



用户指南

Amazon Elastic File System



Amazon Elastic File System: 用户指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 Amazon Elastic File System ?	1
您是否是首次接触 Amazon EFS 的用户 ?	2
工作原理	2
如何将 Amazon EFS 与 Amazon EC2 结合使用	4
Amazon EFS 如何 AWS Direct Connect 与 AWS 托管 VPN 配合使用	6
Amazon EFS 是如何使用的 AWS Backup	7
功能	8
身份验证和访问控制	8
Amazon EFS 中的数据一致性	8
可用性与持久性	9
复制	14
开始使用	15
先决条件	15
创建文件系统并启动 EC2 实例	15
将文件传输到文件系统	16
先决条件	16
清理 资源	17
安装 EFS 客户端	18
EFS 工具的依赖项	18
支持的发行版	19
手动安装 EFS 客户端	20
在 Amazon EC2 Linux 实例上安装 Amazon EFS 客户端	21
在其他 Linux 发行版上安装 Amazon EFS 客户端	22
在 EC2 Mac 实例上安装 EFS 客户端	22
自动安装 EFS 客户端	22
Amazon EFS 客户端在安装过程中的作用	23
Systems Manager 支持的操作系统	23
配置 AWS Systems Manager 安装 EFS 客户端	24
安装和升级 botocore	26
升级 stunnel	26
解决安装 stunnel 时遇到的问题	28
启用 FIPS 模式	29
创建和管理资源	31
实现摘要	32

资源 IDs	33
创建令牌和幂等性	33
创建文件系统	34
创建文件系统所需的 IAM 权限	34
文件系统的配置选项	34
删除文件系统	47
创建文件系统策略	48
创建接入点	49
删除接入点	52
标注资源	53
标签基本知识	53
标签限制	53
使用标签进行访问控制	54
标记资源	54
教程：创建可写的每用户子目录	55
挂载文件系统	57
Linux 的挂载注意事项	57
使用 EFS 挂载帮助程序	59
EFS 挂载帮助程序使用的挂载设置	60
获取支持日志	61
先决条件	62
在 EC2 Linux 上挂载	63
在 EC2 Mac 上挂载	65
从不同区域挂载	66
挂载单区文件系统	67
使用 IAM 授权挂载	70
使用 EFS 接入点进行挂载	71
挂载多个 EC2 实例	71
从另一个账户或 VPC 挂载	72
使用 NFS	75
先决条件	76
NFS 支持	76
安装 NFS 客户端	77
推荐的 NFS 挂载设置	79
使用 DNS 在 EC2 上挂载	80
使用 IP 地址挂载	82

自动挂载文件系统	84
新 EC2 Linux 实例	84
现有 EC2 Linux 实例	86
使用 NFS 的 Linux 和 Mac 实例	89
卸载文件系统	91
教程：使用创建和装载文件系统 AWS CLI	91
先决条件	92
设置 AWS CLI	93
步骤 1：创建 EC2 资源	94
步骤 2：创建 EFS 资源	99
步骤 3：挂载并测试文件系统	102
步骤 4：清除	105
教程：使用本地客户端挂载	106
先决条件	108
步骤 1：创建您的 EFS 资源	109
步骤 2：安装 NFS 客户端	110
步骤 3：在本地客户端上挂载 EFS 文件系统	111
第 4 步：清理资源并保护您的 AWS 账户	113
可选：加密传输中的数据	113
教程：挂载来自其它 VPC 的文件系统	116
先决条件	117
步骤 1：确定挂载目标的可用区 ID	118
步骤 2：确定挂载目标 IP 地址	118
步骤 3：为挂载目标添加主机条目	119
步骤 4：使用 EFS 挂载帮助程序挂载您的文件系统	120
第 5 步：清理资源并保护您的 AWS 账户	121
解决挂载问题	122
在 Windows 实例上挂载文件系统失败	122
服务器拒绝访问	122
自动挂载失败，并且实例没有响应	123
在 /etc/fstab 中挂载多个 Amazon EFS 文件系统失败	123
挂载命令失败，并显示“错误的 fs 类型”错误消息	124
挂载命令失败，并显示“不正确的挂载选项”错误消息	125
使用接入点挂载失败	125
在创建文件系统后文件系统挂载立即失败	125
文件系统挂载挂起，然后失败，并显示超时错误	126

使用 NFS 通过 DNS 名称挂载文件系统失败	126
文件系统挂载失败，并显示错误“nfs 未响应”	127
挂载目标生命周期状态停滞	127
挂载目标生命周期状态显示错误	127
挂载没有响应	128
挂载的客户端断开连接	128
对新挂载的文件系统的操作返回“坏文件句柄”错误	129
卸载文件系统失败	129
传输数据	130
使用 AWS DataSync	130
使用 AWS Transfer Family	130
AWS Transfer Family 与 Amazon EFS 一起使用的先决条件	131
将 EFS 文件系统配置为 AWS Transfer Family	132
管理文件系统	137
了解文件系统状态	137
管理挂载目标	138
挂载目标和可用区	138
创建挂载目标	141
删除挂载目标	146
更改挂载目标 VPC	147
更改挂载目标安全组	148
管理吞吐量	149
管理存储生命周期	151
生命周期管理的文件系统操作	151
配置生命周期策略	152
监控	155
监控工具	156
自动化工具	156
手动监控工具	156
文件系统计量	157
计量对象	157
计量的文件系统大小	158
计量吞吐量	159
查看存储类大小	160
使用监控指标 CloudWatch	162
CloudWatch 指标	162

访问 CloudWatch 指标	167
使用 CloudWatch 指标	169
将公制数学与 CloudWatch 指标配合使用	170
监控挂载尝试的成功与失败	175
创建警报	177
使用 记录 CloudTrail API 调用	179
Amazon EFS 中的信息 CloudTrail	179
了解 Amazon EFS 日志文件条目	180
文件系统的 Amazon EFS 日志 encrypted-at-rest文件条目	187
使用账单和使用情况报告	188
使用成本分配标签	188
用户定义的成本分配标签	189
账单报告	189
使用情况报告	190
了解账单和使用率报告	191
跟踪使用情况报告中的操作	195
性能规格	196
性能摘要	196
存储类	198
性能模式	198
吞吐量模式	199
选择吞吐量模式	199
弹性吞吐量	199
预配置吞吐量	200
对切换吞吐量和更改预置量的限制	202
性能提示	202
平均 I/O 大小	202
优化需要高吞吐量和 IOPS 的工作负载	203
同时连接	203
请求模型	203
NFS 客户端挂载设置	203
优化小文件性能	204
优化目录性能	205
优化 NFS read_ahead_kb 的大小	205
性能问题排查	206
无法创建 EFS 文件系统	207

拒绝访问 NFS 文件系统上允许的文件	207
访问 Amazon EFS 控制台时出错	207
Amazon EC2 实例挂起	207
写入大量数据的应用程序挂起	208
并行打开多个文件时，性能不佳	208
自定义 NFS 设置导致写入延迟	209
使用 Oracle Recovery Manager 创建备份的速度很慢	209
排查 AMI 和内核问题	210
无法更改所有权	210
由于客户端错误，文件系统重复执行操作	210
客户端发生死锁	211
列出大型目录中的文件需要很长时间	211
备份和复制数据	212
备份文件系统	212
如何 AWS Backup 与 Amazon EFS 配合使用	213
所需的 IAM 权限	215
备份性能	215
管理自动备份	215
复制文件系统	217
成本	218
复制性能	218
所需的 IAM 权限	218
配置到新文件系统的复制	219
配置到现有文件系统的复制	223
跨 AWS 账户复制	228
查看复制详细信息	233
删除复制配置	237
使用副本	239
保护数据	240
数据保护	240
数据加密	241
互连网络隐私	250
Identity and access management	251
受众	252
使用身份进行身份验证	252
使用策略管理访问	253

Amazon Elastic File System 如何与 IAM 配合使用	254
基于身份的策略示例	259
基于资源的策略示例	263
AWS 托管策略	266
使用标签	270
使用服务关联角色	273
问题排查	277
控制文件系统数据访问	279
默认文件系统策略	279
客户端的 EFS 操作	279
客户端的 EFS 条件键	280
文件系统策略示例	280
合规性验证	281
恢复能力	281
控制网络访问	281
使用 VPC 安全组	282
使用 VPC 端点	284
NFS 级别用户、组和权限	285
文件和目录权限	286
EFS 文件系统使用案例和权限示例	287
文件系统中文件和目录的用户和组 ID 权限	288
无根挤压	289
权限缓存	291
更改文件系统对象所有权	291
EFS 接入点	291
使用接入点工作	291
接入点可与挂载目标配合使用	292
强制执行用户身份	292
强制执行根目录	293
在 IAM 策略中使用接入点	295
屏蔽对文件系统的公共访问权限	296
使用以下方式阻止公众访问 AWS Transfer Family	296
“公有”的含义	297
网络隔离	300
配额	301
您可以提高的 Amazon EFS 配额	301

请求提高配额	303
您无法更改的 Amazon EFS 资源配额	304
NFS 客户端的配额	305
Amazon EFS 文件系统的配额	306
不支持的 NFSv4 .0 和 4.1 功能	307
其他注意事项	308
排查与限额有关的文件操作错误	308
检查打开的文件和文件锁	309
命令失败，并显示“超出磁盘配额”错误	309
命令失败，并显示“I/O 错误”	309
命令失败，并显示“文件名太长”错误	310
命令失败，并显示“未找到文件”错误	310
命令失败，并显示“链接太多”错误	310
命令失败，并显示“文件太大”错误	311
Amazon EFS API	312
API 终端节点	312
IPv4 端点	313
双栈 (IPv4 和 IPv6) 端点	313
指定端点	313
API 版本	314
相关主题	314
使用 Amazon EFS 的查询 API 请求速率	314
轮询	314
重试或批处理	315
计算睡眠间隔	315
操作	315
CreateAccessPoint	317
CreateFileSystem	326
CreateMountTarget	342
CreateReplicationConfiguration	359
CreateTags	367
DeleteAccessPoint	371
DeleteFileSystem	374
DeleteFileSystemPolicy	378
DeleteMountTarget	382
DeleteReplicationConfiguration	386

DeleteTags	390
DescribeAccessPoints	394
DescribeAccountPreferences	399
DescribeBackupPolicy	402
DescribeFileSystemPolicy	406
DescribeFileSystems	411
DescribeLifecycleConfiguration	417
DescribeMountTargets	421
DescribeMountTargetSecurityGroups	427
DescribeReplicationConfigurations	432
DescribeTags	437
ListTagsForResource	442
ModifyMountTargetSecurityGroups	446
PutAccountPreferences	451
PutBackupPolicy	455
PutFileSystemPolicy	459
PutLifecycleConfiguration	465
TagResource	474
UntagResource	479
UpdateFileSystem	483
UpdateFileSystemProtection	492
数据类型	497
AccessPointDescription	499
BackupPolicy	502
CreationInfo	503
Destination	505
DestinationToCreate	508
FileSystemDescription	511
FileSystemProtectionDescription	516
FileSystemSize	517
LifecyclePolicy	519
MountTargetDescription	521
PosixUser	524
ReplicationConfigurationDescription	526
ResourceIdPreference	528
RootDirectory	529

Tag	531
文档历史记录	532
.....	dl

什么是 Amazon Elastic File System ?

