keluar yang memungkinkan lalu lintas pergi.

Untuk memperbarui akses yang diperlukan untuk grup keamanan Anda

1. Buka VPC konsol Amazon di <https://console.aws.amazon.com/vpc/>.
2. Tambahkan aturan untuk grup EC2 keamanan Anda untuk mengizinkan akses masuk menggunakan Secure Shell (SSH) dari host mana pun. Secara opsional, batasi alamat Sumber.

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

Untuk petunjuk, lihat [Menambahkan dan Menghapus Aturan](#) di Panduan VPC Pengguna Amazon.

3. Tambahkan aturan masuk dan keluar untuk target pemasangan.
  - Tambahkan aturan masuk untuk grup keamanan target mount untuk mengizinkan akses masuk dari grup EC2 keamanan. Identifikasi kelompok EC2 keamanan sebagai sumbernya.
  - Tambahkan aturan keluar untuk membuka TCP koneksi di semua NFS port. Identifikasi kelompok EC2 keamanan sebagai tujuan.

Untuk petunjuk, lihat [Menambahkan dan Menghapus Aturan](#) di Panduan VPC Pengguna Amazon.

4. Verifikasi bahwa kedua grup keamanan sekarang mengotorisasi akses masuk dan keluar.

Untuk informasi selengkapnya tentang grup keamanan, lihat [Grup EC2 keamanan Amazon untuk instans Linux](#).

## Port sumber untuk bekerja dengan EFS

Untuk mendukung serangkaian NFS klien yang luas, Amazon EFS mengizinkan koneksi dari port sumber apa pun. Jika Anda mengharuskan hanya pengguna istimewa yang dapat mengakses AmazonEFS, sebaiknya gunakan aturan firewall klien berikut. Connect ke sistem file Anda menggunakan SSH dan menjalankan 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 OUTPUT rantai (-I OUTPUT 1). Aturan mencegah proses nonkernel (-m owner --uid-owner 1-4294967294) yang tidak memiliki hak istimewa membuka koneksi ke NFS port (). -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 NFS port (TCPport 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 target mount VPC — 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:
  - EC2Contoh Amazon di Amazon lokal VPC
  - EC2contoh dalam terhubung VPCs
  - Server lokal yang terhubung ke Amazon VPC dengan menggunakan AWS Direct Connect dan AWS Virtual Private Network () VPN
- Daftar kontrol akses jaringan (ACLs) untuk VPC subnet klien dan target mount, untuk akses dari luar subnet target mount — Untuk memasang sistem file, klien harus dapat membuat TCP koneksi ke NFS port target mount dan menerima lalu lintas kembali.
- Aturan grup VPC keamanan klien dan mount target, untuk semua akses — Untuk EC2 instance yang 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 NFS port dari instance. Anda dapat mengaktifkan koneksi masuk baik dengan alamat IP (CIDRrentang) atau grup keamanan. Sumber aturan grup keamanan untuk NFS port masuk pada antarmuka jaringan target mount adalah elemen kunci dari kontrol akses sistem file. Aturan masuk selain yang untuk NFS port, 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 NFS port pada salah satu target pemasangan sistem file. Anda dapat mengaktifkan koneksi keluar baik dengan alamat IP (CIDRrentang) atau grup keamanan.

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

## Bekerja dengan VPC titik akhir antarmuka di Amazon EFS

Untuk membuat koneksi pribadi antara virtual private cloud (VPC) dan Amazon EFSAPI, Anda dapat membuat VPC titik akhir antarmuka. Titik akhir menyediakan konektivitas aman ke Amazon EFS API tanpa memerlukan gateway internet, NAT instance, atau koneksi jaringan pribadi virtual (VPN). Untuk informasi selengkapnya, lihat [VPCTitik Akhir Antarmuka](#) di Panduan VPC Pengguna Amazon.

VPCEndpoint antarmuka didukung oleh AWS PrivateLink, fitur yang memungkinkan komunikasi pribadi antara AWS layanan menggunakan alamat IP pribadi. Untuk menggunakannya AWS

PrivateLink, buat VPC titik akhir antarmuka untuk Amazon EFS di Anda VPC menggunakan VPC konsol Amazon, API, atau CLI. Melakukan hal ini menciptakan sebuah elastic network interface di subnet Anda dengan alamat IP pribadi yang melayani EFS API permintaan Amazon. Anda juga dapat mengakses VPC titik akhir dari lingkungan lokal atau dari VPCs penggunaan AWS VPN, AWS Direct Connect, atau VPC peering lainnya. Untuk mempelajari selengkapnya, lihat [Mengakses Layanan Melalui AWS PrivateLink](#) Panduan VPC Pengguna Amazon.

## Membuat titik akhir antarmuka untuk Amazon EFS

Untuk membuat VPC titik akhir antarmuka untuk Amazon EFS, gunakan salah satu dari berikut ini:

- **com.amazonaws.region.elasticfilesystem**— Membuat titik akhir untuk EFS API operasi Amazon.
- **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 EFS titik akhir Amazon, lihat [Amazon Elastic File System](#) di bagian. Referensi Umum Amazon Web Services

Untuk informasi selengkapnya tentang cara membuat titik akhir antarmuka, lihat [Membuat titik akhir antarmuka](#) di VPC Panduan Pengguna Amazon.

## Membuat kebijakan VPC endpoint untuk Amazon EFS

Untuk mengontrol akses ke Amazon EFS API, Anda dapat melampirkan kebijakan AWS Identity and Access Management (IAM) ke VPC titik akhir 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 Layanan dengan VPC Titik Akhir](#) di Panduan VPC Pengguna Amazon.

Contoh berikut menunjukkan kebijakan VPC endpoint yang menolak izin semua orang untuk membuat sistem EFS file melalui endpoint. 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": "*"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat [Menggunakan Kebijakan VPC Titik Akhir](#) di Panduan VPC Pengguna Amazon.

## Network File System (NFS) tingkat pengguna, grup, dan izin

Setelah membuat sistem file, secara default hanya pengguna root (UID0) 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 EFS akses Amazon](#).

Objek sistem EFS file Amazon 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 EFS bagaimana Amazon berperilaku sehubungan dengan izin ini.

Selain itu, pada sistem bergaya Unix, pengguna dan grup dipetakan ke pengidentifikasi numerik, yang digunakan EFS Amazon untuk mewakili kepemilikan file. Untuk AmazonEFS, 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**

NFSProtokol mendukung maksimum 16 grup IDs (GIDs) per pengguna dan tambahan apapun GIDs dipotong dari permintaan NFS klien. Untuk informasi selengkapnya, lihat [Akses ditolak ke file yang diizinkan pada sistem NFS file](#).

Berikut ini, Anda dapat menemukan contoh izin dan diskusi tentang pertimbangan NFS izin untuk Amazon. EFS

## Topik

- [Izin file dan direktori](#)
- [Contoh kasus penggunaan dan izin sistem EFS file Amazon](#)
- [Izin ID pengguna dan grup untuk file dan direktori dalam sistem file](#)
- [Tidak ada perencanaan akar](#)
- [Caching izin](#)
- [Mengubah kepemilikan objek sistem file](#)
- [EFStitik akses](#)

## Izin file dan direktori

File dan direktori dalam sistem EFS file mendukung standar UNIX gaya baca, tulis, dan eksekusi izin berdasarkan pengguna dan ID 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 AWS Identity and Access Management (IAM) kebijakan berbasis sumber daya dan kebijakan identitas untuk mengotorisasi NFS klien dan memberikan izin akses read-only, write, dan root. Anda dapat menggunakan titik EFS akses untuk mengganti pengguna sistem operasi dan informasi identitas grup yang disediakan oleh NFS klien. Untuk informasi selengkapnya, silakan lihat [Menggunakan IAM untuk mengontrol akses data 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 EFS file Amazon

Setelah Anda membuat sistem EFS file Amazon dan memasang target untuk sistem file di AndaVPC, Anda dapat memasang sistem file jarak jauh secara lokal di EC2 instans Amazon 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 EFS file Amazon, diidentifikasi oleh DNS nama sistem file, pada direktori `/efs-mount-point` lokal.

```
sudo mount -t nfs -o
nfsvers=4.1,rsize=1048576,wsize=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 EFS file Amazon. Seorang pengguna dengan hak akses root dapat memberikan pengguna lain akses ke sistem file.
  - Untuk mengubah kepemilikan sistem EFS file Amazon 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 EC2 instance yang memiliki sistem file yang terpasang.

## Izin ID pengguna dan grup untuk file dan direktori dalam sistem file

File dan direktori dalam sistem EFS file Amazon mendukung izin baca, tulis, dan eksekusi standar Unix berdasarkan ID pengguna dan grup. IDs Ketika NFS klien memasang sistem EFS file tanpa menggunakan titik akses, ID pengguna dan ID grup yang disediakan oleh klien dipercaya. Anda dapat menggunakan titik EFS akses untuk mengganti ID pengguna dan grup yang IDs digunakan oleh NFS klien. Ketika 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 hanya menggunakan pengidentifikasi numerik.

### Note

Saat Anda membuat pengguna pada sebuah EC2 instance, Anda dapat menetapkan ID pengguna numerik (UID) dan ID grup (GID) ke 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 IDs Di luar EC2 instance, Amazon EFS tidak melakukan otentikasi apa pun IDs, termasuk ID root 0.

Jika pengguna mengakses sistem EFS file Amazon dari dua EC2 instance yang berbeda, tergantung pada apakah UID untuk pengguna sama atau berbeda pada instance tersebut, Anda melihat perilaku yang berbeda, sebagai berikut:

- Jika pengguna IDs sama pada kedua EC2 instance, Amazon EFS menganggap mereka menunjukkan pengguna yang sama, terlepas dari EC2 instance yang digunakan. Pengalaman pengguna saat mengakses sistem file sama dari kedua EC2 instance.



- Jika pengguna IDs tidak sama pada kedua EC2 instance, Amazon EFS menganggap pengguna sebagai pengguna yang berbeda. Pengalaman pengguna tidak sama saat mengakses sistem EFS file Amazon dari dua EC2 instance yang berbeda.
- Jika dua pengguna berbeda pada EC2 instans yang berbeda berbagi ID, Amazon EFS menganggap mereka sebagai pengguna yang sama.

Anda dapat mempertimbangkan untuk mengelola pemetaan ID pengguna di seluruh EC2 instance secara konsisten. Pengguna dapat memeriksa ID numerik mereka menggunakan `id` perintah.

```
$ id
uid=502(joe) gid=502(joe) groups=502(joe)
```

## Matikan ID Mapper

NFSUtilitas 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 pengguna dan grup IDs menjadi nama, dan sebaliknya. Namun, Amazon hanya EFS berurusan dengan numerikIDs. Kami menyarankan Anda mematikan proses ini pada EC2 instans 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 perencanaan akar

Secara default, root squashing dinonaktifkan pada sistem EFS file. Amazon EFS berperilaku seperti NFS server Linux 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 di server. NFS

Untuk informasi selengkapnya, lihat [Menggunakan IAM untuk mengontrol akses data sistem file](#).

## Aktifkan root squashing menggunakan IAM otorisasi untuk klien NFS

Anda dapat mengonfigurasi Amazon EFS untuk mencegah akses root ke sistem EFS file Amazon Anda untuk semua AWS prinsipal kecuali untuk satu workstation manajemen. Anda melakukan ini dengan mengkonfigurasi AWS Identity and Access Management (IAM) otorisasi untuk klien Network File System (NFS).

Untuk melakukan hal ini memerlukan konfigurasi dua kebijakan IAM izin, sebagai berikut:

- Buat kebijakan sistem EFS file yang secara eksplisit memungkinkan akses baca dan tulis ke sistem file, dan secara implisit menolak akses root.
- Tetapkan IAM identitas ke workstation EC2 manajemen Amazon yang memerlukan akses root ke sistem file dengan menggunakan profil EC2 instans Amazon. Untuk informasi selengkapnya tentang profil EC2 instans Amazon, lihat [Menggunakan Profil Instans](#) di Panduan AWS Identity and Access Management Pengguna.
- Tetapkan kebijakan yang `AmazonElasticFileSystemClientFullAccess` AWS dikelola untuk IAM peran workstation manajemen. Untuk informasi selengkapnya tentang kebijakan AWS terkelolaEFS, lihat [Manajemen identitas dan akses untuk Amazon EFS](#).

Untuk mengaktifkan root squashing menggunakan IAM otorisasi untuk NFS klien, 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. JSONObjek kebijakan 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 IAM konsol di <https://console.aws.amazon.com/iam/>.
2. Buat peran untuk Amazon yang EC2 disebut `EFS-client-root-access`. IAM membuat profil instance dengan nama yang sama dengan EC2 peran yang Anda buat.
3. Tetapkan kebijakan AWS terkelola `AmazonElasticFileSystemClientFullAccess` ke EC2 peran yang Anda buat. Isi kebijakan ini ditampilkan sebagai berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:DescribeMountTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

4. Lampirkan profil instance ke EC2 instance yang Anda gunakan sebagai workstation manajemen, seperti yang dijelaskan berikut ini. Untuk informasi selengkapnya, lihat [Melampirkan IAM Peran ke Instance](#) di Panduan EC2 Pengguna Amazon untuk Instans Linux.
  - a. Buka EC2 konsol Amazon di <https://console.aws.amazon.com/ec2/>.
  - b. Di panel navigasi, pilih Instans.
  - c. Pilih instance. Untuk Tindakan, pilih Pengaturan Instance, lalu pilih peran Lampirkan/Ganti IAM.
  - d. Pilih IAM peran 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 dan `amazon-efs-utils` paketnya, lihat [Menginstal EFS klien Amazon](#).

6. Pasang sistem EFS file 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 EC2 instans Amazon untuk secara otomatis memasang sistem file dengan IAM otorisasi. Untuk informasi selengkapnya tentang pemasangan sistem EFS file dengan IAM otorisasi, lihat [Pemasangan dengan IAM otorisasi](#).

## Caching izin

Amazon EFS menyimpan izin file untuk jangka waktu 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 POSIX `chown_restricted` atribut. 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.

## EFStitik akses

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 diberikan oleh NFS klien. 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 EFS akses Amazon](#).

## Bekerja dengan titik EFS akses Amazon

Titik EFS akses Amazon adalah titik masuk khusus aplikasi ke dalam sistem EFS file yang membuatnya lebih mudah untuk mengelola akses aplikasi ke kumpulan data bersama. Titik akses dapat menegakkan identitas pengguna, termasuk POSIX grup pengguna, untuk 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 AWS Identity and Access Management (IAM) kebijakan untuk menegakkan bahwa aplikasi tertentu menggunakan titik akses tertentu. Dengan menggabungkan IAM kebijakan dengan titik akses, Anda dapat dengan mudah memberikan akses aman ke kumpulan data tertentu untuk aplikasi Anda.

#### Note

Anda perlu membuat setidaknya satu target mount pada sistem EFS file Anda untuk menggunakan titik akses.

Anda dapat membuat titik akses untuk sistem EFS file Amazon yang ada menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), dan file EFSAPI. Sistem EFS file Amazon dapat memiliki [maksimum 1.000 titik akses](#). Anda tidak dapat mengubah jalur akses yang ada setelah dibuat.

Untuk step-by-step prosedur untuk membuat titik akses, lihat [Membuat 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 EFS akses](#).

#### Topik

- [Menegakkan identitas pengguna menggunakan titik akses](#)
- [Menegakkan direktori root dengan titik akses](#)
- [Menggunakan titik akses dalam IAM kebijakan](#)

## 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 POSIX pengguna numerik untuk pengguna.
- ID Grup — ID POSIX grup numerik untuk pengguna.
- Kelompok sekunder IDs — Daftar opsional kelompok sekunderIDs.

Saat penegakan pengguna diaktifkan, Amazon EFS mengganti pengguna dan grup NFS klien 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.
- EFSmempertimbangkan ID pengguna, ID grup, dan grup sekunder IDs dari titik akses saat mengevaluasi izin sistem file. EFSmengabaikan NFS klien. IDs

### Important

Menegakkan identitas pengguna tunduk pada `ClientRootAccess` IAM izin. 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 `ClientRootAccess` IAM izin kepada NFS klien.

## Menegakkan direktori root dengan titik akses

Anda dapat menggunakan titik akses untuk mengganti direktori root untuk sistem file. Saat Anda menerapkan direktori root, NFS klien 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 NFS klien 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 tidak EFS 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 POSIX pengguna numerik untuk digunakan sebagai pemilik direktori root.
- `OwnerGid`— ID POSIX grup numerik untuk digunakan sebagai grup pemilik direktori root.
- `Izin` — Mode Unix dari 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`, `MilikGID`, 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 buat ulang saat berikutnya sistem file dipasang menggunakan titik akses.

#### Note

Jika Anda tidak menentukan kepemilikan dan izin untuk direktori root titik akses, Amazon tidak EFS 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 NFS server Linux dengan `no_subtree_check` opsi diaktifkan.

Dalam NFS protokol, server menghasilkan pegangan file yang digunakan oleh klien sebagai referensi unik saat mengakses file. EFS menghasilkan pegangan file dengan aman yang tidak dapat diprediksi dan spesifik untuk sistem EFS file. Ketika penggantian direktori root ada, EFS tidak mengungkapkan pegangan 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 IAM kebijakan

Anda dapat menggunakan IAM kebijakan untuk menegakkan bahwa NFS klien tertentu, yang diidentifikasi berdasarkan IAM perannya, hanya dapat mengakses titik akses tertentu. Untuk melakukan ini, Anda menggunakan tombol `elasticfilesystem:AccessPointArn` IAM kondisi. `AccessPointArn` ini adalah Nama Sumber Daya Amazon (ARN) dari titik akses tempat sistem file dipasang.



Berikut ini adalah contoh kebijakan sistem file yang memungkinkan IAM peran app1 untuk mengakses sistem file menggunakan titik aksesfsap-01234567. Kebijakan ini juga memungkinkan app2 untuk menggunakan sistem file menggunakan titik aksesfsap-89abcdef.