Amazon Elastic File System (Amazon EFS) 提供无服务器的完全弹性文件存储，这使您无需预置或管理存储容量和性能即可共享文件数据。Amazon EFS 可在不中断应用程序的情况下按需扩展到 PB 级，并可在您添加和移除文件时自动扩涨或收缩。Amazon EFS 具有简单的 Web 服务界面，可让您快速方便地创建和配置文件系统。该服务为您管理所有文件存储基础设施，这意味着您可以避免部署、修补和维护复杂文件系统配置的复杂性。

Amazon EFS 支持 Network File System 版本 4 (NFSv4.1 和 NFSv4.0) 协议，因此，您当前使用的应用程序和工具可以与 Amazon EFS 无缝协作。Amazon EFS 可通过大多数类型的 Amazon Web Services 计算实例进行访问，包括 Amazon EC2、Amazon ECS、Amazon EKS、AWS Lambda 和 AWS Fargate。

这项服务在可扩展性、可用性和持久性方面都十分出众。Amazon EFS 提供以下文件系统类型来满足您的可用性和持久性需求：

- 区域性 (推荐) – 区域性文件系统 (推荐) 在同一个 AWS 区域中跨多个地理位置分离的可用区以冗余方式存储数据。跨多个可用区存储数据能够提供数据的持续可用性，即使 AWS 区域中的一个或多个可用区不可用也是如此。
- 单区 – 单区文件系统在单个可用区内存储数据。将数据存储在单个可用区中能够为数据提供持续可用性。但在可用区整个或部分丢失或损坏的情况下 (不太可能发生)，存储在这些类型的文件系统中的数据可能会丢失。

有关文件系统类型的更多信息，请参阅[EFS 文件系统类型](#)。

Amazon EFS 提供了各种工作负载所需的吞吐量、IOPS 和低延迟。EFS 文件系统可以扩展到 PB 级，提高吞吐量，并允许从计算实例对您的数据进行大规模并行访问。对于大多数工作负载，我们建议使用默认模式，即通用性能模式和弹性吞吐量模式。

- 通用 – 通用性能模式非常适合延迟敏感型应用程序，如 Web 服务环境、内容管理系统、主目录和常规文件服务。
- 弹性 – 弹性吞吐量模式旨在自动纵向扩展或缩减吞吐量性能，以满足工作负载活动的需求。

有关 EFS 性能和吞吐量模式的更多信息，请参阅[Amazon EFS 性能规范](#)。

Amazon EFS 提供文件系统访问语义，如强大的数据一致性和文件锁定。有关更多信息，请参阅 [Amazon EFS 中的数据一致性](#)。Amazon EFS 还支持通过可移植操作系统接口（POSIX）权限控制对文件系统的访问。有关更多信息，请参阅 [在 Amazon EFS 中保护数据](#)。

Amazon EFS 支持身份验证、授权和加密功能，可帮助您满足安全性和合规性要求。Amazon EFS 支持两种形式的文件系统加密：传输中加密和静态加密。您可以在创建 EFS 文件系统时启用静态加密。如果启用，则会加密所有数据和元数据。您可以在挂载文件系统时启用传输中加密。NFS 客户端对 Amazon EFS 的访问由 AWS Identity and Access Management（IAM）策略和网络安全策略（如安全组）控制。有关更多信息，请参阅[Amazon EFS 中的数据加密](#)、[Amazon EFS 的身份和访问管理](#)和[控制 NFS 客户端对 EFS 文件系统的网络访问](#)。

 Note

不支持将 Amazon EFS 与基于 Microsoft Windows 的 Amazon EC2 实例一起使用。

您是否是首次接触 Amazon EFS 的用户？

如果您是首次接触 Amazon EFS 的用户，我们建议您按顺序阅读以下内容：

1. 有关 Amazon EFS 产品和定价的概述，请参阅 [Amazon EFS](#)。
2. 有关 Amazon EFS 技术概述，请参阅[Amazon EFS 的工作原理](#)。
3. 试试[开始使用](#)练习。

如需了解有关 Amazon EFS 的更多信息，请参阅以下主题，其中更详细地讨论了该服务：

- [创建和管理 EFS 资源](#)
- [管理 EFS 文件系统](#)
- [Amazon EFS API](#)

Amazon EFS 的工作原理

Amazon Elastic File System (EFS) 提供了一个简单、无服务器的 set-and-forget弹性文件系统。通过 Amazon EFS，您可以创建文件系统、在 Amazon EC2 实例上挂载文件系统，然后在文件系统中读取和写入数据。您可以通过网络文件系统版本 4.0 和 4.1 () 协议，在您的虚拟私有云 (VPCNFSv4) 中挂载 EFS 文件系统。我们建议将最新一代的 Linux NFSv4 .1 客户端（例如最新的亚马逊 Linux、亚马逊

Linux 2、Red Hat、Ubuntu 和 macOS Big Sur AMIs 中的客户端) 与 EFS 挂载帮助程序结合使用。有关说明，请参阅[安装 Amazon EFS 客户端](#)。

有关支持该协议的 Amazon EC2 Linux 和 macOS 亚马逊机器映像 (AMIs) 的列表，请参阅[NFS 支持](#)。对于某些 AMI，必须安装 NFS 客户端才能将文件系统挂载到 Amazon EC2 实例上。有关说明，请参阅[安装 NFS 客户端](#)。

您可以从多个 NFS 客户端并发访问 EFS 文件系统，因此超出单个连接的应用程序可以访问文件系统。Amazon EC2 和在同一可用区内的多个可用区中运行的其他 AWS 计算实例 AWS 区域 可以访问文件系统，因此许多用户可以访问和共享公共数据源。

有关可在 AWS 区域 何处创建 EFS 文件系统的列表，请参阅[Amazon Web Services 一般参考](#)。

要在 VPC 中访问 EFS 文件系统，请在 VPC 中创建一个或多个挂载目标。挂载目标提供可以在其中挂载 EFS 文件系统的 NFSv4 端点的 IP 地址。您使用其域名服务 (DNS) 名称挂载文件系统，该名称将解析为与 EC2 实例位于同一可用区中的 EFS 挂载目标的 IP 地址。您可以在 AWS 区域中的每个可用区中创建一个挂载目标。如果 VPC 的可用区中有多个子网，则在其中一个子网中创建挂载目标。随后，该可用区中的所有 EC2 实例都将共享该挂载目标。

Note

一个 EFS 文件系统一次只能在一个 VPC 中具有挂载目标。

挂载目标本身设计为具有高可用性。在设计实现高可用性和失效转移到其他可用区的功能时，请务必注意，尽管您的挂载目标在每个可用区中的 IP 地址和 DNS 均为静态，但它们是由多个资源支持的冗余组件。有关挂载目标的更多信息，请参阅[管理挂载目标](#)。

使用文件系统的 DNS 名称挂载文件系统后，您可以像使用任何其他文件系统一样使用它。有关 NFS 级别的权限和相关注意事项的信息，请参阅[网络文件系统 \(NFS \) 级别用户、组和权限](#)。

当使用 AWS Direct Connect 或连接到 Amazon VPC 时，您可以将 EFS 文件系统挂载到本地数据中心服务器上 Site-to-Site VPN。您可以将 EFS 文件系统挂载到本地服务器上，以便将数据集迁移到 EFS、启用云爆发方案或将本地数据备份到 Amazon EFS。

您可以在下文中找到有关如何将 Amazon EFS 与其它服务配合使用的说明。

主题

- [如何将 Amazon EFS 与 Amazon EC2 结合使用](#)
- [Amazon EFS 如何 AWS Direct Connect 与 AWS 托管 VPN 配合使用](#)