```
{
  "Version": "2012-10-17",
  "Id": "MyFileSystemPolicy",
  "Statement": [
    {
      "Sid": "App1Access",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/app1" },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite"
      ],
      "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"
      ],
      "Condition": {
        "StringEquals": {
          "elasticfilesystem:AccessPointArn" : "arn:aws:elasticfilesystem:us-east-1:222233334444:access-point/fsap-89abcdef"
        }
      }
    }
  ]
}
```

## Memblokir akses publik ke sistem EFS file

Fitur akses publik EFS blok Amazon menyediakan pengaturan untuk membantu Anda mengelola akses publik ke sistem EFS file. Secara default, sistem EFS file baru tidak mengizinkan akses publik. Namun, Anda dapat memodifikasi 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, AWS Key Management Service (KMS) kunci)

### Topik

- [Memblokir akses publik dengan AWS Transfer Family](#)
- [Arti “publik”](#)

## Memblokir akses publik dengan AWS Transfer Family

Ketika Anda menggunakan Amazon EFS dengan AWS Transfer Family, permintaan akses sistem file yang diterima dari server Transfer Family yang dimiliki oleh akun yang berbeda dari sistem file diblokir jika sistem file memungkinkan akses publik. Amazon EFS mengevaluasi IAM kebijakan sistem file, dan jika kebijakan tersebut bersifat publik, Amazon 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 EFS file 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:

- Satu set Classless Inter-Domain Routings (CIDRs), menggunakan `aws:SourceIp` Untuk informasi lebih lanjut tentang CIDR, lihat [RFC4632](#) di situs web RFC Editor.
- 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:*"`

Di bawah aturan ini, contoh kebijakan berikut dianggap publik.

```
{
  "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"
    ]
}

```

Anda dapat membuat kebijakan sistem file ini non-publik dengan menggunakan kunci EFS kondisi yang `elasticfilesystem:AccessedViaMountTarget` disetel ke `true`. Anda dapat menggunakan `elasticfilesystem:AccessedViaMountTarget` untuk mengizinkan EFS tindakan yang ditentukan kepada klien yang mengakses sistem EFS file menggunakan target pemasangan sistem file. Kebijakan non-publik berikut menggunakan kunci `elasticfilesystem:AccessedViaMountTarget` kondisi yang disetel ke `true`.

```

{
  "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"
      ],
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    }
  ]
}

```

Untuk informasi selengkapnya tentang kunci EFS kondisi Amazon, lihat [EFS kunci kondisi untuk klien](#). Untuk informasi selengkapnya tentang membuat kebijakan sistem file, lihat [Membuat kebijakan sistem file](#).

## 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. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk HIPAA Keamanan dan Kepatuhan di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat HIPAA aplikasi yang memenuhi syarat.

### Note

Tidak semua Layanan AWS HIPAA memenuhi syarat. Untuk informasi selengkapnya, lihat [Referensi Layanan yang HIPAA Memenuhi Syarat](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi ()). ISO
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber

daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).

- [Amazon GuardDuty](#) — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- [AWS Audit Manager](#) Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

## 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 EFS file Amazon tahan terhadap satu atau beberapa kegagalan Availability Zone dalam file. Wilayah AWS 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](#).

## 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 API panggilan yang AWS dipublikasikan untuk mengakses Amazon EFS melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Transportasi (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju yang sempurna (PFS) seperti (Ephemeral Diffie-Hellman) atau DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

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 EFS kebijakan Amazon untuk mengontrol akses dari titik akhir Amazon Virtual Private Cloud (AmazonVPC) tertentu, atau spesifikVPCs. Secara efektif, ini mengisolasi akses jaringan ke EFS sumber daya Amazon tertentu hanya dari yang spesifik VPC dalam AWS jaringan.

# EFSKuota Amazon

Berikut ini, Anda dapat mengetahui tentang kuota saat bekerja dengan AmazonEFS.

Topik

- [EFSKuota Amazon yang dapat Anda tingkatkan](#)
- [Kuota EFS sumber daya Amazon yang tidak dapat Anda ubah](#)
- [Kuota untuk klien NFS](#)
- [Kuota untuk sistem EFS file Amazon](#)
- [Fitur NFSv4 0 dan 4.1 yang tidak didukung](#)
- [Pertimbangan tambahan](#)
- [Memecahkan masalah kesalahan operasi file yang terkait dengan kuota](#)

## EFSKuota Amazon 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 semua nilai EFS batas Amazon dan meminta peningkatan kuota untuk jumlah sistem EFS file dalam file. Wilayah AWS

Anda juga dapat meminta peningkatan untuk EFS kuota Amazon berikut dengan menghubungi AWS Support. Untuk mempelajari selengkapnya, lihat [Meminta peningkatan kuota](#). Tim EFS layanan Amazon meninjau setiap permintaan secara individual.

- Jumlah sistem file untuk setiap akun pelanggan.
- Kuota throughput elastis per sistem file Regional untuk semua klien yang terhubung dalam file. Wilayah AWS
- Kuota throughput yang disediakan per sistem file Regional untuk semua klien yang terhubung dalam file. Wilayah AWS

Tabel berikut mencantumkan kuota default untuk setiap sumber daya yang dapat Anda ubah.



## Jumlah sistem file per akun pelanggan

Sumber Daya	Kuota bawaan
Jumlah sistem file untuk setiap akun pelanggan dalam Wilayah AWS	1.000

Sistem file regional — Total throughput Elastis default per sistem file untuk semua klien yang terhubung di masing-masing Wilayah AWS

Wilayah AWS	Throughput baca maksimum	Throughput penulisan maksimum (throughput terukur)
Wilayah AS Timur (Ohio)	60 gibibyte per detik ( ) GiBps	5 GiBps
Wilayah AS Timur (Virginia Utara)		
Wilayah AS Barat (Oregon)		
Wilayah Asia Pasifik (Singapura)		
Wilayah Asia Pacific (Tokyo)		
Wilayah Europe (Frankfurt)		
Wilayah Eropa (Irlandia)		
Semua lainnya Wilayah AWS	10 GiBps	1 GiBps

Sistem file regional - Total default throughput yang disediakan per sistem file untuk semua klien yang terhubung di masing-masing Wilayah AWS

Wilayah AWS	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)		
Wilayah Europe (Irlandia)		
Semua lainnya Wilayah AWS	3 GiBps	1 GiBps

## Meminta peningkatan kuota

Untuk meminta peningkatan kuota ini AWS Support, lakukan langkah-langkah berikut. EFSTim Amazon meninjau setiap permintaan peningkatan kuota.

Meminta kenaikan kuota melalui AWS Support

1. Buka halaman [AWS Support Tengah](#), dan masuk jika perlu. Kemudian pilih Buat Kasus.
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 EFS sumber daya Amazon yang tidak dapat Anda ubah

Kuota untuk beberapa EFS sumber daya Amazon tidak dapat diubah, termasuk:

- Kuota untuk sumber daya umum, seperti jumlah titik akses atau koneksi untuk setiap sistem file.
- Kuota throughput Elastis dan Provisioned per sistem file One Zone untuk semua klien yang terhubung dalam file. Wilayah AWS

- Kuota throughput yang meledak per sistem file Regional atau Satu Zona untuk semua klien yang terhubung dalam file. Wilayah AWS

Tabel berikut mencantumkan kuota sumber daya umum, batas throughput sistem file One Zone, dan batas throughput Bursting yang tidak dapat diubah.

Kuota sumber daya umum yang tidak dapat diubah

Sumber Daya	Kuota
Jumlah titik akses untuk setiap sistem file	1.000
Jumlah koneksi untuk setiap sistem file	25.000
Jumlah target pemasangan untuk setiap sistem file di Availability Zone	1
Jumlah target pemasangan untuk setiap cloud pribadi virtual (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 EFS file dari yang lain Akun AWS atau VPC](#).

Sistem file One Zone — Total default Elastic dan Provisioned throughput per sistem file untuk semua klien yang terhubung di masing-masing Wilayah AWS

Wilayah AWS	Throughput baca maksimum	Throughput penulisan maksimum (throughput terukur)
Semua Wilayah AWS	3 GiBps	1 GiBps

Sistem file Regional dan Satu Zona — Total throughput Bursting per sistem file untuk semua klien yang terhubung di masing-masing Wilayah AWS

Wilayah AWS	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 NFS klien 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 EFS klien Amazon (versiamazon-efs-utils ) atau Amazon Driver (). EFS CSI aws-efs-csi-driver Throughput maksimum untuk semua sistem file lainnya adalah 500 MiBps. Untuk informasi selengkapnya tentang kinerja, lihat [Ringkasan kinerja](#). NFS throughput klien dihitung

sebagai jumlah total byte yang dikirim dan diterima, dengan ukuran NFS permintaan 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 menghubungkan ke AmazonEFS, NFS klien yang berada di lokasi atau di tempat lain Wilayah AWS dapat mengamati throughput yang lebih rendah daripada saat menghubungkan ke EFS dari yang sama. Wilayah AWS 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 lokal NFS ke. EFS
- NFSProtokol mendukung maksimum 16 grup IDs (GIDs) per pengguna dan tambahan apa pun GIDs dipotong dari permintaan NFS klien. Untuk informasi selengkapnya, lihat [Akses ditolak ke file yang diizinkan pada sistem NFS file](#).
- Menggunakan Amazon EFS dengan Microsoft Windows tidak didukung.

## Kuota untuk sistem EFS file Amazon

Kuota berikut khusus untuk sistem EFS file Amazon.

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

Sumber Daya	Kuota
* 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, ia mendukung NFSv4 .1 dan NFSv4 .0, kecuali untuk fitur berikut:

- p NFS
- Delegasi klien atau callback jenis apa pun
  - Operasi OPEN selalu kembali OPEN\_DELEGATE\_NONE sebagai tipe delegasi.
  - OPENPengembalian operasi 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

NFSmendukung 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 NFS kesalahan NFS4ERR\_NOTSUPP untuk OPEN perintah apa pun yang menentukan nilai penolakan berbagi selain OPEN4\_SHARE\_DENY\_NONE. NFSKlien Linux tidak menggunakan apa pun selain OPEN4\_SHARE\_DENY\_NONE.

- Daftar kontrol akses (ACLs)
- Amazon EFS tidak memperbarui time\_access atribut pada pembacaan file. EFSPerbaruan Amazon time\_access dalam acara berikut:
  - Saat file dibuat (inode dibuat)
  - Ketika NFS klien membuat panggilan eksplisit setatt

- Pada penulisan ke inode yang disebabkan oleh, misalnya, perubahan ukuran file atau perubahan metadata file
- Setiap atribut inode diperbarui
- Namespace
- Cache balasan persisten
- Keamanan berbasis Kerberos
- NFSv4.1 retensi data
- Set UID pada direktori
- Jenis file yang tidak didukung saat menggunakan CREATE operasi: Blokir perangkat (NF4BLK), perangkat karakter (NF4CHR), direktori atribut (NF4ATTRDIR), dan atribut bernama (NF4NAMEDATTR).
- Atribut yang tidak didukung: FATTR4 FATTR4 \_ ARCHIVEAVAIL, FILES \_ FATTR4 \_, FILES \_ FATTR4 \_ FREE, FILES \_ \_ TOTAL, FATTR4 \_ FS \_ LOCATIONS, FATTR4 \_ MIMETYPE, FATTR4 \_ QUOTA \_ AVAIL \_ \_ FATTR4 \_ HARD, QUOTA \_ AVAIL \_ \_ SOFT, FATTR4 QUOTA \_ \_ USED, dan FATTR4 TIME \_ BACKUP. FATTR4 ACL

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 EFS file Amazon, lihat [Referensi Umum AWS](#).
- Amazon EFS tidak mendukung opsi nconnect pemasangan.
- Anda dapat memasang sistem EFS file Amazon dari server pusat data lokal menggunakan AWS Direct Connect danVPN. Untuk informasi selengkapnya, lihat [Tutorial: Pemasangan dengan klien lokal](#).

# Memecahkan masalah kesalahan operasi file yang terkait dengan kuota

Ketika Anda mengakses sistem EFS file, batas-batas tertentu pada file dalam sistem file berlaku. Melebihi batas ini menyebabkan kesalahan operasi file. Untuk informasi selengkapnya tentang batas berbasis file di AmazonEFS, lihat. [EFSKuota Amazon](#)

Berikut ini, Anda dapat menemukan beberapa kesalahan operasi file umum dan batas yang terkait dengan setiap kesalahan.

## Topik

- [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”](#)

## Perintah gagal dengan kesalahan “Kuota disk terlampaui”

Amazon saat ini EFS 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 lampau, dan kemudian membuat perubahan untuk memenuhi batas itu. Untuk informasi selengkapnya, lihat [Kuota untuk klien NFS](#).



## 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 EFS file Amazon Anda terbuka secara bersamaan pada instance Anda.

- 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 antarmuka I/O file 32-bit, dan menggunakan nomor inode 64-bit. EFS 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 EFS inode 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.

# API Amazon EFS

API Amazon EFS adalah protokol jaringan berdasarkan [HTTP \(RFC 2616\)](#). Untuk setiap panggilan API, Anda membuat permintaan HTTP ke titik akhir EFS khusus wilayah untuk Wilayah AWS 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 tetap operasi dan sintaks untuk setiap operasi diketahui klien tanpa interaksi sebelumnya. Di bagian berikutnya, Anda dapat menemukan deskripsi dari setiap operasi API menggunakan notasi RPC abstrak. Masing-masing memiliki nama operasi yang tidak muncul di kawat. Untuk setiap operasi, topik menentukan pemetaan untuk elemen permintaan HTTP.

Operasi Amazon EFS di mana peta permintaan yang diberikan ditentukan oleh kombinasi metode permintaan ini (GET, PUT, POST, atau DELETE) dan di mana berbagai pola yang sesuai dengan Request-URI-nya. Jika operasi PUT atau POST, Amazon EFS mengekstrak argumen panggilan dari segmen jalur Request-URI, parameter kueri, dan objek JSON di isi permintaan.

## Note

Meskipun nama-nama operasi `CreateFileSystem`, seperti, tidak muncul pada kabel, 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 nama perintah dalam alat baris perintah dan elemen dari API AWS SDK. Misalnya, ada AWS CLI perintah bernama `create-file-system` yang memetakan ke `CreateFileSystem` operasi. Nama operasi juga muncul di AWS CloudTrail log untuk panggilan API Amazon EFS.

## Titik akhir API REST

Titik akhir API adalah nama DNS yang digunakan sebagai host di HTTP URI untuk panggilan API. Titik akhir API ini khusus untuk Wilayah AWS dan berbentuk sebagai berikut.

```
elasticfilesystem.aws-region.amazonaws.com
```

Misalnya, titik akhir API Amazon EFS untuk Wilayah US West (Oregon) adalah sebagai berikut.

```
elasticfilesystem.us-west-2.amazonaws.com
```

Untuk daftar yang Wilayah AWS didukung Amazon EFS (tempat Anda dapat membuat dan mengelola sistem file), lihat [Amazon Elastic File System](#) di Referensi Umum AWS.

Titik akhir API khusus Wilayah menentukan cakupan sumber daya Amazon EFS yang dapat diakses saat Anda membuat panggilan API. Misalnya, saat Anda memanggil `DescribeFileSystems` operasi menggunakan titik akhir sebelumnya, Anda mendapatkan daftar sistem file di Wilayah US West (Oregon) yang telah dibuat di akun Anda.

## Versi API

Versi API yang digunakan untuk panggilan diidentifikasi oleh segmen jalur pertama dari permintaan URI, 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 API Amazon EFS dibatasi untuk masing-masing Akun AWS berdasarkan per-wilayah untuk membantu kinerja layanan. Semua panggilan API Amazon EFS bersama-sama, baik yang berasal dari aplikasi, konsol Amazon EFS, tidak boleh melebihi tingkat permintaan API maksimum yang diizinkan. AWS CLI 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 menggunakan hati-hati ketika polling dan dengan menggunakan retries backoff eksponensial.

## Polling

Aplikasi Anda mungkin perlu memanggil operasi API berulang kali untuk memeriksa pembaruan status. Sebelum Anda memulai polling, berikan waktu permintaan untuk berpotensi menyelesaikan. Saat Anda memulai polling, gunakan interval tidur yang sesuai antara permintaan berturut-turut. Untuk hasil terbaik, gunakan interval tidur yang meningkat.

## Pemrosesan coba ulang atau batch

Aplikasi Anda mungkin perlu mencoba kembali permintaan API setelah gagal, atau untuk memproses beberapa sumber daya (misalnya, semua sistem file Amazon EFS Anda). Untuk merendahkan tingkat permintaan API, gunakan interval tidur yang sesuai antara permintaan berturut-turut. 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 dari algoritme ini, lihat [Pengulang Kesalahan dan Backoff Eksponensial AWS di dalam 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 EFS akses. Titik akses adalah tampilan khusus aplikasi ke dalam sistem EFS file yang menerapkan pengguna dan grup sistem operasi, dan jalur sistem file, ke permintaan sistem file apa pun yang dibuat melalui titik akses. Pengguna dan grup sistem operasi mengesampingkan informasi identitas apa pun yang diberikan oleh NFS klien. 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. Untuk mempelajari lebih lanjut, lihat [Memasang sistem file menggunakan titik EFS akses](#).

### Note

Jika beberapa permintaan untuk membuat titik akses pada sistem file yang sama dikirim secara berurutan, dan sistem file mendekati batas 1.000 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 lakukan 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](#).

## Sintaks Permintaan

```
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"
    }
  ]
}
```

## URIParameter Permintaan

Permintaan tidak menggunakan URI parameter apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam JSON format.

### ClientToken

Serangkaian hingga 64 ASCII karakter yang EFS digunakan Amazon untuk memastikan pembuatan idempoten.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

Wajib: Ya

### FileSystemId

ID dari sistem EFS file yang menyediakan akses ke titik akses.

Jenis: 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 EFS file yang diekspos oleh titik akses sebagai direktori root sistem file Anda ke NFS klien menggunakan titik akses. Klien yang menggunakan jalur akses hanya dapat mengakses direktori root dan di bawahnya. Jika `RootDirectory >` yang `Path` ditentukan tidak ada, EFS Amazon 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`, `MilikGID`, dan izin 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.

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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### [AccessPointArn](#)

Nama Sumber Daya Amazon (ARN) unik yang terkait dengan titik akses.

Jenis: 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 AmazonEFS.

Jenis: 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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

### [FileSystemId](#)

ID sistem EFS file tempat titik akses berlaku.

Jenis: 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.

Jenis: String

Nilai yang Valid: `creating | available | updating | deleting | deleted | error`

## Name

Nama titik akses. Ini adalah nilai Name tag.

Jenis: String

## OwnerId

Mengidentifikasi Akun AWS yang memiliki sumber daya titik akses.

Jenis: String

Kendala Panjang: Panjang maksimum 14.

Pola:  $^(\d{12})|(\d{4}-\d{4}-\d{4})\$$

## PosixUser

POSIXIdentitas lengkap, termasuk ID pengguna, ID grup, dan grup sekunder IDs pada titik akses yang digunakan untuk semua operasi file oleh NFS klien menggunakan titik akses.

Tipe: Objek [PosixUser](#)

## RootDirectory

Direktori pada sistem EFS file yang diekspos oleh titik akses sebagai direktori root ke NFS klien 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.

HTTPKode Status: 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>

HTTPKode Status: 403

## BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

## FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

## IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

HTTPKode Status: 409

## InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## ThrottlingException

Dikembalikan ketika `CreateAccessPoint` API tindakan dipanggil terlalu cepat dan jumlah Titik Akses pada sistem file mendekati [batas 120](#).

HTTPKode Status: 429

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDKuntuk .NET](#)
- [AWS SDKuntuk C ++](#)
- [AWS SDKuntuk Go v2](#)
- [AWS SDKuntuk Java V2](#)
- [AWS SDKuntuk JavaScript V3](#)
- [AWS SDKuntuk PHP V3](#)
- [AWS SDKuntuk Python](#)
- [AWS SDKpara Ruby V3](#)

## CreateFileSystem

Membuat sistem file baru yang kosong. Operasi ini memerlukan token pembuatan dalam permintaan yang EFS digunakan Amazon 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 EFS ditetapkan Amazon, 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 token pembuatan yang dibuat UUID secara acak.

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 EFS Pengguna Amazon.

### 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 EFS kinerja Amazon](#).

Anda dapat mengatur mode throughput untuk sistem file menggunakan ThroughputMode parameter.

Setelah sistem file sepenuhnya dibuat, Amazon EFS menetapkan status siklus hidupnya available, di mana Anda dapat membuat satu atau beberapa target mount untuk sistem file di sistem file Anda. VPC Untuk informasi selengkapnya, lihat [CreateMountTarget](#). Anda memasang sistem EFS file Amazon Anda pada EC2 instance di Anda VPC dengan menggunakan target mount. Untuk informasi selengkapnya, lihat [AmazonEFS: Cara Kerjanya](#).

Operasi ini memerlukan izin untuk tindakan elasticfilesystem:CreateFileSystem.

Sistem file dapat ditandai pada pembuatan. Jika tag ditentukan dalam tindakan pembuatan, IAM lakukan 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](#).

## Sintaks Permintaan

```
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"  
}
```

## URIParameter Permintaan

Permintaan tidak menggunakan URI parameter apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam JSON format.

### 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 [jenis sistem EFS file](#) di Panduan EFS Pengguna Amazon.

#### Note

Sistem file One Zone tidak tersedia di semua Availability Zone di Wilayah AWS mana Amazon EFS tersedia.

Jenis: 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 EFS Panduan Pengguna Amazon.

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

String hingga 64 ASCII karakter. Amazon EFS menggunakan ini untuk memastikan kreasi idempoten.

Jenis: 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 AWS Key Management Service kunci (kunci) KMS yang ada. Jika Anda tidak menentukan KMS kunci, maka KMS kunci default untuk AmazonEFS, `/aws/elasticfilesystem`, digunakan untuk melindungi sistem file terenkripsi.

Tipe: Boolean

Wajib: Tidak

### KmsKeyId

ID KMS kunci yang ingin Anda gunakan untuk melindungi sistem file terenkripsi. Parameter ini diperlukan hanya jika Anda ingin menggunakan KMS kunci non-default. Jika parameter ini tidak ditentukan, KMS kunci default untuk Amazon EFS digunakan. Anda dapat menentukan ID KMS kunci menggunakan format berikut:

- ID kunci - Pengidentifikasi unik dari kunci, misalnya `1234abcd-12ab-34cd-56ef-1234567890ab`.

- ARN- Nama Sumber Daya Amazon (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 - Sebuah 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 simetris KMS. Anda tidak dapat menggunakan KMS kunci asimetris dengan sistem EFS file Amazon.

Jenis: 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`.

Jenis: 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 Support. Untuk informasi selengkapnya, lihat [EFSKuota Amazon yang dapat Anda tingkatkan](#) di Panduan EFS Pengguna Amazon.

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.

Jenis: 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 respons HTTP 201.

Data berikut dikembalikan dalam JSON format oleh layanan.

### 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 Wilayah AWS us-east-1, dan memiliki lokasi yang sama di setiap. Akun AWS

Jenis: 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 EFS penyimpanan](#) di Panduan EFS Pengguna Amazon.

Jenis: 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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

## Encrypted

Nilai Boolean yang, jika benar, menunjukkan bahwa sistem file dienkripsi.

Jenis: Boolean

## FileSystemArn

Amazon Resource Name (ARN) untuk sistem EFS file, 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`

Jenis: String

## FileSystemId

ID sistem file, yang ditetapkan oleh AmazonEFS.

Jenis: 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.

Jenis: 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.

Jenis: 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.

Jenis: 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](#).

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 0.

## OwnerId

Akun AWS Yang menciptakan sistem file.

Jenis: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

## PerformanceMode

Mode performa sistem file.

Jenis: 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 bidangnya `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 EFS Pengguna Amazon.

Jenis: 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.

HTTPKode Status: 400

#### FileSystemAlreadyExists

Dikembalikan jika sistem file yang Anda coba buat sudah ada, dengan token pembuatan yang Anda berikan.

HTTPKode Status: 409

#### FileSystemLimitExceeded

Dikembalikan jika Akun AWS telah membuat jumlah maksimum sistem file yang diizinkan per akun.

HTTPKode Status: 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 Terprovsioned. Coba lagi nanti.

HTTPKode Status: 503

#### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

#### ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 MiB/s telah tercapai.

HTTPKode Status: 400

#### UnsupportedAvailabilityZone

Dikembalikan jika EFS fungsionalitas Amazon yang diminta tidak tersedia di Availability Zone yang ditentukan.

HTTPKode Status: 400

## Contoh

### Buat sistem file terenkripsi EFS

Contoh berikut mengirimkan POST permintaan 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 EFS file terenkripsi dengan ketersediaan One Zone

Contoh berikut mengirimkan POST permintaan 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## CreateMountTarget

Membuat target mount untuk sistem file. Anda kemudian dapat me-mount sistem file pada EC2 instance dengan menggunakan target mount.

Anda dapat membuat satu target pemasangan di setiap Availability Zone di VPC. Semua EC2 instance di dalam Availability Zone tertentu berbagi target mount tunggal untuk sistem file tertentu. VPC Jika Anda memiliki beberapa subnet di Availability Zone, Anda membuat target mount di salah satu subnet. EC2 instance tidak perlu berada di subnet yang sama dengan target mount untuk mengakses sistem file mereka.

Anda hanya dapat membuat satu target mount untuk sistem file One Zone. Anda harus membuat target mount itu di Availability Zone yang sama di mana sistem file berada. Gunakan `AvailabilityZoneName` dan `AvailabilityZoneId` properti di objek [DescribeFileSystems](#) respons untuk mendapatkan informasi ini. Gunakan `subnetId` terkait dengan Availability Zone sistem file saat membuat target mount.

Untuk informasi selengkapnya, lihat [AmazonEFS: Cara Kerjanya](#).

Untuk membuat target mount untuk sistem file, status siklus hidup sistem file harus `available`. Untuk informasi selengkapnya, lihat [DescribeFileSystems](#).

Dalam permintaan, berikan yang berikut:

- ID sistem file tempat Anda membuat target mount.
- ID subnet, yang menentukan hal berikut:
  - VPC di mana Amazon EFS membuat target pemasangan
  - Availability Zone di mana Amazon EFS membuat target mount
  - Rentang alamat IP dari mana Amazon EFS memilih alamat IP dari target mount (jika Anda tidak menentukan alamat IP dalam permintaan)

Setelah membuat target pemasangan, Amazon EFS mengembalikan respons yang menyertakan `MountTargetId` dan `IpAddress`. Anda menggunakan alamat IP ini saat memasang sistem file dalam sebuah EC2 instance. Anda juga dapat menggunakan DNS nama target mount saat memasang sistem file. EC2 instance tempat Anda memasang sistem file dengan menggunakan target mount dapat menyelesaikan DNS nama target mount ke alamat IP-nya. Untuk informasi selengkapnya, lihat [Cara Kerjanya: Ikhtisar Implementasi](#).

Perhatikan bahwa Anda dapat membuat target mount untuk sistem file hanya dalam satu VPC, dan hanya ada satu target mount per Availability Zone. Artinya, jika sistem file sudah memiliki satu atau lebih target mount yang dibuat untuknya, subnet yang ditentukan dalam permintaan untuk menambahkan target mount lain harus memenuhi persyaratan berikut:

- Harus milik yang VPC sama dengan subnet dari target mount yang ada
- Tidak boleh berada di Availability Zone yang sama dengan subnet dari target mount yang ada

Jika permintaan memenuhi persyaratan, Amazon EFS melakukan hal berikut:

- Membuat target mount baru di subnet yang ditentukan.
- Juga membuat antarmuka jaringan baru di subnet sebagai berikut:
  - Jika permintaan menyediakan `IpAddress`, Amazon EFS menetapkan alamat IP tersebut ke antarmuka jaringan. Jika tidak, Amazon EFS memberikan alamat gratis di subnet (dengan cara yang sama seperti `EC2 CreateNetworkInterface` panggilan Amazon ketika permintaan tidak menentukan alamat IP pribadi utama).
  - Jika permintaan menyediakan `SecurityGroups`, antarmuka jaringan ini dikaitkan dengan grup keamanan tersebut. Jika tidak, itu milik grup keamanan default untuk subnet. VPC
  - Menetapkan deskripsi di Mount target `fsmt-id` for file system `fs-id` `fsmt-id` mana ID target mount, dan `fs-id` merupakan `FileSystemId`
  - Menetapkan `requesterManaged` properti antarmuka jaringan ke `true`, dan `requesterId` nilai ke EFS.

Setiap target EFS pemasangan Amazon memiliki satu antarmuka jaringan yang dikelola pemohon yang EC2 sesuai. Setelah antarmuka jaringan dibuat, Amazon EFS menetapkan `NetworkInterfaceId` bidang dalam deskripsi target mount ke ID antarmuka jaringan, dan `IpAddress` bidang 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.



Kami menyarankan Anda membuat target pemasangan di setiap Availability Zone. Ada pertimbangan biaya untuk menggunakan sistem file di Availability Zone melalui target mount yang dibuat di Availability Zone lain. Untuk informasi selengkapnya, lihat [Amazon EFS](#). Selain itu, dengan selalu menggunakan target mount lokal ke Availability Zone instance, Anda menghilangkan skenario kegagalan sebagian. Jika Availability Zone di mana target mount Anda dibuat turun, maka Anda tidak dapat mengakses sistem file Anda melalui target mount tersebut.

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`

## Sintaks Permintaan