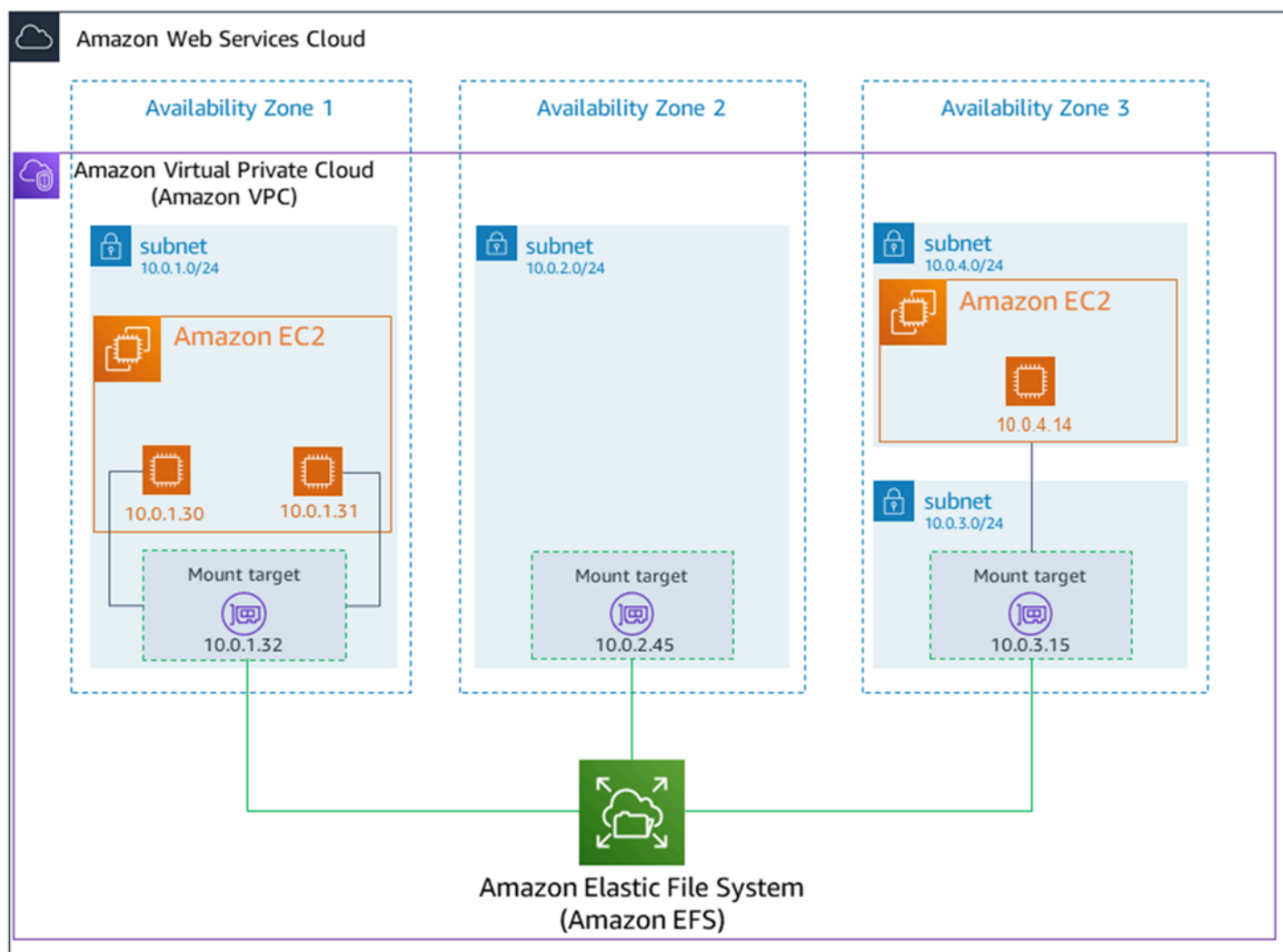
- [Amazon EFS 是如何使用的 AWS Backup](#)

如何将 Amazon EFS 与 Amazon EC2 结合使用

本节介绍如何将 Amazon EFS 区域性和单区文件系统挂载到 Amazon VPC 中的 EC2 实例。

区域性 EFS 文件系统

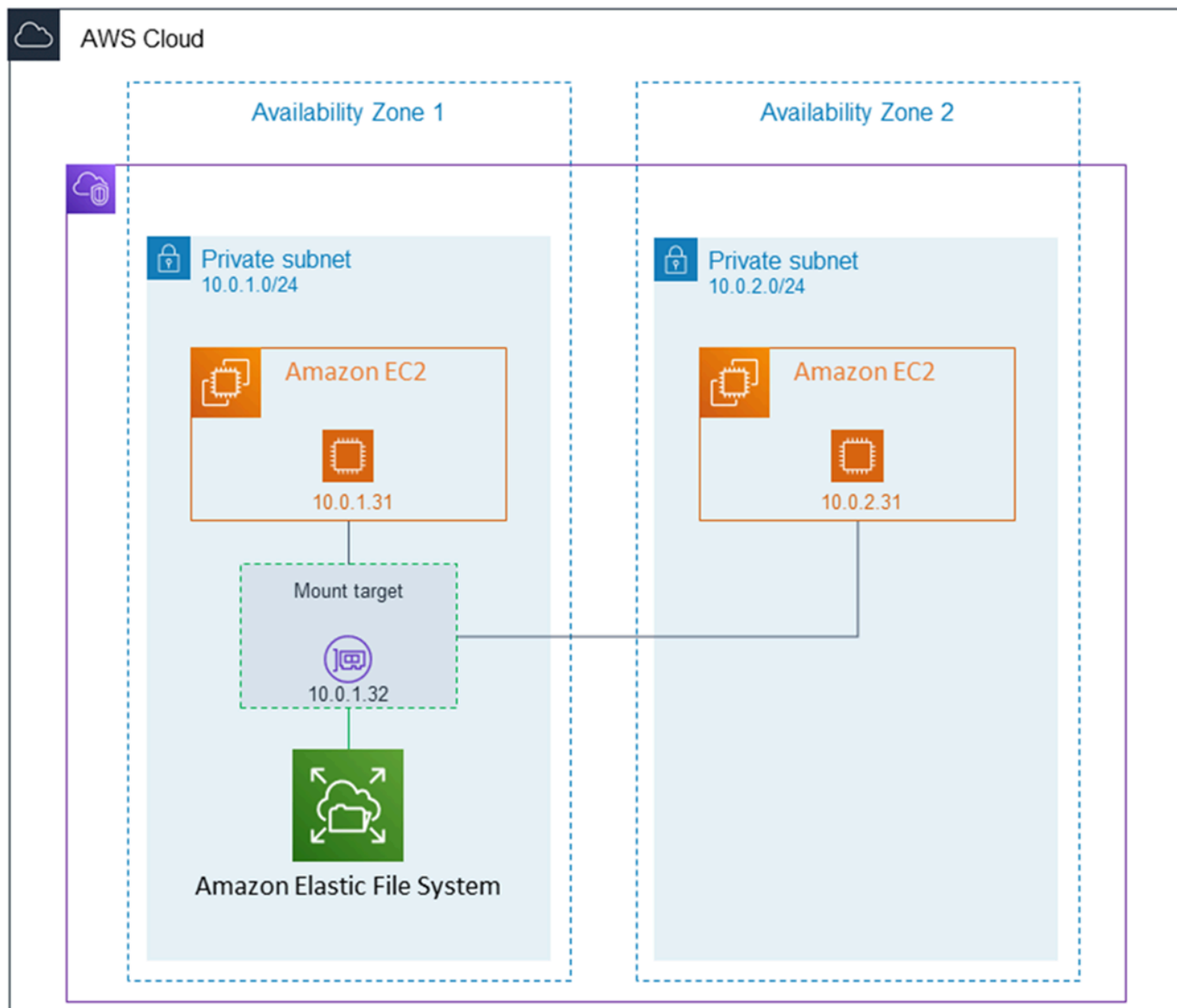
下图显示了多个 EC2 实例访问为 AWS 区域中的多个可用区配置的 Amazon EFS 文件系统。



在此图中，虚拟私有云（VPC）有三个可用区。因为文件系统是区域性的，所以在每个可用区中都创建了一个挂载目标。出于性能和成本原因，我们建议您从同一可用区内的挂载目标访问文件系统。其中一个可用区具有两个子网。但是，将仅在一个子网中创建挂载目标。有关更多信息，请参阅 [使用 EFS 挂载帮助程序挂载 EFS 文件系统](#)。

单区 EFS 文件系统

下图显示了多个 EC2 实例从 AWS 区域中的不同可用区访问单区文件系统。



在此图中，VPC 有两个可用区，每个可用区都有一个子网。由于文件系统类型为单区，因此它只能有一个挂载目标。为了提高性能并降低成本，我们建议您从与要挂载文件系统的 EC2 实例位于同一可用区的挂载目标访问该文件系统。

在此示例中，us-west-2c 可用区中的 EC2 实例将为访问不同可用区中的挂载目标支付 EC2 数据访问费用。有关更多信息，请参阅 [挂载单区文件系统](#)。

Amazon EFS 如何 AWS Direct Connect 与 AWS 托管 VPN 配合使用

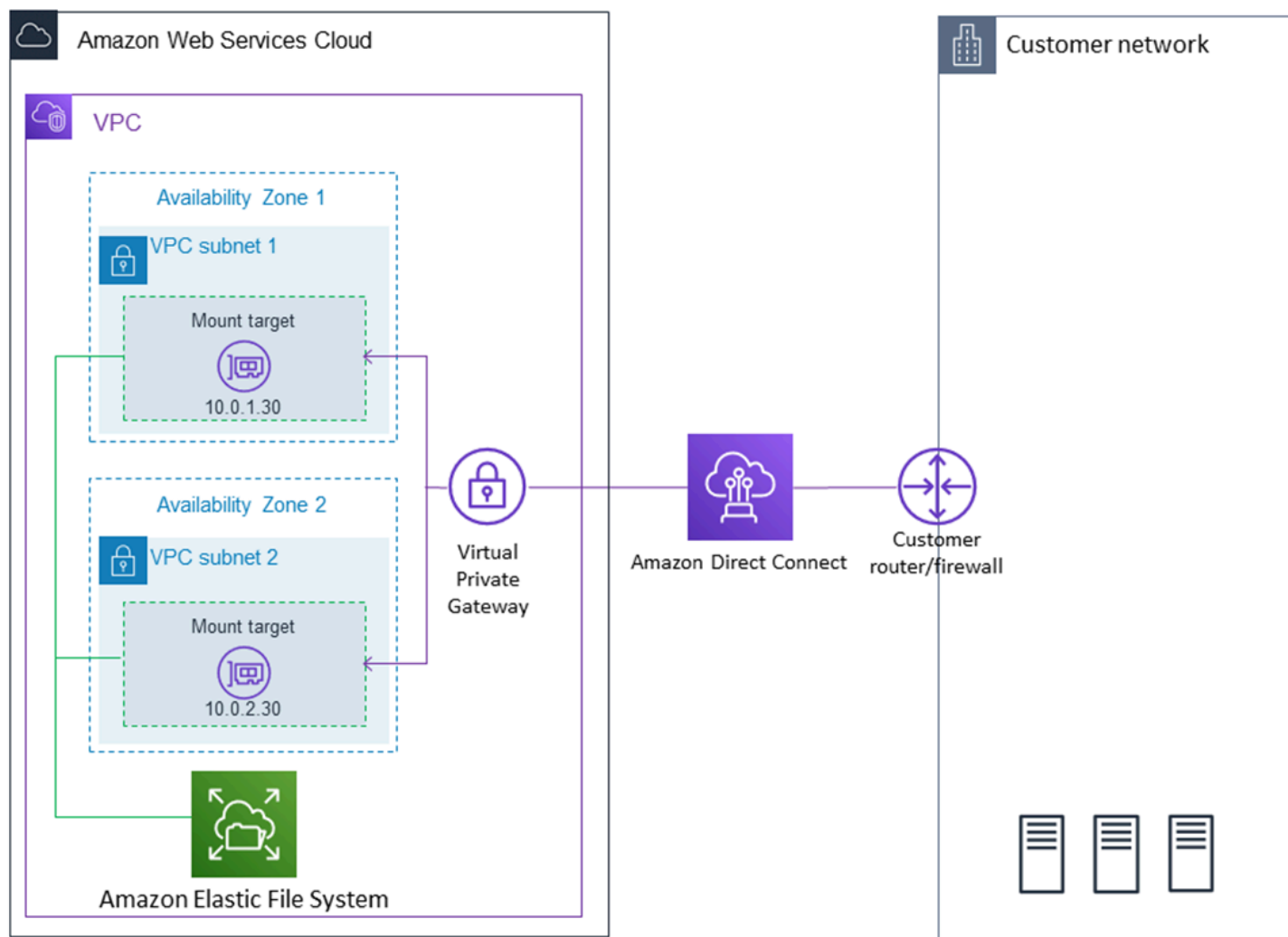
通过使用安装在本地服务器上的 Amazon EFS 文件系统，您可以将本地数据迁移到 Amazon EFS 文件系统中 AWS Cloud 托管的数据中。您还可以利用突增功能。换句话说，您可以将数据从本地服务器移动到 Amazon EFS，并在您的 Amazon VPC 中的一组 Amazon EC2 实例上对其进行分析。然后，您可以将结果永久存储在您的文件系统中，或将结果移回本地服务器。