```
POST /2015-02-01/mount-targets HTTP/1.1
Content-type: application/json
```

```
{
  "FileSystemId": "string",
  "IpAddress": "string",
  "SecurityGroups": [ "string" ],
  "SubnetId": "string"
}
```

## URIParameter Permintaan

Permintaan tidak menggunakan URI parameter apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam JSON format.

### [FileSystemId](#)

ID sistem file yang akan digunakan untuk membuat target pemasangan.

Jenis: 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](#)

IPv4Alamat yang valid dalam rentang alamat subnet yang ditentukan.

Jenis: 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

### [SecurityGroups](#)

Hingga lima kelompok VPC keamananIDs, dari formulirsg-xxxxxxx. Ini harus sama dengan subnet VPC yang ditentukan.

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 diambahkan target pemasangan. Untuk sistem file One Zone, gunakan subnet yang terkait dengan Availability Zone sistem file.

Jenis: 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",
  "LifecycleState": "string",
  "MountTargetId": "string",
  "NetworkInterfaceId": "string",
  "OwnerId": "string",
  "SubnetId": "string",
  "VpcId": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Data berikut dikembalikan dalam JSON format oleh layanan.

### AvailabilityZoneId

Pengenal 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

Jenis: 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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

### [FileSystemId](#)

ID sistem file tempat target mount dimaksudkan.

Jenis: 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](#)

Alamat di mana sistem file dapat dipasang dengan menggunakan target mount.

Jenis: 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}$`

### [LifecycleState](#)

Status siklus hidup target pemasangan.

Jenis: String

Nilai yang Valid: `creating | available | updating | deleting | deleted | error`

### [MountTargetId](#)

ID target pemasangan yang ditetapkan sistem.

Jenis: String

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsmt-[0-9a-f]{8,40}$`

### [NetworkInterfaceId](#)

ID antarmuka jaringan yang EFS dibuat Amazon saat membuat target pemasangan.

Jenis: String

### OwnerId

Akun AWS ID yang memiliki sumber daya.

Jenis: String

Kendala Panjang: Panjang maksimum 14.

Pola:  $^(\backslash d\{12\}) | (\backslash d\{4\} - \backslash d\{4\} - \backslash d\{4\})\$$

### SubnetId

ID subnet target mount.

Jenis: String

Kendala Panjang: Panjang minimum 15. Panjang maksimum 47.

Pola:  $^subnet - [0-9a-f]\{8, 40\}\$$

### VpcId

ID cloud pribadi virtual (VPC) tempat target mount dikonfigurasi.

Jenis: String

## Kesalahan

### AvailabilityZonesMismatch

Dikembalikan jika Availability Zone yang ditetapkan untuk target mount berbeda dari Availability Zone yang ditentukan untuk penyimpanan One Zone. Untuk informasi selengkapnya, lihat [Redundansi penyimpanan Regional dan Satu Zona](#).

HTTPKode Status: 400

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

## FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

## IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

HTTPKode Status: 409

## InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## IpAddressInUse

Dikembalikan jika permintaan ditentukan `IpAddress` yang sudah digunakan di subnet.

HTTPKode Status: 409

## MountTargetConflict

Dikembalikan jika target pemasangan akan melanggar salah satu batasan yang ditentukan berdasarkan target pemasangan sistem file yang ada.

HTTPKode Status: 409

## NetworkInterfaceLimitExceeded

Akun panggilan telah mencapai batas untuk antarmuka jaringan elastis untuk spesifik Wilayah AWS. Hapus beberapa antarmuka jaringan atau minta kuota akun dinaikkan. Untuk informasi selengkapnya, lihat [VPCKuota Amazon](#) di Panduan VPC Pengguna Amazon (lihat entri Antarmuka jaringan per Wilayah di tabel Antarmuka jaringan).

HTTPKode Status: 409

## NoFreeAddressesInSubnet

Dikembalikan jika `IpAddress` tidak ditentukan dalam permintaan dan tidak ada alamat IP gratis di subnet.

HTTPKode Status: 409

## SecurityGroupLimitExceeded

Dikembalikan jika ukuran SecurityGroups yang ditentukan dalam permintaan lebih besar dari lima.

HTTPKode Status: 400

## SecurityGroupNotFound

Dikembalikan jika salah satu grup keamanan yang ditentukan tidak ada di cloud pribadi virtual subnet (VPC).

HTTPKode Status: 400

## SubnetNotFound

Dikembalikan jika tidak ada subnet dengan ID SubnetId yang disediakan dalam permintaan.

HTTPKode Status: 400

## UnsupportedAvailabilityZone

Dikembalikan jika EFS fungsionalitas Amazon yang diminta tidak tersedia di Availability Zone yang ditentukan.

HTTPKode Status: 400

## Contoh

Tambahkan target mount ke sistem file

Permintaan berikut membuat target mount untuk sistem file. Permintaan menentukan nilai hanya untuk yang diperlukan FileSystemId dan SubnetId parameter. Permintaan tidak memberikan opsional IpAddress dan SecurityGroups parameter. Untuk IpAddress, operasi menggunakan salah satu alamat IP yang tersedia di subnet yang ditentukan. Dan, operasi menggunakan grup keamanan default yang terkait dengan VPC for theSecurityGroups.

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
```

```
{  
  "MountTargetId": "fsmt-55a4413c",  
  "NetworkInterfaceId": "eni-01234567",  
  "FileSystemId": "fs-01234567",  
  "LifecycleState": "available",  
  "SubnetId": "subnet-01234567",  
  "OwnerId": "231243201240",  
  "IpAddress": "172.31.22.183"  
}
```

## Tambahkan target mount ke sistem file

Permintaan berikut menentukan semua parameter permintaan untuk membuat 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"
}
```

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## CreateReplicationConfiguration

Membuat konfigurasi replikasi yang mereplikasi sistem file yang ada ke sistem EFS file read-only yang baru. Untuk informasi selengkapnya, lihat [EFSreplikasi Amazon](#) di Panduan EFS Pengguna Amazon. Konfigurasi replikasi menentukan hal berikut:

- Sistem file sumber — Sistem EFS file yang ingin direplikasi. Sistem file sumber tidak dapat menjadi sistem file tujuan dalam konfigurasi replikasi yang ada.
- Wilayah AWS — Wilayah AWS Di mana sistem file tujuan dibuat. EFSReplikasi Amazon tersedia Wilayah AWS di semua yang EFS tersedia. Wilayah harus diaktifkan. Untuk informasi selengkapnya, lihat [Mengelola Wilayah AWS](#) dalam Panduan Referensi Referensi AWS Umum.
- Konfigurasi sistem file tujuan — Konfigurasi sistem file tujuan tempat sistem file sumber akan direplikasi. Hanya ada satu sistem file tujuan dalam konfigurasi replikasi.

Parameter untuk konfigurasi replikasi meliputi:

- File system ID — ID dari sistem file tujuan untuk replikasi. Jika tidak ada ID yang disediakan, maka EFS buat sistem file baru dengan pengaturan default. Untuk sistem file yang ada, perlindungan penyimpanan replikasi sistem file harus dinonaktifkan. Untuk informasi selengkapnya, lihat [Mereplikasi ke sistem file yang ada](#).
- Availability Zone - Jika Anda ingin sistem file tujuan menggunakan penyimpanan One Zone, Anda harus menentukan Availability Zone untuk membuat sistem file di. Untuk informasi selengkapnya, lihat [jenis sistem EFS file](#) di Panduan EFS Pengguna Amazon.
- Enkripsi — Semua sistem file tujuan dibuat dengan enkripsi saat istirahat diaktifkan. Anda dapat menentukan AWS Key Management Service (AWS KMS) kunci yang digunakan untuk mengenkripsi sistem file tujuan. Jika Anda tidak menentukan KMS kunci, kunci yang dikelola layanan KMS untuk Amazon akan EFS digunakan.

### Note

Setelah sistem file dibuat, Anda tidak dapat mengubah KMS kuncinya.

Untuk sistem file tujuan baru, properti berikut ditetapkan secara default:

- Mode kinerja - Mode kinerja sistem file tujuan cocok dengan sistem file sumber, kecuali sistem file tujuan menggunakan penyimpanan EFS One Zone. Dalam hal ini, mode kinerja 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 memodifikasi mode throughput.
- manajemen siklus hidup - manajemen siklus hidup tidak diaktifkan pada sistem file tujuan. Setelah sistem file tujuan dibuat, Anda dapat mengaktifkan manajemen siklus hidup.
- Pencadangan otomatis — Pencadangan harian otomatis diaktifkan pada sistem file tujuan. Setelah sistem file dibuat, Anda dapat mengubah pengaturan ini.

Untuk informasi selengkapnya, lihat [EFSreplikasi Amazon](#) di Panduan EFS Pengguna Amazon.

## Sintaks Permintaan

```
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"
    }
  ]
}
```

## URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

### [SourceFileSystemId](#)

Menentukan sistem EFS file Amazon 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 JSON format.

### 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,
      "Region": "string",
      "Status": "string"
    }
  ],
  "OriginalSourceFileSystemArn": "string",
  "SourceFileSystemArn": "string",
  "SourceFileSystemId": "string",
  "SourceFileSystemRegion": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Data berikut dikembalikan dalam JSON format oleh layanan.

### CreationTime

Menjelaskan kapan konfigurasi replikasi dibuat.

Tipe: Timestamp

### Destinations

Sebuah array objek tujuan. Hanya satu objek tujuan yang didukung.

Tipe: Array objek [Destination](#)

### OriginalSourceFileSystemArn

Amazon Resource Name (ARN) dari sistem EFS file sumber asli dalam konfigurasi replikasi.

Jenis: String

### SourceFileSystemArn

Amazon Resource Name (ARN) dari sistem file sumber saat ini dalam konfigurasi replikasi.

Jenis: String

### SourceFileSystemId

ID dari sistem EFS file Amazon sumber yang sedang direplikasi.

Jenis: String

Batasan Panjang: Panjang maksimum 128.

Pola:  $^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})\$$

### SourceFileSystemRegion

Wilayah AWS Di mana sistem EFS file sumber berada.

Jenis: 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.

HTTPKode Status: 400

### ConflictException

Dikembalikan jika sistem file sumber dalam replikasi dienkripsi tetapi sistem file tujuan tidak dienkripsi.

HTTPKode Status: 409

### FileSystemLimitExceeded

Dikembalikan jika Akun AWS telah membuat jumlah maksimum sistem file yang diizinkan per akun.

HTTPKode Status: 403

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak "tersedia".

HTTPKode Status: 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.

HTTPKode Status: 503

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### ReplicationNotFound

Dikembalikan jika sistem file yang ditentukan tidak memiliki konfigurasi replikasi.

HTTPKode Status: 404

ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 MiB/s telah tercapai.

HTTPKode Status: 400

UnsupportedAvailabilityZone

Dikembalikan jika EFS fungsionalitas Amazon yang diminta tidak tersedia di Availability Zone yang ditentukan.

HTTPKode Status: 400

ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia Wilayah AWS di mana permintaan dibuat.

HTTPKode Status: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## CreateTags

### Note

DEPRECATED- `CreateTags` tidak digunakan lagi dan tidak dipertahankan. Untuk membuat tag untuk EFS sumber daya, gunakan [TagResource](#) API tindakan.

Membuat atau menimpa 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 menimpa 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`.

### Sintaks Permintaan

```
POST /2015-02-01/create-tags/FileSystemId HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [FileSystemId](#)

ID dari sistem file yang tag Anda ingin memodifikasi (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 JSON format.

### [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 HTTP tubuh kosong.

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

## InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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`.

### Sintaks Permintaan

```
DELETE /2015-02-01/access-points/AccessPointId HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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 HTTP tubuh kosong.

## Kesalahan

### AccessPointNotFound

Dikembalikan jika AccessPointId nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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 secara manual menghapus target mount yang dilampirkan ke sistem file sebelum Anda dapat menghapus sistem EFS file. 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 EFS Replikasi. 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`.

## Sintaks Permintaan

```
DELETE /2015-02-01/file-systems/FileSystemId HTTP/1.1
```

## URIParameter Permintaan

Permintaan menggunakan URI parameter 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 HTTP tubuh kosong.

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemInUse

Dikembalikan jika sistem file memiliki target mount.

HTTPKode Status: 409

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

InternalServerError

Kembali jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Contoh

Hapus sistem file

Contoh berikut mengirimkan DELETE permintaan 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)

- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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`.

### Sintaks Permintaan

```
DELETE /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [FileSystemId](#)

Menentukan sistem EFS file untuk menghapus `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 HTTP tubuh kosong.

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

HTTPKode Status: 409

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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 melepas dukungan 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 Anda VPC dengan menggunakan target mount 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 mengembalikan 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`

## Sintaks Permintaan

```
DELETE /2015-02-01/mount-targets/MountTargetId HTTP/1.1
```

## URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

### [MountTargetId](#)

ID target mount untuk dihapus (String).

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsm-t-[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 HTTP tubuh kosong.

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### DependencyTimeout

Layanan habis waktu mencoba memenuhi permintaan, dan klien harus mencoba panggilan lagi.

HTTPKode Status: 504

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil.

Akun AWS

HTTPKode Status: 404

## Contoh

Hapus target pemasangan sistem file

Contoh berikut mengirimkan DELETE permintaan 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DeleteReplicationConfiguration

Menghapus konfigurasi replikasi. Menghapus konfigurasi replikasi mengakhiri proses replikasi. Setelah konfigurasi replikasi dihapus, sistem file tujuan menjadi Writeable dan perlindungan penipaan replikasi diaktifkan kembali. Untuk informasi selengkapnya, lihat [Menghapus konfigurasi replikasi](#).

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DeleteReplicationConfiguration`.

### Sintaks Permintaan

```
DELETE /2015-02-01/file-systems/SourceFileSystemId/replication-configuration HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [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 HTTP tubuh kosong.

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### ReplicationNotFound

Dikembalikan jika sistem file yang ditentukan tidak memiliki konfigurasi replikasi.

HTTPKode Status: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)





## DeleteTags

### Note

DEPRECATED- DeleteTags tidak digunakan lagi dan tidak dipertahankan. Untuk menghapus tag dari EFS sumber daya, gunakan [UntagResource](#) API tindakan.

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 Billing and Cost Management Pengguna.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DeleteTags`.

### Sintaks Permintaan

```
POST /2015-02-01/delete-tags/FileSystemId HTTP/1.1
Content-type: application/json

{
  "TagKeys": [ "string" ]
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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 JSON format.

### 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:  $^(?![aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_\p{.}:/=+\-@]+)\$$

Diperlukan: Ya

## Sintaxis Respons

```
HTTP/1.1 204
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan HTTP tubuh kosong.

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

## InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DescribeAccessPoints

Mengembalikan deskripsi titik EFS akses Amazon tertentu jika `AccessPointId` disediakan. Jika Anda memberikan `EFSFileSystemId`, ia mengembalikan deskripsi semua titik akses untuk sistem file itu. Anda dapat memberikan salah satu `AccessPointId` atau a `FileSystemId` dalam permintaan, tetapi tidak keduanya.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeAccessPoints`.

### Sintaks Permintaan

```
GET /2015-02-01/access-points?  
AccessPointId=AccessPointId&FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken  
HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### AccessPointId

(Opsional) Menentukan titik EFS akses 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 memberikan `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": {
      "OwnerGid": 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### [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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

## Kesalahan

### AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)





## DescribeAccountPreferences

Mengembalikan pengaturan preferensi akun untuk Akun AWS yang terkait dengan pengguna yang membuat permintaan, saat ini Wilayah AWS.

### Sintaks Permintaan

```
GET /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string"
}
```

### URIParameter Permintaan

Permintaan tidak menggunakan URI parameter apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam JSON format.

#### 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.

Jenis: 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### [NextToken](#)

Hadir jika ada lebih banyak catatan daripada yang dikembalikan dalam respons. Anda dapat menggunakan NextToken dalam permintaan berikutnya untuk mengambil deskripsi tambahan.

Jenis: 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 Wilayah AWS.

Tipe: Objek [ResourceIdPreference](#)

## Kesalahan

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DescribeBackupPolicy

Mengembalikan kebijakan cadangan untuk sistem EFS file yang ditentukan.

### Sintaks Permintaan

```
GET /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [FileSystemId](#)

Menentukan sistem EFS file yang untuk mengambil. 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

## [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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### PolicyNotFound

Dikembalikan jika kebijakan sistem file default berlaku untuk sistem EFS file yang ditentukan.

HTTPKode Status: 404

### ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia Wilayah AWS di mana permintaan dibuat.

HTTPKode Status: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DescribeFileSystemPolicy

Mengembalikan FileSystemPolicy untuk sistem EFS file yang ditentukan.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeFileSystemPolicy`.

### Sintaks Permintaan

```
GET /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [FileSystemId](#)

Menentukan sistem EFS file mana untuk 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.



Data berikut dikembalikan dalam JSON format oleh layanan.

### FileSystemId

Menentukan sistem EFS file yang FileSystemPolicy berlaku.

Jenis: 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

JSONDiformat FileSystemPolicy untuk sistem EFS file.

Jenis: 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## PolicyNotFound

Dikembalikan jika kebijakan sistem file default berlaku untuk sistem EFS file yang ditentukan.

HTTPKode Status: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DescribeFileSystems

Mengembalikan deskripsi sistem EFS file Amazon tertentu jika sistem file `CreationToken` atau `FileSystemId` disediakan. Jika tidak, ia mengembalikan deskripsi semua sistem file yang dimiliki oleh pemanggil Akun AWS di titik Wilayah AWS 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 disetel 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`.

### Sintaks Permintaan

```
GET /2015-02-01/file-systems?  
CreationToken=CreationToken&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems  
HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [CreationToken](#)

(Opsional) Membatasi daftar ke sistem file dengan token penciptaan ini (String). Anda menentukan token pembuatan saat membuat sistem EFS file Amazon.

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 paginasi pada 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### FileSystems

Array deskripsi sistem file.

Tipe: Array objek [FileSystemDescription](#)

### Marker

Hadir jika disediakan oleh pemanggil dalam permintaan (String).

Jenis: 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.

Jenis: 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Contoh

Ambil daftar 10 sistem file

Contoh berikut mengirimkan GET permintaan 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DescribeLifecycleConfiguration

Mengembalikan LifecycleConfiguration objek saat ini untuk sistem EFS file Amazon yang ditentukan. 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.

### Sintaks Permintaan

```
GET /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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`

### Sintaks Permintaan

```
GET /2015-02-01/mount-targets?  
AccessPointId=AccessPointId&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems&MountTargetId=MountTargetId  
HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [AccessPointId](#)

(Opsional) ID titik akses yang target mount yang ingin Anda daftarkan. Itu harus dimasukkan dalam permintaan Anda jika `MountTargetId` atau `FileSystemId` 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 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",
    "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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### Marker

Jika permintaan termasuk `Marker`, respon mengembalikan nilai dalam bidang ini.

Jenis: 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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.



Pola: . +

## Kesalahan

### AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil. Akun AWS

HTTPKode Status: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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:DescribeMountTargetSecurityGroupstindakan` pada sistem file target mount.
- `ec2:DescribeNetworkInterfaceAttributeaksi` pada antarmuka jaringan target mount.

### Sintaks Permintaan

```
GET /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### 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.

HTTPKode Status: 400

### IncorrectMountTargetState

Dikembalikan jika target pemasangan tidak dalam keadaan yang benar untuk operasi.

HTTPKode Status: 409

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil.  
Akun AWS

HTTPKode Status: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DescribeReplicationConfigurations

Mengambil konfigurasi replikasi untuk sistem file tertentu. Jika sistem file tidak ditentukan, semua konfigurasi replikasi untuk Akun AWS in Wilayah AWS diambil.

### Sintaks Permintaan

```
GET /2015-02-01/file-systems/replication-configurations?  
FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [FileSystemId](#)

Anda dapat mengambil konfigurasi replikasi untuk sistem file tertentu dengan memberikan ID sistem file-nya.

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` parameter-nya. 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,
          "Region": "string",
          "Status": "string"
        }
      ],
      "OriginalSourceFileSystemArn": "string",
      "SourceFileSystemArn": "string",
      "SourceFileSystemId": "string",
      "SourceFileSystemRegion": "string"
    }
  ]
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Data berikut dikembalikan dalam JSON format oleh layanan.

### NextToken

Anda dapat menggunakan NextToken dari respons sebelumnya dalam permintaan berikutnya untuk mengambil deskripsi tambahan.

Jenis: 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### ReplicationNotFound

Dikembalikan jika sistem file yang ditentukan tidak memiliki konfigurasi replikasi.

HTTPKode Status: 404

### ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia Wilayah AWS di mana permintaan dibuat.

HTTPKode Status: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## DescribeTags

### Note

DEPRECATED- DescribeTags Tindakan ini tidak digunakan lagi dan tidak dipertahankan. Untuk melihat tag yang terkait dengan EFS sumber daya, gunakan ListTagsForResource API tindakan.

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`.

### Sintaks Permintaan

```
GET /2015-02-01/tags/FileSystemId?Marker=Marker&MaxItems=MaxItems HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### Marker

Jika permintaan menyertakan `aMarker`, respons mengembalikan nilai itu di bidang ini.

Jenis: 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.

Jenis: 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)



## ListTagsForResource

Daftar semua tag untuk EFS sumber daya tingkat atas. Anda harus memberikan ID sumber daya yang ingin Anda ambil tag.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:DescribeAccessPoints`.

### Sintaks Permintaan

```
GET /2015-02-01/resource-tags/ResourceId?MaxResults=MaxResults&NextToken=NextToken
HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### [MaxResults](#)

(Opsional) Menentukan jumlah maksimum objek tag untuk kembali dalam respon. Nilai defaultnya 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 EFS sumber daya yang ingin Anda ambil tag untuk. Anda dapat mengambil tag untuk sistem EFS file dan titik akses menggunakan titik API akhir 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### [NextToken](#)

NextToken hadir jika payload respons diberi paginasi. Anda dapat menggunakan NextToken dalam permintaan berikutnya untuk mengambil halaman berikutnya dari deskripsi titik akses.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: .+

### [Tags](#)

Array tag untuk EFS sumber daya yang ditentukan.

Tipe: Array objek [Tag](#)

## Kesalahan

### AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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.

### Sintaks Permintaan

```
PUT /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
Content-type: application/json

{
  "SecurityGroups": [ "string" ]
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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 JSON format.

### SecurityGroups

Sebuah array hingga lima kelompok VPC keamananIDs.

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

## Sintaxis Respons

```
HTTP/1.1 204
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 204 dengan HTTP tubuh kosong.

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### IncorrectMountTargetState

Dikembalikan jika target pemasangan tidak dalam keadaan yang benar untuk operasi.

HTTPKode Status: 409

## InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## MountTargetNotFound

Dikembalikan jika tidak ada target mount dengan ID yang ditentukan ditemukan di pemanggil.  
Akun AWS

HTTPKode Status: 404

## SecurityGroupLimitExceeded

Dikembalikan jika ukuran SecurityGroups yang ditentukan dalam permintaan lebih besar dari lima.

HTTPKode Status: 400

## SecurityGroupNotFound

Dikembalikan jika salah satu grup keamanan yang ditentukan tidak ada di cloud pribadi virtual subnet (VPC).

HTTPKode Status: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)



## PutAccountPreferences

Gunakan operasi ini untuk mengatur preferensi akun saat ini Wilayah AWS untuk menggunakan sumber daya panjang 17 karakter (63 bit) atau 8 karakter pendek (32 bit) IDs untuk sistem EFS file baru dan memasang sumber daya target. Semua sumber daya yang IDs ada tidak terpengaruh oleh perubahan apa pun yang Anda buat. Anda dapat mengatur preferensi ID selama periode keikutsertaan sebagai EFS transisi ke sumber daya yang panjang. IDs Untuk informasi selengkapnya, lihat [Mengelola EFS sumber daya Amazon 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.

## Sintaks Permintaan

```
PUT /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json
```

```
{
  "ResourceIdType": "string"
}
```

## URIParameter Permintaan

Permintaan tidak menggunakan URI parameter apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam JSON format.

### [ResourceIdType](#)

Menentukan preferensi ID EFS sumber daya untuk mengatur untuk pengguna Akun AWS, saat ini Wilayah AWS, 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.

Jenis: 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### ResourceIdPreference

Menjelaskan jenis sumber daya dan preferensi ID-nya untuk pengguna Akun AWS, saat ini Wilayah AWS.

Tipe: Objek ResourceIdPreference

## Kesalahan

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## PutBackupPolicy

Memperbarui kebijakan cadangan sistem file. Gunakan tindakan ini untuk memulai atau menghentikan pencadangan otomatis sistem file.

### Sintaks Permintaan

```
PUT /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
Content-type: application/json

{
  "BackupPolicy": {
    "Status": "string"
  }
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### FileSystemId

Menentukan sistem EFS file untuk memperbarui kebijakan cadangan 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 JSON format.

#### 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### [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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak "tersedia".

HTTPKode Status: 409

InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

ValidationException

Dikembalikan jika AWS Backup layanan tidak tersedia Wilayah AWS di mana permintaan dibuat.

HTTPKode Status: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## PutFileSystemPolicy

Menerapkan Amazon EFS `FileSystemPolicy` ke sistem EFS file Amazon. Kebijakan sistem file adalah kebijakan IAM berbasis sumber daya 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 API operasi ini. EFS kebijakan sistem file 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 EFS file default](#).

### Note

EFS kebijakan sistem file memiliki batas 20.000 karakter.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:PutFileSystemPolicy`.