在将 Amazon EFS 与本地服务器结合使用时，请注意以下事项：

- 您的本地服务器必须有一个基于 Linux 的操作系统。我们建议使用 Linux 内核版本 4.0 或更高版本。
- 为了简单起见，我们建议您使用挂载目标 IP 地址而不是 DNS 名称在本地服务器上挂载 Amazon EFS 文件系统。

对您的 Amazon EFS 文件系统的本地访问不会产生额外费用。您需要为 Direct Connect 连接您的 Amazon VPC 付费。有关更多信息，请参阅[Direct Connect 定价](#)。

下图显示了如何从本地（挂载了文件系统的本地服务器）访问 Amazon EFS 文件系统的示例。



如果您能通过本地服务器和 VPC 之间的 Direct Connect 连接到达挂载目标的子网，则可以使用您的 VPC 中的任何挂载目标。要从本地服务器访问 Amazon EFS，请向挂载目标安全组添加规则，以允许从本地服务器进入 NFS 端口（2049）的入站流量。有关更多信息，包括详细步骤，请参阅[先决条件](#)。

Amazon EFS 是如何使用的 AWS Backup

要为您的文件系统实施全面的备份，您可以将 Amazon EFS 与配合使用 AWS Backup。AWS Backup 是一项完全托管的备份服务，可轻松集中和自动执行云端和本地 AWS 服务的数据备份。使用 AWS Backup，您可以集中配置备份策略并监控 AWS 资源的备份活动。相对于备份操作，Amazon EFS 始终优先处理文件系统操作。要了解有关使用备份 EFS 文件系统的更多信息 AWS Backup，请参阅[备份 EFS 文件系统](#)。

Amazon EFS 的功能

以下是 Amazon EFS 的功能。

主题

- [身份验证和访问控制](#)
- [Amazon EFS 中的数据一致性](#)
- [EFS 文件系统的可用性和持久性](#)
- [复制](#)

身份验证和访问控制

您必须具有有效的凭证才能使用 Amazon EFS 管理控制台，以及发起 Amazon EFS API 请求，例如创建文件系统。此外，您还必须拥有创建或访问其他 EFS 和 AWS 资源的权限。

必须向您在 AWS Identity and Access Management (IAM) 中创建的用户和角色授予创建或访问资源的权限。有关权限的更多信息，请参阅 [Amazon EFS 的身份和访问管理](#)。

NFS 客户端的 IAM 授权是 Amazon EFS 的一个附加安全选项，它使用 IAM 来简化网络文件系统（NFS）客户端的大规模访问管理。通过针对 NFS 客户端的 IAM 授权，您可以使用 IAM 以本质上可扩展的方式管理对 EFS 文件系统的访问。NFS 客户端的 IAM 授权也针对云环境进行了优化。有关对 NFS 客户端使用 IAM 授权的更多信息，请参阅 [使用 IAM 控制文件系统访问](#)。

Amazon EFS 中的数据一致性

Amazon EFS 提供了应用程序期望从 NFS 获得 close-to-open 的一致性语义。

在 Amazon EFS 中，针对区域性文件系统的写入操作在以下情况下将持久存储在各个可用区中：

- 应用程序执行同步写入操作（例如，使用带 open 标记的 O_DIRECT Linux 命令或使用 fsync Linux 命令）。
- 应用程序关闭文件。

根据访问模式的不同，Amazon EFS 可以提供比 close-to-open 语义更强的一致性保证。执行同步数据访问和执行非附加写入操作的应用程序在数据访问方面具有 read-after-write 一致性。

文件锁定

NFS 客户端应用程序可以使用 NFS 版本 4 文件锁定（包括字节范围锁定）对 Amazon EFS 文件进行读写操作。

请记住以下关于 Amazon EFS 如何锁定文件的内容：

- Amazon EFS 仅支持建议锁定，read/write 操作在执行前不检查锁定是否存在冲突。例如，为了避免原子操作出现文件同步问题，您的应用程序必须了解 NFS 语义（例如 close-to-open 一致性）。
- 在已连接的所有实例和访问文件的用户中，任何一个特定文件最多可以有 512 个锁。

EFS 文件系统的可用性和持久性

本节介绍 Amazon Elastic File System (Amazon EFS) 文件系统的文件系统类型和存储类选项。

EFS 文件系统类型

Amazon EFS 提供区域性和单区文件系统类型。

- 区域性 – 区域性文件系统（推荐）在同一个 AWS 区域中跨多个地理位置分离的可用区以冗余方式存储数据。跨多个可用区存储数据可为数据提供持续可用性，即使其中一个或多个可用区不可用 AWS 区域也是如此。
- 单区 – 单区文件系统在单个可用区内存储数据。将数据存储在单个可用区中能够为数据提供持续可用性。但在可用区整个或部分丢失或损坏的情况下（不太可能发生），存储在这些类型的文件系统中的数据可能会丢失。

在不太可能发生的全部或部分 AWS 可用区丢失或损坏的情况下，One Zone 存储类别中的数据可能会丢失。例如，火灾和水灾等事件可能导致数据丢失。除了这些类型的事件外，我们的单区存储类还使用与区域性存储类相似的工程设计，以保护对象免受独立磁盘、主机和机架级故障的影响，每种存储类都旨在提供 99.999999999% 的数据持久性。

为了增强数据保护，Amazon EFS 会自动使用备份单区域文件系统 AWS Backup。您可以将文件系统备份恢复到任何可操作的可用区 AWS 区域，也可以将其恢复到其他可用区 AWS 区域。使用创建和管理的 EFS 文件系统备份 AWS Backup 会复制到三个可用区，并且专为持久性而设计。有关更多信息，请参阅[中的弹性 AWS Backup](#)。

Note

单区域文件系统仅适用于某些可用区。有关列出了可在其中使用单区文件系统的可用区的表格，请参阅[单区文件系统支持的可用区](#)。

下表比较了文件系统类型，包括其可用性、持久性和其它注意事项。

文件系统类型	设计专门针对	持久性 (设计目标)	可用性	可用区	其他考虑因素
区域性	需要最高持久性和可用性的数据。	99.999999 999% (11 个 9)	99.99%	>=3	无
单区	不需要最高持久性和可用性的数据。	99.999999 999% (11 个 9)	99.99%	1	无法灵活地应对可用区丢失的情况

单区文件系统支持的可用区

单区域文件系统仅适用于某些可用区。下表列出了您可以在其中使用 One Z IDs one 文件系统的每个可用区的 AWS 区域 和可用区。要查看您账户中可用 IDs 区的映射，请参阅 Resource A ccess Manager 用户指南中的 AWS AWS 资源[可用区 IDs](#)。

支持单区文件系统的可用区

AWS 区域 名称	AWS 区域 代码	支持的可用区 IDs
美国东部 (俄亥俄州)	us-east-2	use2-az1、use2-az2、use2-az3
美国东部 (弗吉尼亚州北部)	us-east-1	use1-az1、use1-az2、use1-az4、use1-az5、use1-az6
美国西部 (北加利福尼亚)	us-west-1	usw1-az1、usw1-az3

AWS 区域 名称	AWS 区域 代码	支持的可用区 IDs
美国西部 (俄勒冈州)	us-west-2	usw2-az1、usw2-az2、usw2-az3、usw2-az4
非洲 (开普敦)	af-south-1	afs1-az1、afs1-az2、afs1-az3
亚太地区 (香港)	ap-east-1	ape1-az1、ape1-az2、ape1-az3
亚太地区 (孟买)	ap-south-1	aps1-az1、aps1-az2、aps1-az3
亚太地区 (大阪)	ap-northeast-3	apne3-az1、apne3-az2、apne3-az3
亚太地区 (首尔)	ap-northeast-2	apne2-az1、apne2-az2、apne2-az3
亚太地区 (新加坡)	ap-southeast-1	apse1-az1、apse1-az2
亚太地区 (悉尼)	ap-southeast-2	apse2-az1、apse2-az2、apse2-az3
亚太地区 (东京)	ap-northeast-1	apne1-az1、apne1-az4
加拿大 (中部)	ca-central-1	cac1-az1、cac1-az2
中国 (北京)	cn-north-1	cnn1-az1、cnn1-az2
中国 (宁夏)	cn-northwest-1	cnnw1-az1、cnnw1-az2、cnnw1-az3
欧洲地区 (法兰克福)	eu-central-1	uc1-az1、eu1az2、eu1-az3
欧洲地区 (爱尔兰)	eu-west-1	euw1-az1、euw1-az2、euw1-az3
欧洲 (伦敦)	eu-west-2	euw2-az1、euw2-az2