### Sintaks Permintaan

```
PUT /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
Content-type: application/json

{
  "BypassPolicyLockoutSafetyCheck": boolean,
  "Policy": "string"
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### FileSystemId

ID sistem EFS file 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 JSON format.

### [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, IAM prinsipal yang membuat permintaan untuk membuat `PutFileSystemPolicy` permintaan future pada sistem file ini. Setel `BypassPolicyLockoutSafetyCheck` ke `True` hanya ketika Anda bermaksud untuk mencegah IAM prinsipal yang membuat permintaan dari membuat `PutFileSystemPolicy` permintaan berikutnya pada sistem file ini. Nilai default-nya adalah `False`.

Tipe: Boolean

Wajib: Tidak

### [Policy](#)

`FileSystemPolicy` yang Anda ciptakan. Menerima definisi kebijakan yang JSON diformat. EFS kebijakan sistem file memiliki batas 20.000 karakter. Untuk mengetahui selengkapnya tentang elemen yang membentuk kebijakan sistem file, lihat [Kebijakan berbasis sumber daya](#) di Amazon. EFS

Jenis: 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### FileSystemId

Menentukan sistem EFS file yang FileSystemPolicy berlaku.

Jenis: 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

JSONDiformat FileSystemPolicy untuk sistem EFS file.

Jenis: 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika FileSystemId nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

## IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

HTTPKode Status: 409

## InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 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.

HTTPKode Status: 400

## Contoh

### Buat sebuah EFS `FileSystemPolicy`

Permintaan berikut membuat `FileSystemPolicy` yang memungkinkan semua AWS prinsipal untuk me-mount sistem EFS file 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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 EFS file Amazon 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. `PutLifecycleConfiguration` panggilan 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.

## Sintaks Permintaan

```
PUT /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
Content-type: application/json
```

```
{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "string",
      "TransitionToIA": "string",
      "TransitionToPrimaryStorageClass": "string"
    }
  ]
}
```

## URIParameter Permintaan

Permintaan menggunakan URI parameter 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 JSON format.

### [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 `put-lifecycle-configuration` CLI perintah atau `PutLifecycleConfiguration` API tindakan, 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### 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.

HTTPKode Status: 400



## FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

## IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

HTTPKode Status: 409

## InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 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 EFS penyimpanan](#) 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"
    }
  ]
}
```

put-lifecycle-configuration CLI Permintaan contoh

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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## TagResource

Membuat tag untuk sumber EFS daya. Anda dapat membuat tag untuk sistem EFS file dan titik akses menggunakan API operasi ini.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:TagResource`.

### Sintaks Permintaan

```
POST /2015-02-01/resource-tags/ResourceId HTTP/1.1
Content-type: application/json
```

```
{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### ResourceId

ID yang menentukan EFS sumber daya 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 JSON format.

#### 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 HTTP tubuh kosong.

## Kesalahan

### AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)



## UntagResource

Menghapus tag dari sumber EFS daya. Anda dapat menghapus tag dari sistem EFS file dan titik akses menggunakan API operasi ini.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:UntagResource`.

### Sintaks Permintaan

```
DELETE /2015-02-01/resource-tags/ResourceId?tagKeys=TagKeys HTTP/1.1
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter berikut.

#### ResourceId

Menentukan EFS sumber daya 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 kunci-nilai yang ingin Anda hapus dari sumber daya yang ditentukan EFS.

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 HTTP tubuh kosong.

## Kesalahan

### AccessPointNotFound

Dikembalikan jika `AccessPointId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### BadRequest

Dikembalikan jika permintaan salah bentuk atau berisi kesalahan seperti nilai parameter yang tidak valid atau parameter wajib yang hilang.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## UpdateFileSystem

Memperbarui mode throughput atau jumlah throughput yang disediakan dari sistem file yang ada.

### Sintaks Permintaan

```
PUT /2015-02-01/file-systems/FileSystemId HTTP/1.1
Content-type: application/json

{
  "ProvisionedThroughputInMibps": number,
  "ThroughputMode": "string"
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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 JSON format.

#### 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 Support. Untuk informasi selengkapnya, lihat [EFSKuota Amazon yang dapat Anda tingkatkan](#) di Panduan EFS Pengguna Amazon.

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.

Jenis: 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### 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 Wilayah AWS us-east-1, dan memiliki lokasi yang sama di setiap Akun AWS

Jenis: 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 EFS penyimpanan](#) di Panduan EFS Pengguna Amazon.

Jenis: 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.

Jenis: 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

Amazon Resource Name (ARN) untuk sistem EFS file, 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`

Jenis: String

### FileSystemId

ID sistem file, yang ditetapkan oleh AmazonEFS.

Jenis: 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.

Jenis: 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.

Jenis: 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.

Jenis: 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](#).

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 0.

### OwnerId

Akun AWS Yang menciptakan sistem file.

Jenis: String



Kendala Panjang: Panjang maksimum 14.

Pola:  $^{\wedge}(\backslash d\{12\}) | (\backslash d\{4\} - \backslash d\{4\} - \backslash d\{4\}) \$$

### PerformanceMode

Mode performa sistem file.

Jenis: 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 bidangnya `Timestamp`. `Timestamp` nilainya adalah bilangan bulat detik sejak 1970-01-01T 00:00:00 Z.

`SizeInBytes` nilai 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 EFS Pengguna Amazon.

Jenis: 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak “tersedia”.

HTTPKode Status: 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.

HTTPKode Status: 503

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

### ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 MiB/s telah tercapai.

HTTPKode Status: 400

## TooManyRequests

Dikembalikan jika Anda tidak menunggu setidaknya 24 jam sebelum mengubah mode throughput, atau mengurangi nilai Throughput yang Disediakan.

HTTPKode Status: 429

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk Ruby V3](#)

## UpdateFileSystemProtection

Memperbarui perlindungan pada sistem file.

Operasi ini memerlukan izin untuk tindakan `elasticfilesystem:UpdateFileSystemProtection`.

### Sintaks Permintaan

```
PUT /2015-02-01/file-systems/FileSystemId/protection HTTP/1.1
Content-type: application/json

{
  "ReplicationOverwriteProtection": "string"
}
```

### URIParameter Permintaan

Permintaan menggunakan URI parameter 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 JSON format.

#### 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 dengan EFS replikasi.
- **REPLICATING**— Sistem file digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dimodifikasi hanya dengan EFS replikasi.

Jika konfigurasi replikasi dihapus, perlindungan penimpanan replikasi sistem file diaktifkan kembali dan sistem file menjadi dapat ditulis.

Jenis: 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.

Data berikut dikembalikan dalam JSON format oleh layanan.

### [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 dengan EFS replikasi.
- **REPLICATING**— Sistem file digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dimodifikasi hanya dengan EFS replikasi.

Jika konfigurasi replikasi dihapus, perlindungan penyimpanan replikasi sistem file diaktifkan kembali, sistem file menjadi dapat ditulis.

Jenis: 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.

HTTPKode Status: 400

### FileSystemNotFound

Dikembalikan jika `FileSystemId` nilai yang ditentukan tidak ada di pemohon. Akun AWS

HTTPKode Status: 404

### IncorrectFileSystemLifecycleState

Dikembalikan jika status siklus hidup sistem file tidak "tersedia".

HTTPKode Status: 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.

HTTPKode Status: 503

### InternalServerError

Dikembalikan jika terjadi kesalahan di sisi server.

HTTPKode Status: 500

## ReplicationAlreadyExists

Dikembalikan jika sistem file sudah termasuk dalam konfigurasi replikasi. >

HTTPKode Status: 409

## ThroughputLimitExceeded

Dikembalikan jika mode throughput atau jumlah throughput yang disediakan tidak dapat diubah karena batas throughput 1024 MiB/s telah tercapai.

HTTPKode Status: 400

## TooManyRequests

Dikembalikan jika Anda tidak menunggu setidaknya 24 jam sebelum mengubah mode throughput, atau mengurangi nilai Throughput yang Disediakan.

HTTPKode Status: 429

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK untuk .NET](#)
- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk JavaScript V3](#)
- [AWS SDK untuk PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK untuk 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 EFS file.

### Daftar Isi

#### AccessPointArn

Nama Sumber Daya Amazon (ARN) unik yang terkait dengan titik akses.

Jenis: 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 AmazonEFS.

Jenis: 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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

Wajib: Tidak

## FileSystemId

ID sistem EFS file tempat titik akses berlaku.

Jenis: 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.

Jenis: 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.

Jenis: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

Wajib: Tidak

## PosixUser

POSIXIdentitas lengkap, termasuk ID pengguna, ID grup, dan grup sekunder IDs pada titik akses yang digunakan untuk semua operasi file oleh NFS klien menggunakan titik akses.

Tipe: Objek [PosixUser](#)

Wajib: Tidak

## RootDirectory

Direktori pada sistem EFS file yang diekspos oleh titik akses sebagai direktori root ke NFS klien 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C ++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk 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 mengaktifkan backup otomatis untuk sistem file.
- **DISABLED**— Cadangan otomatis dimatikan untuk sistem file.
- **DISABLING**— EFS mematikan backup otomatis untuk sistem file.

Jenis: String

Nilai yang Valid: `ENABLED` | `ENABLING` | `DISABLED` | `DISABLING`

Wajib: Ya

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk 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 buat dengan pengaturan ini ketika 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`, `MilikGID`, dan izin 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.

### 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 POSIX grup 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 POSIX pengguna 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 POSIX izin untuk diterapkan ke `RootDirectory`, dalam format angka oktal yang mewakili bit mode file.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 4.

Pola: `^[0-7]{3,4}$`

Diperlukan: Ya

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

## Destination

Menjelaskan sistem file tujuan dalam konfigurasi replikasi.

### Daftar Isi

#### FileSystemId

ID sistem EFS file Amazon tujuan.

Jenis: 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

Wilayah AWS Di mana sistem file tujuan berada.

Jenis: 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 sistem EFS file tujuan.

- `PausedStatus` terjadi sebagai akibat dari memilih keluar dari wilayah sumber atau tujuan setelah konfigurasi replikasi dibuat. Untuk melanjutkan replikasi untuk sistem file, Anda perlu kembali memilih untuk. Wilayah AWS Untuk informasi selengkapnya, lihat [Mengelola Wilayah AWS](#) dalam Panduan Referensi AWS Umum.
- `ErrorStatus` terjadi ketika sumber atau sistem file tujuan (atau keduanya) dalam keadaan gagal dan tidak dapat dipulihkan. Untuk informasi selengkapnya, lihat [Memantau status replikasi](#) di Panduan EFS Pengguna Amazon. Anda harus menghapus konfigurasi replikasi, dan kemudian mengembalikan cadangan terbaru dari sistem file yang gagal (baik sumber atau tujuan) ke sistem file baru.

Jenis: 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

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)



## DestinationToCreate

Menjelaskan sistem file tujuan baru atau yang sudah ada untuk konfigurasi replikasi.

### Daftar Isi

#### AvailabilityZoneName

Untuk membuat sistem file yang menggunakan penyimpanan One Zone, tentukan nama Availability Zone untuk membuat sistem file tujuan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

Wajib: Tidak

#### FileSystemId

ID sistem file yang akan digunakan untuk tujuan. Replikasi tanpa replikasi sistem file harus dinonaktifkan. Jika Anda tidak memberikan ID, maka EFS buat sistem file baru untuk tujuan replikasi.

Jenis: 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 KMS kunci, Amazon EFS menggunakan KMS kunci default Anda untuk AmazonEFS,/aws/elasticfilesystem. ID ini dapat berupa salah satu dari berikut:

- ID Kunci - Pengidentifikasi unik kunci, misalnya `1234abcd-12ab-34cd-56ef-1234567890ab`.

- ARN- Nama Sumber Daya Amazon (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`

Jenis: 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 Wilayah AWS tempat untuk membuat sistem file tujuan.

Jenis: 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}$`

Diperlukan: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk 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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: .+

Wajib: Ya

#### FileSystemId

ID sistem file, yang ditetapkan oleh AmazonEFS.

Jenis: 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.

Jenis: 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.

Jenis: String

Kendala Panjang: Panjang maksimum 14.

Pola: `^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

Wajib: Ya

### PerformanceMode

Mode performa sistem file.

Jenis: 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.

`Timestamp` Nilainya adalah bilangan bulat detik sejak 1970-01-01T 00:00:00 Z.

`SizeInBytes` Nilai 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 Wilayah AWS 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 EFS penyimpanan](#) di Panduan EFS Pengguna Amazon.

Jenis: 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

Amazon Resource Name (ARN) untuk sistem EFS file, 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.

Jenis: 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: `^[a-zA-Z0-9_./=-@]*$`

Wajib: Tidak

### 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.

Wajib: Tidak

### ThroughputMode

Menampilkan mode throughput sistem file. Untuk informasi selengkapnya, lihat [Mode throughput](#) di Panduan EFS Pengguna Amazon.

Jenis: String

Nilai yang Valid: `bursting` | `provisioned` | `elastic`

Wajib: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C ++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

# FileSystemProtectionDescription

Menjelaskan perlindungan pada sistem file.

## Daftar Isi

### ReplicationOverwriteProtection

Status perlindungan penimpaan replikasi sistem file.

- **ENABLED**— Sistem file tidak dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file dapat ditulis. Perlindungan penimpaan replikasi secara default **ENABLED**.
- **DISABLED**— Sistem file dapat digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya baca dan hanya dapat dimodifikasi dengan EFS replikasi.
- **REPLICATING**— Sistem file digunakan sebagai sistem file tujuan dalam konfigurasi replikasi. Sistem file hanya-baca dan hanya dimodifikasi hanya dengan EFS replikasi.

Jika konfigurasi replikasi dihapus, perlindungan penimpaan replikasi sistem file diaktifkan kembali, sistem file menjadi dapat ditulis.

Jenis: String

Nilai yang Valid: **ENABLED** | **DISABLED** | **REPLICATING**

Wajib: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C ++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk 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 `put-lifecycle-configuration` CLI perintah atau `PutLifecycleConfiguration` API tindakan, 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.

Jenis: 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.

Jenis: 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.

Jenis: String

Nilai yang Valid: AFTER\_1\_ACCESS

Wajib: Tidak

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C ++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

## MountTargetDescription

Memberikan deskripsi target mount.

### Daftar Isi

#### FileSystemId

ID sistem file tempat target mount dimaksudkan.

Jenis: 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.

Jenis: String

Nilai yang Valid: `creating | available | updating | deleting | deleted | error`

Wajib: Ya

#### MountTargetId

ID target pemasangan yang ditetapkan sistem.

Jenis: String

Kendala Panjang: Panjang minimum 13. Panjang maksimum 45.

Pola: `^fsmt-[0-9a-f]{8,40}$`

Wajib: Ya

#### SubnetId

ID subnet target mount.

Jenis: String

Kendala Panjang: Panjang minimum 15. Panjang maksimum 47.

Pola: `^subnet-[0-9a-f]{8,40}$`

Wajib: Ya

#### AvailabilityZoneId

Pengenal 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

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.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `.+`

Wajib: Tidak

#### IpAddress

Alamat di mana sistem file dapat dipasang dengan menggunakan target mount.

Jenis: 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

#### NetworkInterfaceId

ID antarmuka jaringan yang EFS dibuat Amazon saat membuat target pemasangan.

Tipe: String

Wajib: Tidak

OwnerId

Akun AWS ID yang memiliki sumber daya.

Jenis: String

Kendala Panjang: Panjang maksimum 14.

Pola:  $^{\backslash}d\{12\}) | (\backslash}d\{4}-\backslash}d\{4}-\backslash}d\{4})\$$

Wajib: Tidak

VpcId

ID cloud pribadi virtual (VPC) tempat target mount dikonfigurasi.

Tipe: String

Wajib: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C ++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

## PosixUser

POSIX identitas 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 NFS klien menggunakan titik akses.

### Daftar Isi

#### Gid

ID POSIX grup yang 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 POSIX pengguna yang digunakan untuk semua operasi sistem file menggunakan titik akses ini.

Tipe: Long

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Ya

#### SecondaryGids

POSIX kelompok sekunder IDs digunakan untuk semua operasi sistem file menggunakan titik akses ini.

Jenis: 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:



- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

## ReplicationConfigurationDescription

Menjelaskan konfigurasi replikasi untuk sistem file tertentu.

### Daftar Isi

#### CreationTime

Menjelaskan kapan konfigurasi replikasi dibuat.

Tipe: Timestamp

Wajib: Ya

#### Destinations

Sebuah array objek tujuan. Hanya satu objek tujuan yang didukung.

Tipe: Array objek [Destination](#)

Wajib: Ya

#### OriginalSourceFileSystemArn

Amazon Resource Name (ARN) dari sistem EFS file sumber asli dalam konfigurasi replikasi.

Tipe: String

Diperlukan: Ya

#### SourceFileSystemArn

Amazon Resource Name (ARN) dari sistem file sumber saat ini dalam konfigurasi replikasi.

Tipe: String

Diperlukan: Ya

#### SourceFileSystemId

ID dari sistem EFS file Amazon sumber yang sedang direplikasi.

Jenis: 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

Wilayah AWS Di mana sistem EFS file sumber berada.

Jenis: 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}$`

Diperlukan: Ya

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

## ResourceIdPreference

Menjelaskan jenis sumber daya dan preferensi ID-nya untuk pengguna Akun AWS, saat ini Wilayah AWS.

### Daftar Isi

#### ResourceIdType

Mengidentifikasi preferensi ID EFS sumber daya, baik LONG\_ID (17 karakter) atau SHORT\_ID (8 karakter).

Jenis: String

Nilai yang Valid: LONG\_ID | SHORT\_ID

Wajib: Tidak

#### Resources

Mengidentifikasi EFS sumber daya Amazon 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 ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C ++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

## RootDirectory

Menentukan direktori pada sistem EFS file Amazon yang menyediakan akses ke titik akses. Titik akses mengekspos jalur sistem file yang ditentukan selagi direktori root dari sistem file Anda ke aplikasi menggunakan titik akses. NFSklien 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` ditentukan tidak ada, EFS buat direktori root menggunakan `CreationInfo` pengaturan saat 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 EFS file untuk mengekspos sebagai direktori root ke NFS klien menggunakan titik akses untuk mengakses sistem EFS file. Sebuah jalur dapat memiliki hingga empat subdirektori. Jika jalur yang ditentukan tidak ada, Anda diminta untuk menyediakan `CreationInfo`.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 100.

Pola: `^(\\|\\(?!\\.)+[\\$#<>;`|&?{}^*\\/n]+){1,4}$`

Diperlukan: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk 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 :.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola:  $^(?![aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_\cdot:/+=\-\@]+)\$$

Wajib: Ya

#### Value

Nilai dari kunci tag.

Jenis: String

Batasan Panjang: Panjang maksimum 256.

Pola:  $^([\p{L}\p{Z}\p{N}_\cdot:/+=\-\@]*)\$$

Diperlukan: Ya

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan ini API di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK untuk C ++](#)
- [AWS SDK untuk Java V2](#)
- [AWS SDK untuk Ruby V3](#)

## Riwayat dokumen

- APIversi: 2015-02-01
- Pembaruan dokumentasi terbaru: 14 Oktober 2024

Tabel berikut menjelaskan perubahan penting pada Panduan Pengguna Amazon Elastic File System setelah Juli 2018. Untuk pemberitahuan tentang pembaruan dokumentasi, Anda dapat berlangganan RSS umpan.

Perubahan	Deskripsi	Tanggal
<a href="#">Batas throughput elastis meningkat</a>	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 <a href="#">Total throughput Elastis default untuk semua klien yang terhubung di masing-masing Wilayah AWS klien</a> .	Oktober 14, 2024
<a href="#">Memperbarui kebijakan AWS terkelola yang ada</a>	Elemen opsional Sid (ID pernyataan) sekarang disertakan dalam pernyataan AmazonElasticFileS ystemReadOnlyAccess kebijakan. Nilai Sid adalah ElasticFileSystemR eadOnlyAccess . Untuk informasi selengkapnya tentang elemen Sid kebijakan , lihat <a href="#">elemen IAM JSON kebijakan: Sid</a> .	Agustus 7, 2024



<a href="#">Batas throughput elastis meningkat</a>	Batas throughput elastis telah meningkat untuk spesifik Wilayah AWS. Untuk informasi selengkapnya, lihat <a href="#">Total throughput Elastis default untuk semua klien yang terhubung di masing-masing Wilayah AWS klien</a> .	Juli 31, 2024
<a href="#">Peningkatan kuota untuk target mount</a>	Jumlah maksimum target mount untuk setiap virtual private cloud (VPC) meningkat dari 400 menjadi 1.400. Untuk informasi selengkapnya, lihat <a href="#">kuota EFS sumber daya Amazon yang tidak dapat Anda ubah</a> .	15 Mei 2024
<a href="#">Peningkatan batas throughput gabungan untuk sistem file Elastic</a>	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 EFS klien Amazon (amazon-efs-utils versi) atau Amazon EFS CSI Driver (). <a href="#">aws-efs-csi-driver</a> Untuk informasi selengkapnya, lihat tabel ringkasan kinerja di <a href="#">EFSperforma Amazon</a> .	April 30, 2024

<a href="#">Batas throughput elastis meningkat</a>	Batas throughput elastis telah meningkat untuk spesifik Wilayah AWS. Untuk informasi selengkapnya, lihat <a href="#">Total throughput Elastis default untuk semua klien yang terhubung di masing-masing Wilayah AWS klien</a> .	Maret 13, 2024
<a href="#">Meningkat IOPS</a>	Sistem file yang menggunakan an throughput Elastic dapat mendorong maksimum 90.000 pembacaan untuk data yang jarang diakses. Untuk informasi selengkapnya, lihat <a href="#">Ringkasan kinerja</a> .	Januari 22, 2024
<a href="#">Memperbarui kebijakan AWS terkelola yang ada</a>	Izin elasticfilesystem: UpdateFileSystemProtection ditambahkan ke AmazonElasticFileSystemFull Access kebijakan yang ada untuk mengizinkan prinsipal memperbarui perlindungan pada sistem file. Untuk informasi selengkapnya, lihat <a href="#">Amazon EFS memperbarui kebijakan AWS terkelola</a> .	27 November 2023
<a href="#">Replikasi ke sistem file yang ada</a>	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 <a href="#">Sistem file tujuan</a> .	27 November 2023

<a href="#">Perindungan sistem file ditambahkan</a>	Perindungan penimpanan replikasi telah ditambahkan ke sistem file dan diaktifkan secara default. Perindungan mencegah sistem file digunakan sebagai tujuan dalam konfigurasi replikasi. Untuk informasi selengkapnya, lihat <a href="#">Perindungan sistem file</a> .	27 November 2023
<a href="#">Kelas penyimpanan baru, jenis sistem file, dan kebijakan siklus hidup</a>	Amazon EFS sekarang menawarkan kelas penyimpanan EFS Arsip, jenis sistem file, dan kebijakan siklus hidup Transisi ke Arsip. Untuk informasi selengkapnya, lihat <a href="#">Jenis sistem file dan kelas penyimpanan</a> .	26 November 2023