AWS 区域 名称	AWS 区域 代码	支持的可用区 IDs
欧洲地区 (米兰)	eu-south-1	eus1-az1、eus1-az2、eus1-az3
欧洲地区 (巴黎)	eu-west-3	euw3-az1、euw3-az3
欧洲地区 (斯德哥尔摩)	eu-north-1	eun1-az1、eun1-az2、eun1-az3
中东 (巴林)	me-south-1	mes1-az1、mes1-az2、mes1-az3
南美洲 (圣保罗)	sa-east-1	sae1-az1、sae1-az2、sae1-az3
AWS GovCloud (美国东部)	us-gov-east-1	usge1-az1、usge1-az2、usge1-az3
AWS GovCloud (美国西部)	us-gov-west-1	usgw1-az1、usgw1-az2、usgw1-az3

EFS 存储类

Amazon EFS 提供了专为实现最有效的存储而设计的不同存储类，具体取决于用例。

- EFS 标准 – EFS 标准存储类使用固态驱动器 (SSD) 存储为频繁访问的文件提供最低延迟级别。新的文件系统数据首先写入 EFS 标准存储类，然后可以使用生命周期管理分层到 EFS 不频繁访问存储类和 EFS 存档存储类。
- EFS 不频繁访问 (IA)– 一种成本优化的存储类，适用于每个季度仅访问几次的数据。
- EFS 归档 – 一种成本优化的存储类，适用于每年访问几次或更少次的数据。

具有弹性吞吐量的 EFS 文件系统支持 EFS 存档存储类。一旦文件系统在存档存储类中有数据，就不能将文件系统的吞吐量更新为“突发”或“预调配”。

比较存储类

下表对存储类进行了比较。有关每个存储类的性能的更多详细信息，请参阅[Amazon EFS 性能规范](#)。

存储类	设计专门针对	第一个字节读取延迟	持久性 (设计目标) ¹	可用性 SLA	可用区	每个文件的最低账单费用 ²	最小存储持续时间
EFS 标准	需要亚毫秒级快速延迟性能的活跃数据	亚毫秒级	99.999999999% (11 个 9)	99.99% (区域性)	=>3 (区域性)	不适用	不适用
EFS 不频繁访问	每季度仅访问几次的非活动数据。	几十毫秒		99.9% (单区)	1 (单区)	128KiB	不适用
EFS 归档	每年访问几次或更少次的非活动数据	几十毫秒		99.9% (区域性)	=>3 (区域性)	128KiB	90 天

¹ 由于 One Zone 文件系统将数据存储存储在单个 AWS 可用区中，因此在发生影响可用区内所有数据副本的灾难或其他故障或可用区被破坏时，存储在这些类型的文件系统的数据可能会丢失。

²太平洋时间 2023 年 11 月 26 日中午 12 点或之后更新的生命周期策略会将小于 128 KiB 的文件分层到 IA 类中。有关 Amazon EFS 如何对各个文件和元数据进行计量和计费的更多信息，请参阅[Amazon EFS 如何报告文件系统和对象大小](#)。

存储类别计费

您需要为每个存储类别中的数据量付费。读取 IA 或存档存储中的文件时，或对于使用生命周期管理在存储类之间转换的数据，您还需要支付数据访问费用。AWS 账单显示每种存储类的容量，以及对文件系统的存储类的计量访问。要了解更多信息，请参阅 [Amazon EFS 定价](#)。

此外，不频繁访问 (IA) 和归档存储类对于每个 128KiB 的文件具有最低账单费用。对小于 128KiB 的文件的支持仅适用于太平洋时间 2023 年 11 月 26 日中午 12:00 或之后更新的生命周期策略。有关 Amazon EFS 如何对各个文件和元数据进行计量和计费的更多信息，请参阅[Amazon EFS 如何报告文件系统和对象大小](#)。

其他计费注意事项视吞吐量模式而定。

- 对于采用弹性吞吐量模式的文件系统，除了存储费用，您还需要按每月为文件系统传输的元数据和数据总量付费。
- 对于使用预调配吞吐量的文件系统，将对超出为您提供的预调配吞吐量（基于 EFS 标准存储类中的数据量）的部分进行计费。
- 对于使用突增吞吐量的文件系统，允许的吞吐量仅基于 EFS 标准存储类中存储的数据量确定。

有关 EFS 吞吐量模式的更多信息，请参阅[吞吐量模式](#)。

Note

使用备份支持生命周期管理的 EFS 文件系统时 AWS Backup，您不会产生数据访问费用。要了解有关 Amazon AWS Backup EFS 的更多信息，请参阅[备份 EFS 文件系统](#)。

生命周期管理

要管理您的文件系统，使其在整个生命周期中经济高效地存储，请使用生命周期管理。生命周期管理根据为文件系统定义的生命周期配置，自动在存储类之间转换数据。生命周期配置是一组生命周期策略，用于定义何时将文件系统数据转换为其它存储类。有关更多信息，请参阅[管理存储生命周期](#)。

复制

您可以使用复制功能根据自己的喜好创建 Amazon EFS 文件系统的副本。AWS 区域 复制会自动透明地将 EFS 文件系统上的数据和元数据复制到在您选择的中创建的新目标 EFS 文件系统。AWS 区域 EFS 会自动使源文件系统和目标文件系统保持同步。复制是连续的，旨在提供分钟级的恢复点目标 (RPO) 和恢复时间目标 (RTO)。这些特征可以帮助您实现合规性和业务连续性目标。有关更多信息，请参阅[复制 EFS 文件系统](#)。

开始使用 Amazon EFS

如果您是首次使用 Amazon Elastic File System (Amazon EFS)，请按以下步骤操作，开始使用您的第一个 EFS 文件系统。

1. [查看入门先决条件](#)
2. [创建 EFS 文件系统并启动 EC2 实例](#)
3. [使用将文件传输到您的 EFS 文件系统 AWS DataSync](#)
4. [清理资源并保护您的 AWS 账户](#)

先决条件

在完成入门步骤之前，请确保您符合以下要求：

- 您具有设置好的 Amazon EC2，并且熟悉如何启动 EC2 实例。你需要一个 AWS 账户、一个具有管理权限的用户、一个 key pair 和一个安全组。有关更多信息，请参阅《Amazon EC2 用户指南》中的 [Amazon EC2 入门](#)。
- 您的 Amazon Virtual Private Cloud (Amazon VPC)、EC2 和 EFS 资源都位于同一 AWS 区域中，并且您在该区域具有默认 VPC。如果没有默认 VPC，或者要从具有新的或现有安全组的新 VPC 中挂载文件系统，请参阅[使用 VPC 安全组](#)。
- 您没有更改默认安全组的默认入站访问规则。

您也可以使用 AWS Command Line Interface (AWS CLI) 命令执行类似的入门练习来调用 EFS API。有关更多信息，请参阅 [教程：创建 EFS 文件系统并将其挂载到 EC2 实例上 AWS CLI](#)。

创建 EFS 文件系统并启动 EC2 实例

在确保您具备进行本入门练习的先决条件后，您可以创建 EFS 文件系统并启动 EC2 实例。要完成所有必要步骤以开始使用第一个 EFS 文件系统，最快的方法是在实例启动期间使用 EC2 新启动向导。

Note

不能将 Amazon EFS 与基于 Microsoft Windows 的 EC2 实例结合使用。

使用 EC2 启动向导创建 EFS 文件系统并启动您的 EC2 实例

有关在创建 EC2 实例启动时创建和挂载 EFS 文件系统的说明，请参阅《Amazon EC2 用户指南》中的[将 Amazon EFS 与 Amazon EC2 结合使用](#)。

以下是您在实例启动期间创建 EFS 文件系统时要执行的步骤。

1. 使用您选择的密钥对和网络设置创建在 Linux 操作系统上运行的 EC2 实例。
2. 创建具有推荐设置并自动挂载到 EC2 实例的共享 EFS 文件系统。
3. 启动 EC2 实例，以便 EFS 文件系统随时可用于文件传输功能。

或者，您可以在 Amazon EFS 控制台中，使用推荐设置或自定义设置创建文件系统。您还可以使用 AWS CLI 和 API 创建文件系统。有关创建文件系统的所有选项的更多信息，请参阅[创建 EFS 文件系统](#)。

使用将文件传输到您的 EFS 文件系统 AWS DataSync

创建 EFS 文件系统后，您可以使用将文件从现有文件系统传输到该文件系统 AWS DataSync。DataSync 是一项数据传输服务，可通过互联网简化、自动化和加速本地存储系统与 AWS 存储服务之间的数据移动和复制，或者。Direct Connect DataSync 可以传输您的文件数据以及文件系统元数据，例如所有权、时间戳和访问权限。

有关 DataSync 的更多信息，请参阅[AWS DataSync](#)。

先决条件

在将文件传输到 EFS 文件系统之前，请确保您具备以下条件：

- 您可以从其中传输数据的源 NFS 文件系统。需要能够通过 NFS 版本 3、版本 4 或 4.1 访问该源系统。示例文件系统包括位于本地数据中心的文件系统、自主管理型云端文件系统和 EFS 文件系统。
- 您已准备好使用 DataSync。要了解更多信息，请参阅《AWS DataSync 用户指南》中的[AWS DataSync 入门](#)。

使用将文件传输到 EFS 文件系统 AWS DataSync

有关使用 DataSync 将文件传输到 EFS 文件系统的说明，请参阅 AWS DataSync 用户指南 AWS DataSync 中的[使用传输数据](#)。

以下是使用将文件传输到 EFS 文件系统时要执行的步骤 DataSync。

1. 连接到您的 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 在您的环境中下载、部署并激活代理。
3. 创建并配置源和目标位置。
4. 创建并配置任务。
5. 运行任务，将文件从源传输到目标。

清理资源并保护您的 AWS 账户

完成本入门练习后，请执行以下步骤以清理资源并保护您的 AWS 账户。

清理资源并保护您的账户

1. 连接到您的 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 使用以下命令卸载 EFS 文件系统。

```
$ sudo umount efs
```

3. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
4. 删除您在入门练习的第一步中创建的 EFS 文件系统。
 - a. 选择要从文件系统列表中删除的 EFS 文件系统。
 - b. 对于操作，选择删除文件系统。
 - c. 在永久删除文件系统对话框中，键入要删除的 EFS 文件系统的文件系统 ID，然后选择删除文件系统。
5. 终止您为此入门练习启动的 EC2 实例。有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[终止 Amazon EC2 实例](#)。
6. 如果您为本入门练习创建了安全组，则请将其删除。有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[Delete a security group](#)。

Warning

不要删除您的 VPC 的默认安全组。

安装 Amazon EFS 客户端

我们建议您安装 Amazon EFS 客户端 (`amazon-efs-utils`)，它是 Amazon EFS 工具的开源集合。Amazon EFS 客户端包含挂载帮助程序，它是一个帮助简化 EFS 文件系统挂载的程序。该客户端还支持使用 Amazon CloudWatch 来监控 EFS 文件系统的挂载状态，它还包括一些工具，可以更轻松地对 Amazon EFS 文件系统传输中的数据执行加密。

您可以在运行[支持的发行版](#)的 Amazon EC2 (EC2) 实例上手动安装 Amazon EFS 客户端。对于某些支持的操作系统，您也可以 AWS Systems Manager 将其配置为自动安装或更新软件包。有关可与之配合使用的发行版列表 AWS Systems Manager，请参阅[Systems Manager 支持的操作系统](#)。

Important

建议您始终使用最新版本的 `amazon-efs-utils`，以确保访问其全部功能。例如，2.3 或更高版本支持使用 IPv6 地址进行挂载，但早期版本不支持。

主题

- [EFS 工具的依赖项](#)
- [支持的发行版](#)
- [手动安装 Amazon EFS 客户端](#)
- [使用自动安装或更新 Amazon EFS 客户端 AWS Systems Manager](#)
- [安装和升级 botocore](#)
- [升级 stunnel](#)
- [启用 FIPS 模式](#)

EFS 工具的依赖项

`amazon-efs-utils` 具有以下依赖项，将在安装 `amazon-efs-utils` 软件包时安装这些依赖项：

- NFS 客户端
 - RHEL、CentOS、Amazon Linux 和 Fedora 发行版的 `nfs-utils`
 - Debian 和 Ubuntu 发行版的 `nfs-common`
- 网络中继 (`stunnel` 软件包 4.56 或更高版本)

- Python (3.4 或更高版本)
- OpenSSL 1.0.2 或更高版本

 Note

默认情况下，当使用带有传输层安全性协议 (TLS) 的 EFS 挂载帮助程序时，挂载帮助程序会强制实施证书主机名检查。EFS 挂载帮助程序使用 stunnel 程序提供 TLS 功能。某些版本的 Linux 不包含默认支持这些 TLS 功能的 stunnel 版本。在使用这些 Linux 版本之一时，使用 TLS 挂载 EFS 文件系统将失败。

安装 amazon-efs-utils 软件包后，升级 stunnel。请参阅[升级 stunnel](#)。

您可以使用 AWS Systems Manager 来管理 Amazon EFS 客户端，并自动执行在 EC2 实例上安装或更新 amazon-efs-utils 软件包所需的任务。有关更多信息，请参阅 [使用自动安装或更新 Amazon EFS 客户端 AWS Systems Manager](#)。

有关加密问题，请参阅[排除加密故障](#)。

支持的发行版

已针对以下 Linux 和 Mac 发行版对 Amazon EFS 客户端进行了验证：

发行版	包类型	init 系统
Amazon Linux 2023 (AL2023)	rpm	systemd
Amazon Linux 2 (AL2)	rpm	systemd
CentOS 8	rpm	systemd
Amazon Linux 1 (AL1) 2017.09	rpm	upstart

 Note

AL1 AMI 于 2023 年 12 月 31 日结束其生命周期，并在自 2024 年 4 月或之后发布的 amazon-

发行版	包类型	init 系统
efs-utils 软件包 (版本 2.0 及更高版本) 中不受支持。		
Debian 11	deb	systemd
Fedora 29 - 32	rpm	systemd
macOS Big Sur		launchd
macOS Monterey		launchd
macOS Ventura		launchd
macOS Sonoma		launchd
OpenSUSE Leap, Tumbleweed	rpm	systemd
Oracle8	rpm	systemd
红帽企业 Linux (RHEL) 8、9、10	rpm	systemd
SUSE Linux Enterprise Server (SL ES) 12、15	rpm	systemd
Ubuntu 16.04 LTS、18.04 LTS、20.04 LTS、22.04 LTS	deb	systemd

有关该软件包已通过验证的受支持发行版的完整列表，请参阅 Github 上的 [amazon-efs-utils README](#)。

手动安装 Amazon EFS 客户端

您可以在 Amazon EC2 Linux (EC2) 实例，以及运行 macOS Big Sur、macOS Monterey 和 macOS Ventura 的 EC2 Mac 实例上手动安装 Amazon EFS 客户端。有关支持 Amazon EFS 客户端的发行版列表，请参阅[支持的发行版](#)

以下几部分将介绍受支持的操作系统的安装过程。

主题

- [在 Amazon EC2 Linux 实例上安装 Amazon EFS 客户端](#)
- [在其他 Linux 发行版上安装 Amazon EFS 客户端](#)
- [在运行 macOS Big Sur、macOS Monterey 或 macOS Ventura 的 EC2 Mac 实例上安装 Amazon EFS 客户端](#)

在 Amazon EC2 Linux 实例上安装 Amazon EFS 客户端

用于从以下位置在 Amazon EC2 Linux 实例上安装的 `amazon-efs-utils` 软件包：

- 适用于 Amazon Linux 的亚马逊机器映像 (AMI) 软件包存储库。以下说明用于从 AMI 软件包存储库安装 `amazon-efs-utils` 软件包。
- AWS [efs-utils](#) GitHub 存储库。有关从安装 `amazon-efs-utils` 软件包的更多信息 GitHub，请参阅[在其他 Linux 发行版上安装 Amazon EFS 客户端](#)。

Note

- 如果您正在使用 Direct Connect，可以在中找到安装说明[先决条件](#)。
- Amazon Linux 1 (AL1) AMI 于 2023 年 12 月 31 日结束其生命周期，并在自 2024 年 4 月及之后发布的 `amazon-efs-utils` 软件包 (版本 2.0 及更高版本) 中不受支持。我们建议您将应用程序升级到 Amazon Linux 2023 (AL2023)，其中包含直到 2028 年的长期支持。

在 EC2 Linux 实例上安装 AMI ##### `amazon-efs-utils` 软件包

1. 确保您创建了一个 AL2023 或 Amazon Linux 2 (AL2) EC2 实例。有关如何执行该操作的信息，请参阅[步骤 1：启动实例](#)。
2. 通过安全 Shell (SSH) 访问您的实例的终端，然后使用相应的用户名登录。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
3. 要安装 `amazon-efs-utils` 软件包，请运行以下命令。

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

在其他 Linux 发行版上安装 Amazon EFS 客户端

如果您不想从 Amazon Linux AMI amazon-efs-utils 软件包存储库中获取软件包，也可以在上使用该软件包 GitHub。

克隆该软件包后，您可以使用以下方法之一构建并安装 amazon-efs-utils，具体取决于您的 Linux 发行版支持的软件包类型：

- RPM - AL2023、Amazon Linux 2、Red Hat Linux、CentOS 和类似的发行版支持该软件包类型。
- DEB – Ubuntu、Debian 和类似的发行版支持该软件包类型。

有关为其它 Linux 发行版安装 amazon-efs-utils 软件包的说明，请参阅 Github 上 amazon-efs-utils 自述文件中的 [On other Linux distributions](#)。

在运行 macOS Big Sur、macOS Monterey 或 macOS Ventura 的 EC2 Mac 实例上安装 Amazon EFS 客户端

amazon-efs-utils 软件包可安装在运行 macOS Big Sur、macOS Monterey 或 macOS Ventura 的 EC2 Mac 实例上。

有关在 Mac 实例上安装 amazon-efs-utils 软件包的说明，请参阅 Github 上 amazon-efs-utils 自述文件中的 [On MacOS Big Sur, macOS Monterey, macOS Sonoma and macOS Ventura distribution](#)。

后续步骤

在 EC2 实例上安装 amazon-efs-utils 后，继续执行后续步骤以挂载文件系统：

- [安装botocore](#)以便您可以使用 Amazon CloudWatch 监控文件系统的挂载状态。
- [升级到最新版本的 stunnel](#) 以启用传输中数据加密。
- 使用 EFS 挂载帮助程序[挂载您的文件系统](#)。

使用自动安装或更新 Amazon EFS 客户端 AWS Systems Manager

您可以使用 AWS Systems Manager 来简化 Amazon EFS 客户端 (amazon-efs-utils) 的管理。AWS Systems Manager 是一项可用于查看和控制基础架构的 AWS 服务 AWS。借 AWS Systems

Manager 助，您可以自动执行在 Amazon EC2 (EC2) 实例上安装或更新amazon-efs-utils软件包所需的任务。通过 Systems Manager 功能（例如 Distributor 和 State Manager），您可以自动执行以下流程：

- 维护对 Amazon EFS 客户端的版本控制。
- 集中存储并系统地将 Amazon EFS 客户端分发到您的 Amazon EC2 实例。
- 自动化将 EC2 实例保持在定义状态的过程。

有关更多信息，请参阅 [AWS Systems Manager 《用户指南》](#)。

Amazon EFS 客户端在安装过程中的作用

您可以使用 Amazon EFS 客户端自动监控亚马逊 CloudWatch 日志中的文件系统挂载状态，并针对所选 Linux 发行版升级stunnel到最新版本。当您使用 Systems Manager 在 Amazon EC2 实例上安装 Amazon EFS 客户端时，它会执行以下操作：