<a href="#">Meningkat IOPS</a>	Sistem file throughput elastis sekarang mendukung maksimum 65.000 operasi baca dan 50.000 tulis IOPS untuk data yang jarang diakses, dan sekarang mendukung 250.000 pembacaan IOPS untuk data yang sering diakses. Untuk informasi selengkapnya, lihat <a href="#">Ringkasan kinerja</a> .	26 November 2023
<a href="#">Hapus konfigurasi replikasi dari sistem file sumber</a>	Konfigurasi replikasi sekarang dapat dihapus dari sistem file sumber. Untuk informasi selengkapnya, lihat <a href="#">Menghapus konfigurasi replikasi</a> .	September 19, 2023

---

<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Wilayah Israel (Tel Aviv).	Agustus 7, 2023
<a href="#">Peningkatan kinerja untuk sistem file mode Tujuan Umum</a>	Sistem file mode Tujuan EFS Umum Amazon sekarang mendukung hingga 55.000 operasi baca per detik dan 25.000 operasi tulis. Untuk informasi selengkapnya, lihat <a href="#">Kuota untuk Sistem EFS File Amazon</a> .	3 Agustus 2023
<a href="#">Batas throughput yang disediakan meningkat</a>	Batas throughput yang disediakan telah meningkat untuk spesifik. Wilayah AWS Untuk informasi selengkapnya, lihat <a href="#">Total throughput standar yang disediakan untuk semua klien yang terhubung di masing-masing klien</a> . Wilayah AWS	Juni 21, 2023
<a href="#">Dukungan Wilayah yang Diperluas untuk EFS replikasi</a>	EFSreplikasi sekarang tersedia Wilayah AWS di semua yang EFS tersedia. Untuk informasi selengkapnya, lihat <a href="#">EFSreplikasi Amazon</a> .	28 April 2023

<a href="#">Peningkatan batas throughput elastis</a>	Batas throughput elastis telah meningkat untuk spesifik Wilayah AWS. Untuk informasi selengkapnya, lihat tabel <a href="#">Total default Elastic throughput untuk semua klien yang terhubung di masing-masing Wilayah AWS klien</a> .	17 April 2023
<a href="#">Elastic menggantikan Bursting sebagai mode throughput default</a>	Mode throughput default (dan direkomendasikan) untuk sistem file sekarang Elastis, bukan Bursting. Untuk informasi selengkapnya, lihat <a href="#">Mode throughput</a> .	13 April 2023
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Wilayah Asia Pasifik (Melbourne).	12 April 2023
<a href="#">Support ditambahkan untuk macOS Ventura</a>	Amazon sekarang EFS dapat diinstal pada instance EC2 Mac yang berjalan di macOS Ventura. Untuk informasi selengkapnya, lihat <a href="#">Distribusi yang didukung</a> .	10 April 2023
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Wilayah Asia Pasifik (Hyderabad).	16 Februari 2023
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Eropa (Spanyol) Wilayah AWS.	19 Januari 2023

<a href="#">Batas titik akses untuk sistem file telah meningkat</a>	Jumlah maksimum titik akses yang dapat dimiliki oleh sistem file tunggal telah meningkat dari 120 menjadi 1.000. Untuk informasi selengkapnya, lihat <a href="#">Kuota sumber daya</a> .	Januari 17, 2023
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Eropa (Zurich). Wilayah AWS	15 Desember 2022
<a href="#">Support ditambahkan untuk kebijakan siklus hidup satu hari</a>	Anda sekarang dapat memilih satu hari untuk kebijakan siklus hidup Transisi ke IA. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan kebijakan Siklus Hidup</a> .	27 November 2022
<a href="#">Mengurangi latensi baca dan tulis</a>	Latensi untuk membaca dan menulis data file telah berkurang untuk penyimpanan One Zone dan sistem file penyimpanan Standar. Untuk informasi selengkapnya, lihat <a href="#">Ringkasan kinerja</a> .	27 November 2022
<a href="#">Mode throughput tambahan ditambahkan</a>	Mode throughput elastis ditambahkan sebagai opsi throughput untuk sistem EFS file Amazon. Untuk informasi lebih lanjut, lihat <a href="#">Throughput elastis</a> .	27 November 2022

---

<a href="#"><u>Wilayah AWS Dukungan tambahan ditambahkan</u></a>	Amazon sekarang EFS tersedia untuk semua pengguna di Wilayah Timur Tengah (UAE).	Oktober 17, 2022
<a href="#"><u>Support ditambahkan untuk EFS Replikasi</u></a>	Amazon EFS telah menghapus batas sebelumnya di mana EFS replikasi tidak mendukung soket dan pipa bernama, atau FIFOs.	15 September 2022
<a href="#"><u>Batas jumlah kunci file per koneksi telah meningkat</u></a>	Jumlah kunci file per koneksi telah meningkat dari 8192 menjadi 65.536. Untuk informasi selengkapnya, lihat <a href="#"><u>Kuota untuk NFS klien</u></a> .	4 Mei, 2022
<a href="#"><u>Batas untuk proses menggunakan kunci file dihapus</u></a>	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 <a href="#"><u>Kuota untuk NFS klien</u></a> .	4 Mei, 2022
<a href="#"><u>Wilayah AWS Dukungan tambahan ditambahkan</u></a>	Amazon sekarang EFS tersedia untuk semua pengguna di Asia Pasifik (Jakarta) Wilayah AWS.	27 Januari 2022

[Support ditambahkan untuk EFS Replikasi](#)

Gunakan EFS Replikasi untuk mereplikasi data dan metadata pada sistem EFS file ke sistem EFS file lain sesuai pilihan Anda. Wilayah AWS 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 EFS file Amazon baru dan sumber daya target mount sekarang ditetapkan 17 karakterIDs. Untuk informasi selengkapnya, lihat [Bekerja dengan EFS sumber daya Amazon](#).

Oktober 22, 2021

[Support ditambahkan untuk EFS Intelligent-Tiering](#)

EFSIntelligent-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 [EFSIntelligent-Tiering](#) dan 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 Wilayah AWS 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](#)

Sekarang Anda dapat menggunakan EFS mount helper untuk memasang sistem EFS file Amazon yang menggunakan kelas penyimpanan One Zone ke EC2 instance yang berada di Availability Zone yang berbeda. Anda dapat menggunakan az opsi baru untuk menentukan Availability Zone dari sistem EFS file Amazon. 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. Wilayah AWS Kelas penyimpanan EFS One Zone dan One Zone-Infrequent Access (One Zone-IA) adalah opsi hemat biaya untuk menyimpan data yang tidak memerlukan ketahanan Multi-AZ dari kelas penyimpanan Standar dan Standar IA. EFS Untuk informasi selengkapnya, lihat [Menggunakan kelas EFS penyimpanan](#).

9 Maret 2021

[Wilayah AWS Dukungan tambahan ditambahkan](#)

Amazon sekarang EFS tersedia untuk semua pengguna di Asia Pasifik (Osaka) Wilayah AWS.

3 Maret 2021

[Support ditambahkan untuk instans Amazon EC2 macOS yang menjalankan macOS Big Sur](#)

Anda sekarang dapat memasang sistem EFS file Amazon Anda dari instance EC2 macOS yang menjalankan macOS Big Sur dengan menggunakan EFS mount helper atau dengan menggunakan perintah mount. NFS Untuk informasi selengkapnya, lihat [Memasang dengan EFS mount helper](#) atau [Memasang sistem file tanpa EFS mount helper](#).

23 Februari 2021

<a href="#">EFSKonsol Amazon baru tersedia di AWS GovCloud (US) Wilayah</a>	EFSKonsol Amazon baru sekarang tersedia di AWS GovCloud (US) Wilayah AWS.	10 Februari 2021
<a href="#">Support ditambahkan untuk EFS CloudWatch metrik Amazon baru MeteredIO Bytes</a>	Anda dapat menggunakan <code>MeteredIOBytes</code> 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 <a href="#">CloudWatchMetrik Amazon untuk Amazon EFS</a> .	28 Januari 2021
<a href="#">Amazon EFS meningkatkan throughput baca sistem file sebesar 300%</a>	Sistem EFS file Amazon sekarang mengukur permintaan baca pada sepertiga tingkat permintaan lainnya.	28 Januari 2021
<a href="#">Support ditambahkan untuk EFS CloudWatch metrik Amazon baru StorageBytes</a>	Anda dapat menggunakan <code>StorageBytes</code> 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 <a href="#">CloudWatch metrik Amazon untuk Amazon EFS</a> .	11 Januari 2021

[Gunakan AWS Transfer Family untuk mengakses sistem EFS file Amazon](#)

Anda dapat menggunakan an AWS Transfer Family untuk mentransfer file masuk dan keluar dari sistem EFS file Amazon Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS Transfer Family untuk mengakses file di sistem EFS file Anda.](#)

6 Januari 2021

[Gunakan AWS Systems Manager untuk mengelola EFS klien Amazon \(amazon-efs-utils \)](#)

Anda dapat menggunakan AWS Systems Manager untuk menginstal atau memperbarui EFS klien Amazon (amazon-efs-utils ) secara otomatis pada EC2 instans Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS Systems Manager untuk menginstal atau memperbarui EFS klien Amazon secara otomatis.](#)

29 September 2020

[Menegakkan pembuatan sistem file terenkripsi EFS](#)

Anda dapat menggunakan kunci kondisi `elasticfilesystem:Encrypted` AWS Identity and Access Management (IAM) untuk memaksakan bahwa pengguna membuat sistem EFS file Amazon yang dienkripsi saat istirahat. Untuk informasi selengkapnya, lihat [Menegakkan Pembuatan Sistem EFS File Amazon yang Dienkripsi saat Istirahat.](#)

16 September 2020

<a href="#">Throughput Amazon EFS per klien meningkat 100%</a>	EFS sekarang mendukung hingga 500MB/s of per-client throughput, a 100% increase from the previous limit of 250 MB/s. Untuk informasi selengkapnya, lihat <a href="#">Kuota untuk sistem EFS file Amazon</a> .	23 Juli 2020
<a href="#">Support ditambahkan untuk backup harian otomatis sistem file Amazon EFS</a>	Pencadangan harian otomatis sekarang diaktifkan secara default saat membuat sistem file menggunakan konsol. EFS Untuk informasi selengkapnya, lihat <a href="#">Menggunakan AWS Backup dengan Amazon EFS</a> .	Juli 16, 2020
<a href="#">Alur kerja Quick Create baru menyederhanakan pembuatan sistem file Amazon EFS</a>	Menggunakan opsi Buat Cepat di EFS konsol, Anda dapat membuat sistem EFS file menggunakan pengaturan yang direkomendasikan layanan dengan satu tombol. Untuk informasi selengkapnya, lihat <a href="#">Sistem EFS file CreateYour Amazon</a> .	Juli 16, 2020
<a href="#">EFSKonsol Amazon baru sekarang tersedia</a>	EFSKonsol baru memudahkan Anda untuk menggunakan Amazon EFS dan menyederhanakan pengelolaan sistem EFS file Anda.	Juli 16, 2020

[Amazon EFS meningkatkan throughput minimum sistem file](#)

Sistem EFS file Amazon 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 EFS Umum Amazon sekarang mendukung hingga 35.000 operasi baca per detik, peningkatan 400% dari batas sebelumnya 7.000. Untuk informasi selengkapnya, lihat [Kuota untuk Sistem EFS File Amazon](#).

1 April 2020

[Wilayah AWS Dukungan tambahan ditambahkan](#)

Amazon sekarang EFS tersedia untuk semua pengguna di Beijing dan Ningxia Wilayah AWS.

22 Januari 2020

[Support ditambahkan untuk IAM otorisasi untuk klien NFS](#)

Anda sekarang dapat menggunakan AWS Identity and Access Management (IAM) untuk mengelola NFS akses ke sistem EFS file Amazon. Untuk informasi selengkapnya, lihat [Menggunakan AWS IAM untuk Mengontrol NFS Akses ke Amazon EFS](#).

13 Januari 2020

[Support ditambahkan untuk EFS Access Points](#)

Titik EFS akses Amazon adalah titik masuk khusus aplikasi ke dalam sistem EFS file Amazon yang memudahkan pengelolaan akses aplikasi ke kumpulan data bersama. Untuk informasi selengkapnya, lihat [Bekerja dengan Titik EFS Akses Amazon](#).

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 IAM peran terkait layanan](#)

Amazon EFS sekarang menggunakan peran terkait layanan berdasarkan IAM, membuatnya lebih mudah untuk diatur EFS dengan menambahkan izin yang diperlukan secara otomatis. Untuk informasi selengkapnya, lihat [Menggunakan Peran Tertaut Layanan untuk Amazon EFS](#).

10 Desember 2019

[Wilayah AWS Dukungan tambahan ditambahkan](#)

Amazon sekarang EFS tersedia untuk semua pengguna di Eropa (Stockholm) Wilayah AWS.

20 November 2019

<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Asia Pasifik (Hong Kong) Wilayah AWS.	20 November 2019
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Amerika Selatan (São Paulo). Wilayah AWS	20 November 2019
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Timur Tengah (Bahrain) Wilayah AWS.	20 November 2019
<a href="#">Kebijakan manajemen Siklus Hidup 7 hari baru ditambahkan</a>	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 Manajemen <a href="#">EFSSiklus Hidup</a> .	6 November 2019
<a href="#">Support ditambahkan untuk VPC Endpoint Antarmuka</a>	Anda dapat membuat koneksi pribadi antara cloud pribadi virtual Anda dan Amazon EFS untuk menelepon EFSAPI. Untuk informasi selengkapnya, lihat <a href="#">Bekerja dengan VPC Titik Akhir</a> .	22 Oktober 2019



[Pasang sistem EFS file saat meluncurkan EC2 instance baru.](#)

Sekarang Anda dapat mengonfigurasi EC2 instans Amazon baru untuk memasang sistem EFS file Anda saat diluncurkan di Launch Instance Wizard. Untuk informasi lebih lanjut, lihat [Langkah 2. Buat EC2 sumber daya Anda dan luncurkan EC2 instans Anda.](#)

17 Oktober 2019

[Support untuk Service Quotas ditambahkan](#)

Anda sekarang dapat melihat semua EFS batas Amazon di konsol Service Quotas. Untuk informasi selengkapnya, lihat [EFSBatas Amazon.](#)

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 Manajemen [EFSSiklus Hidup.](#)

9 Juli 2019

<a href="#">EFSManajemen Siklus Hidup sekarang tersedia di semua sistem EFS file.</a>	Fitur Manajemen EFS Siklus Hidup sekarang tersedia di semua sistem EFS file. Pembatasan sebelumnya berdasarkan kapan sistem file dibuat sekarang dihapus. Untuk informasi selengkapnya, lihat Manajemen <a href="#">EFS Siklus Hidup</a> .	9 Juli 2019
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Eropa (Paris) Wilayah AWS.	12 Juni 2019
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Asia Pasifik (Mumbai) Wilayah AWS.	5 Juni 2019
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Kanada (Tengah) Wilayah AWS.	1 Mei 2019
<a href="#">APIPembaruan: Tag sekarang menjadi bagian dari muatan CreateFileSystem operasi</a>	Anda sekarang dapat menyertakan tag saat menggunakan AWS API dan CLI CreateFileSystem operasi untuk membuat sistem EFS file Amazon. Untuk informasi selengkapnya, lihat <a href="#">CreateFile eSystem</a> dan <a href="#">Membuat Sistem File Menggunakan AWS CLI</a> .	19 Februari 2019

[Fitur baru: Kelas penyimpanan EFS Akses Jarang dan manajemen siklus hidup](#)

Amazon EFS Infrequent Access adalah kelas penyimpanan yang dioptimalkan biaya untuk file yang jarang diakses. EFS manajemen siklus hidup secara otomatis mentransisikan file dari penyimpanan Standar ke Akses Jarang. Untuk informasi selengkapnya, lihat [Kelas EFS Penyimpanan](#).

13 Februari 2019

[Wilayah AWS Dukungan tambahan ditambahkan](#)

Amazon sekarang EFS tersedia untuk semua pengguna di Eropa (London) Wilayah AWS.

23 Januari 2019

[AWS Backup Integrasi layanan dengan Amazon EFS](#)

Sistem EFS file Amazon 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 lebih lanjut, lihat [AWS Backup dan Amazon EFS](#).

16 Januari 2019

[Dukungan koneksi Transit Gateway ke sistem penyimpanan lokal ditambahkan.](#)

Sistem EFS file Amazon sekarang dapat diakses menggunakan koneksi Transit Gateway ke sistem penyimpanan lokal. Untuk informasi selengkapnya, lihat [Memasang dari Akun Lain atau VPC](#) dan [Walkthrough: Pasang Sistem File dari yang Berbeda](#). VPC

6 Desember 2018

[EFSFile 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 VPC peering antar wilayah ditambahkan](#)

Amazon EFS sekarang dapat diakses melalui VPN koneksi dan koneksi VPC peering antar wilayah. Untuk informasi selengkapnya, lihat [Mentransfer File dari Sistem File Lokal ke Amazon EFS Menggunakan AWS DataSync](#).

23 Oktober 2018

<a href="#">VPN dan dukungan koneksi VPC peering antar wilayah ditambahkan</a>	Sistem EFS file Amazon sekarang dapat diakses melalui VPN koneksi dan koneksi VPC peering antar wilayah. Untuk informasi selengkapnya, lihat <a href="#">Memasang dari Akun Lain atau VPC</a> dan <a href="#">Cara EFS Kerja Amazon dengan Direct Connect dan VPNs</a> .	23 Oktober 2018
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Asia Pasifik (Singapura) Wilayah AWS.	13 Juli 2018
<a href="#">Memperkenalkan mode Throughput yang Disediakan</a>	Anda sekarang dapat menyediakan throughput untuk sistem file baru atau yang sudah ada dengan mode Provisioned Throughput yang baru. Untuk informasi selengkapnya, lihat <a href="#">Mode throughput</a> .	12 Juli 2018
<a href="#">Wilayah AWS Dukungan tambahan ditambahkan</a>	Amazon sekarang EFS tersedia untuk semua pengguna di Asia Pasifik (Tokyo) Wilayah AWS.	11 Juli 2018

Tabel berikut menjelaskan perubahan penting pada Panduan Pengguna Amazon Elastic File System sebelum Juli 2018.

Perubahan	Deskripsi	Tanggal Diubah
Wilayah AWS Dukungan	Amazon sekarang EFS tersedia untuk semua pengguna di AWS Wilayah Asia Pasifik (Seoul).	30 Mei 2018

Perubahan	Deskripsi	Tanggal Diubah
tambahan ditambahkan		
Menambahkan 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 <a href="#">Menggunakan matematika metrik dengan CloudWatch metrik</a> .	4 April 2018
Menambahkan amazon-efs-utils set alat sumber terbuka, dan menambahkan enkripsi dalam perjalanan	<p>amazon-efs-utils Alat-alat tersebut adalah seperangkat file executable open-source yang menyederhanakan aspek penggunaan AmazonEFS, seperti pemasangan. Tidak ada biaya tambahan untuk digunakan amazon-efs-utils , dan Anda dapat mengunduh alat-alat ini dari GitHub. Untuk informasi selengkapnya, lihat <a href="#">Menginstal EFS klien Amazon</a>.</p> <p>Juga dalam rilis ini, Amazon EFS sekarang mendukung enkripsi dalam perjalanan melalui terowongan Transport Layer Security (TLS). Untuk informasi selengkapnya, lihat <a href="#">Mengkripsi data di Amazon EFS</a>.</p>	4 April 2018
Batas sistem file yang diperbarui per Wilayah AWS	Amazon EFS telah meningkatkan batas jumlah sistem file untuk semua akun di semua Wilayah AWS s. Untuk informasi selengkapnya, lihat <a href="#">Kuota EFS sumber daya Amazon yang tidak dapat Anda ubah</a> .	15 Maret 2018
Wilayah AWS Dukungan tambahan ditambahkan	Amazon sekarang EFS tersedia untuk semua pengguna di AS Barat (California N.) Wilayah AWS.	14 Maret 2018

Perubahan	Deskripsi	Tanggal Diubah
Enkripsi data saat istirahat	Amazon EFS sekarang mendukung enkripsi data saat istirahat. Untuk informasi selengkapnya, lihat <a href="#">Mengenkripsi data di Amazon EFS</a> .	14 Agustus 2017
Dukungan Wilayah tambahan ditambahkan	Amazon sekarang EFS tersedia untuk semua pengguna di Wilayah Eropa (Frankfurt).	Juli 20, 2017
Nama sistem file menggunakan Sistem Nama Domain (DNS)	Amazon EFS sekarang mendukung DNS nama untuk sistem file. DNSNama sistem file secara otomatis menyelesaikan alamat IP target mount di Availability Zone untuk instance Amazon EC2 yang menghubungkan. Untuk informasi selengkapnya, lihat <a href="#">Memasang di Amazon EC2 dengan DNS nama</a> .	20 Desember 2016
Peningkatan dukungan tag untuk sistem file	Amazon EFS sekarang mendukung 50 tag per sistem file. Untuk informasi selengkapnya tentang tag di AmazonEFS, lihat <a href="#">Sumber daya penandaan EFS</a> .	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 EFS file Amazon yang dapat dibuat per akun untuk masing-masing Wilayah AWS 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.