- 使用[安装和升级 botocore](#)中描述的相同步骤安装 botocore 软件包。Amazon EFS 客户端使用 botocore 监控 EFS 文件系统的挂载状态。
- 通过更新启用对 CloudWatch 日志中的 EFS 文件系统装载状态的监控efs-utils.conf。有关更多信息，请参阅 [监控挂载尝试的成功与失败](#)。
- 对于运行 RHEL7 或 CentOS7 的 EC2 实例，Amazon EFS 客户端会自动升级 stunnel，如[升级 stunnel](#) 中所述。要使用 TLS 成功挂载 EFS 文件系统，需要升级 stunnel，而 RHEL7 和 CentOS7 附带的 stunnel 版本不支持 Amazon EFS 客户端（amazon-efs-utils）。

Systems Manager 支持的操作系统

您的 EC2 实例必须运行以下操作系统之一，才能用于自动更新或安装 Amazon EFS 客户端。AWS Systems Manager

平台	平台版本	架构
Amazon Linux 2023 (AL2023)	AL2023	x86_64、arm64 (Graviton2 或更高版本的处理器)
Amazon Linux 2 (AL2)	2.0	x86_64、arm64 (Amazon Linux 2 , A1 实例类型)

平台	平台版本	架构
Amazon Linux 1 (AL1)	2017.09、2018.03	x86_64
Note Amazon Linux 1 (AL1) AMI 于 2023 年 12 月 31 日结束其生命周期，并在自 2024 年 4 月及之后发布的 amazon-efs-utils 软件包 (版本 2.0 及更高版本) 中不受支持。我们建议您将应用程序升级到 Amazon Linux 2023 (AL2023) ，其中包含直到 2028 年的长期支持。		
CentOS	7、8	x86_64
Red Hat Enterprise Linux (RHEL)	8、9	x86_64 , arm64
SUSE Linux Enterprise Server (SLES)	12、15	x86_64
Ubuntu Server	16.04、18.04、20.04	x86_64、arm64 (Ubuntu Server 16 及更高版本，A1 实例类型)

配置 AWS Systems Manager 安装 EFS 客户端

要设置 Systems Manager 以自动安装或更新 amazon-efs-utils 软件包，需要进行两项一次性配置。

1. 使用所需权限配置 AWS Identity and Access Management (IAM) 实例配置文件。
2. 配置状态管理器用于安装或更新的关联（包括时间表）。

步骤 1：使用所需权限配置 IAM 实例配置文件

默认情况下，AWS Systems Manager 无权管理您的 Amazon EFS 客户端，也无权安装或更新 amazon-efs-utils 软件包。您必须通过使用 AWS Identity and Access Management (IAM) 实例配置文件来授予对 Systems Manager 的访问权限。实例配置文件是一个容器，可在启动时将 IAM 角色信息传递给 Amazon EC2 (EC2) 实例。

使用 AmazonElasticFileSystemsUtils AWS 托管权限策略为角色分配适当的权限。您可以为实例配置文件创建新角色，或将 AmazonElasticFileSystemsUtils 权限策略添加到现有角色中。然后，您必须使用此实例配置文件启动 EC2 实例。有关更多信息，请参阅[配置 Systems Manager 所需的实例权限](#)。

步骤 2：配置 State Manager 使用的关联

amazon-efs-utils 软件包包含在发行版中，可供您随时部署到托管 EC2 实例。要查看可供安装的最新版本，您可以使用 AWS Systems Manager 控制台或首选的 AWS 命令行工具。amazon-efs-utils 要访问分销商，请打开<https://console.aws.amazon.com/systems-manager/>并在左侧导航窗格中选择分销商。在由 Amazon 拥有部分中找到 AmazonEFSUtils。选择 AmazonEFSUtils 以查看软件包详细信息。有关更多信息，请参阅[查看软件包](#)。

使用 State Manager，您可以立即或按计划在托管 EC2 实例上安装或更新 amazon-efs-utils 软件包。此外，您也可以确保自动将 amazon-efs-utils 安装在新 EC2 实例上。有关使用 Distributor 和 State Manager 安装或更新软件包的更多信息，请参阅[使用 Distributor](#)。

要使用 Systems Manager 控制台在实例上自动安装或更新 amazon-efs-utils 软件包，请参阅[计划安装或更新软件包（控制台）](#)。这将提示您为 State Manager 创建关联，该关联定义要应用于一组实例的状态。创建关联时使用以下输入：

- 对于参数，选择“操作”>“安装和安装类型”>“In-place更新”。
- 对于目标，建议的设置是选择所有实例，将所有新的和现有的 EC2 实例注册为目标，以自动安装或更新 AmazonEFSUtils。或者，您也可以指定实例标签，手动选择实例，或者选择资源组以将关联应用于实例的子集。如果您指定实例标签，则必须使用标签启动您的 EC2 实例，以允许 S AWS Systems Manager 自动安装或更新 Amazon EFS 客户端。
- 对于指定计划，建议的 AmazonEFSUtils 设置是每 30 天。您可以使用控件为关联创建 cron 或频率计划。

AWS Systems Manager 要使用将 EFS 文件系统挂载到多个 EC2 实例，请参阅[将 EFS 挂载到多个 EC2 实例](#)。

安装和升级 **botocore**

Amazon EFS 客户端 **botocore** 用于与其他 AWS 服务进行交互。如果要在 Logs 中监控 EFS 文件系统的装载尝试成功或失败，则需要此 CloudWatch 选项。有关更多信息，请参阅[监控挂载尝试的成功与失败](#)。

有关安装和升级 **botocore** 的说明，请参阅 Github 上 `amazon-efs-utils` 自述文件中的[Installing botocore](#)。

升级 **stunnel**

使用 EFS 挂载帮助程序加密传输中数据需要 OpenSSL 版本 1.0.2 或更新版本，以及同时支持在线证书状态协议 (OCSP) 和证书主机名检查的 **stunnel** 版本。EFS 挂载帮助程序使用 **stunnel** 程序提供 TLS 功能。请注意，某些版本的 Linux 不包含默认支持这些 TLS 功能的 **stunnel** 版本。在使用这些 Linux 发行版之一时，使用 TLS 挂载 EFS 文件系统将失败。

在安装 EFS 挂载帮助程序后，您可以按照以下说明升级您的系统的 **stunnel** 版本。

在亚马逊 Linux、亚马逊 Linux 2 和其他支持的 Linux 发行版上升级 **stunnel** (除了 [SLES 12](#))

1. 在 Web 浏览器中，转到 **stunnel** 下载页面 <https://stunnel.org/downloads.html>。
2. 找到以 `tar.gz` 格式提供的最新 **stunnel** 版本。记下该文件的名称，因为您在接下来的步骤中需要用到。
3. 在 Linux 客户端上打开一个终端，然后按提供的顺序运行以下命令。
 - a. 对于 RPM :

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

对于 DEB :

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

- b. ***latest-stunnel-version*** 替换为您之前在步骤 2 中记下的文件名。

```
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. 当前的 stunnel 软件包安装在 bin/stunnel 中。请使用以下命令删除该目录，以便可以安装新版本。

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

h. 安装最新版本：

```
sudo make install
```

i. 创建符号链接：

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

升级 macOS 上的 stunnel

- 打开 EC2 Mac 实例上的终端，然后运行以下命令以升级到最新版本的 stunnel。

```
brew upgrade stunnel
```

升级 SLES 12 的 stunnel

- 运行以下命令，并按照 zypper 软件包管理器的说明在运行 SLES12 的计算实例上升级 stunnel。

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

```
sudo zypper install -y stunnel
```

在安装某个具有所需功能的 stunnel 版本后，可以使用 TLS 和 Amazon EFS 建议的设置挂载文件系统。

解决安装 stunnel 时遇到的问题

如果您无法安装 stunnel，请尝试禁用证书主机名检查。此外，请启用在线证书状态协议（OCSP）以提供尽可能强的安全性。

主题

- [禁用证书主机名检查](#)
- [启用在线证书状态协议](#)

禁用证书主机名检查

如果无法安装所需的依赖项，您可以选择在 Amazon EFS 挂载帮助程序配置中禁用证书主机名检查。我们建议您不要在生产环境中禁用此功能。要禁用证书主机名检查，请执行以下操作：

1. 通过 Secure Shell（SSH）访问您的 EC2 实例的终端，然后使用相应的用户名登录。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 使用所选的文本编辑器打开 `/etc/amazon/efs/efs-utils.conf` 文件。
3. 将 `stunnel_check_cert_hostname` 值设置为 `false`。
4. 保存对该文件的更改，然后关闭该文件。

有关使用传输中的数据加密的更多信息，请参阅[挂载 EFS 文件系统](#)。

启用在线证书状态协议

为了在无法从您的 VPC 访问 CA 时最大限度地提高文件系统的可用性，当您选择加密传输中数据时，默认情况下不启用在线证书状态协议（OCSP）。Amazon EFS 使用 [Amazon 证书颁发机构](#)（CA）来颁发和签署其 TLS 证书，CA 指示客户端使用 OCSP 来检查是否有被吊销的证书。必须可以通过 Internet 从 Virtual Private Cloud 访问 OCSP 终端节点，以检查证书的状态。在该服务中，Amazon EFS 持续监视证书状态，并颁发新证书以替换它检测到的任何已撤销证书。

为了提供最高安全性，您可以启用 OCSP，以便 Linux 客户端可以检查撤销的证书。OCSP 可防止恶意使用已撤销的证书，这种情况不太可能发生在您的 VPC 中。如果撤销 EFS TLS 证书，Amazon 将发布安全公告，并发布拒绝已撤销的证书的新版 EFS 挂载帮助程序。

在 Linux 客户端上启用 OCSP，以便将来都可以与 EFS 建立 TLS 连接

1. 打开 Linux 客户端上的一个终端。
2. 使用所选的文本编辑器打开 `/etc/amazon/efs/efs-utils.conf` 文件。
3. 将 `stunnel_check_cert_validity` 值设置为 `true`。
4. 保存对该文件的更改，然后关闭该文件。

在 `##` 命令中启用 OCSP

- 使用以下挂载命令在挂载文件系统时启用 OCSP。

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

启用 FIPS 模式

如果您的操作系统在挂载文件系统时使用符合联邦信息处理标准 (FIPS) 的端点，则必须在 Amazon EFS 客户端中启用 FIPS 模式。要启用 FIPS 模式，需要修改操作系统上的 `efs-utils.conf` 文件。

Note

FIPS 模式要求已安装的 OpenSSL 版本必须使用 FIPS 进行编译。有关如何使用 FIPS 配置 OpenSSL 的更多信息，请参阅 [OpenSSL FIPS 自述文件](#)。

在 Amazon EFS 客户端中启用 FIPS 模式

1. 通过 Secure Shell (SSH) 访问您的 Amazon EC2 实例的终端，然后使用相应的用户名登录。有关更多信息，请参阅《Amazon Elastic Compute Cloud 用户指南》中的[连接到您的 EC2 实例](#)。
2. 使用所选的文本编辑器打开 `/etc/amazon/efs/efs-utils.conf` 文件。
3. 找到包含以下文本的行：

```
"fips_mode_enabled = false"
```

4. 将该文本更改为以下内容：

```
"fips_mode_enabled = true"
```

5. 保存更改。

创建和管理 EFS 资源

Amazon EFS 提供弹性的共享文件存储。您创建的文件系统支持来自多个 Amazon EC2 实例的并发读写权限。也可以从文件系统的创建地的所有可用区访问 AWS 区域 该文件系统。

您可以使用网络文件系统版本 4.0 和 4.1 协议 () 在基于亚马逊 VPC 的虚拟私有云 (VPC) 中的 EC2 实例上挂载 Amazon EFS 文件系统。NFSv4 有关更多信息，请参阅 [Amazon EFS 的工作原理](#)。

例如，假设您的 VPC 中启动了一个或多个 EC2 实例。现在您想要在这些实例上创建和使用一个文件系统。以下是在 VPC 中使用 Amazon EFS 文件系统时必须执行的典型步骤：

- 创建 Amazon EFS 文件系统 – 创建文件系统时，我们建议使用名称标签。名称标签值显示在控制台中，便于识别文件系统。您也可以向文件系统添加其他可选标签。
- 为文件系统创建挂载目标 – 为了在 VPC 中访问文件系统和将文件系统挂载到 Amazon EC2 实例上，您必须在 VPC 子网中创建挂载目标。
- 创建安全组 – Amazon EC2 实例和挂载目标都必须具有关联的安全组。这些安全组充当虚拟防火墙，控制它们之间的流量。您可以使用与挂载目标关联的安全组来控制文件系统的入站流量。为此，请向挂载目标安全组添加一条入站规则，允许从特定 EC2 实例进行访问。然后，您可以将文件系统仅挂载到该 EC2 实例上。

主题

- [实现摘要](#)
- [资源 IDs](#)
- [创建令牌和幂等性](#)
- [创建 EFS 文件系统](#)
- [删除 EFS 文件系统](#)
- [创建文件系统策略](#)
- [创建接入点](#)
- [删除接入点](#)
- [标记 EFS 资源](#)
- [教程：创建可写的每用户子目录](#)

实现摘要

在 Amazon EFS 中，文件系统是主要资源。每个文件系统都有许多属性，例如，ID、创建令牌、创建时间、以字节为单位的文件系统大小、为文件系统创建的挂载目标的数量，以及文件系统生命周期策略。

Amazon EFS 还支持使用其他资源来配置主要资源。这些包括挂载目标和访问点：

- 挂载目标 – 要访问您的文件系统，您必须在 VPC 中创建挂载目标。每个挂载目标具有以下属性：挂载目标 ID、在其中创建挂载目标的子网 ID、为其创建挂载目标的文件系统 ID、可以挂载文件系统的 IP 地址、VPC 安全组以及挂载目标状态。您可以在 mount 命令中使用 IP 地址或 DNS 名称。

每个文件系统都具有以下形式的 DNS 名称。

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

您可以在 mount 命令中指定此 DNS 名称以挂载 Amazon EFS 文件系统。假设您在 EC2 实例或本地服务器上的主目录中创建 efs-mount-point 子目录。然后，您可以使用挂载命令来挂载文件系统。例如，在 Amazon Linux AMI 中，您可以使用以下 mount 命令。

```
$ 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
```

有关更多信息，请参阅 [管理挂载目标](#)。

- 接入点 – 接入点将操作系统用户、组和文件系统路径应用于使用接入点发出的任何文件系统请求。访问点的操作系统用户和组覆盖 NFS 客户端提供的任何身份信息。文件系统路径作为访问点的根目录向客户端公开。这可确保每个应用程序在访问共享的基于文件的数据集时始终使用正确的操作系统身份和正确的目录。使用访问点的应用程序只能访问其自己的目录及之下目录中的数据。有关更多信息，请参阅 [使用接入点工作](#)。

挂载目标和标签是与文件系统关联的子资源。您只能在现有文件系统的上下文中创建它们。

Amazon EFS 为您提供 API 操作来创建和管理这些资源。除了每个资源的创建和删除操作外，Amazon EFS 还支持描述操作，此操作使您能够检索资源信息。可使用以下选项创建和管理这些资源：

- 使用 Amazon EFS 控制台 – 有关示例，请参阅 [开始使用](#)。

- 使用 Amazon EFS 命令行界面 (CLI) – 有关示例，请参阅 [教程：创建 EFS 文件系统并将其挂载到 EC2 实例上 AWS CLI](#)。
- 也可以通过编程方式管理这些资源，如下所示：
 - 使用 AWS SDKs — 通过封装底层 Amazon EFS API 来 AWS SDKs 简化您的编程任务。软件开发工具包客户端还通过使用您提供的访问密钥验证您的请求。有关更多信息，请参阅[示例代码和库](#)。
 - 直接从您的应用程序调用 Amazon EFS API — 如果您由于某种 SDKs 原因无法使用，则可以直接从您的应用程序调用 Amazon EFS API。但是，使用该选项时您需要编写必需的代码来验证请求。有关 Amazon EFS API 的更多信息，请参阅 [Amazon EFS API](#)。

资源 IDs

Amazon EFS 在创建所有 EFS 资源时会为其分配唯一的资源标识符 (IDs)。所有 EFS 资源均 IDs 由资源标识符以及数字 0—9 和小写字母 a—f 的组合组成。

2021 年 10 月之前，IDs 分配给新创建的文件系统和挂载目标资源的连字符后使用 8 个字符（例如）。fs-12345678 从 2021 年 5 月到 2021 年 10 月，我们将这些资源类型更改为在连字符后面使用 17 个字符（例如，fs-1234567890abcdef0）。IDs 根据您创建账户的时间，您的文件系统和挂载目标资源可能短一些 IDs，但这些类型的任何新资源都会收到更长的时间 IDs。资源 ID 永远不会改变。

创建令牌和幂等性

幂等性可确保 API 请求仅完成一次。对于幂等性请求，如果原始请求成功完成，则后续请求不会产生额外影响。这对于防止在您与 Amazon EFS API 交互时创建重复任务非常有用。

Amazon EFS API 支持客户端请求令牌的幂等性。客户端请求令牌是您在发出创建作业请求时指定的唯一字符串。

客户端请求令牌可以是包含最多 64 个 ASCII 字符的任意字符串。如果您在成功请求后的一分钟内重复使用客户端请求令牌，API 将返回原始请求的作业详细信息。

如果您使用控制台，它会替您生成令牌。如果您使用控制台中的自定义创建流程，为您生成的创建令牌具有以下格式：

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

如果您使用“快速创建”来创建具有服务建议设置的文件系统，创建令牌的格式如下：

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

创建 EFS 文件系统

您可以按照[入门练习](#)中所述，在创建新的 EC2 启动实例时创建文件系统。但是，您也可以使用 Amazon EFS 控制台、AWS Command Line Interface (AWS CLI) 或 Amazon EFS API 创建文件系统。

通过 Amazon EFS 控制台创建文件系统时，根据您是希望文件系统使用推荐设置还是希望自定义设置，您可以选择不同的选项。

- 您可以使用“快速创建”功能，快速创建采用以下推荐设置的文件系统。
 - 区域可用性
 - 生命周期策略：文件系统创建 30 天后将其自动转换为 EFS 不频繁访问（IA）存储，90 天后将其转换为 EFS 存档存储，且不转换为 EFS 标准存储
 - 启用静态数据加密
 - 弹性吞吐量模式
 - 通用性能模式
 - 使用指定子网 IPv4 上的可用文件系统创建的每个可用区域 AWS 区域中配置的挂载目标
- 您可以使用“自定义”功能，根据自己选择的设置创建文件系统。

有关文件系统设置及推荐值的列表，请参阅[文件系统的配置选项](#)。

创建文件系统所需的 IAM 权限

要创建 EFS 资源，例如文件系统和接入点，您必须拥有相应的 API 操作和资源的 AWS Identity and Access Management (IAM) 权限。

创建 IAM 用户，并使用用户策略向他们授予执行 Amazon EFS 操作的权限。也可以使用角色来授予跨账户权限。Amazon Elastic File System 还使用一个 IAM 服务相关角色，其中包括 AWS 服务代表您呼叫他人所需的权限。有关管理 API 操作权限的更多信息，请参阅[Amazon EFS 的身份和访问管理](#)。

文件系统的配置选项

EFS 文件系统将按下表所列设置进行配置。

- 如果您使用“快速创建”功能创建文件系统，该文件系统会采用配置项对应的推荐值完成创建。

- 如果您使用“自定义”功能创建自定义文件系统，您可以更改配置项的推荐值。

EFS 文件系统设置

设置	说明
文件系统类型	推荐设置：区域性 文件系统类型决定了 EFS 文件系统在 AWS 区域中存储数据的可用性和持久性。 • 区域性文件系统可跨 AWS 区域中的所有可用区以冗余方式存储数据和元数据。您还可以在 AWS 区域中的每个可用区中创建挂载目标。区域性提供最高级别的可用性和持久性。 • 单区文件系统可将数据和元数据以冗余方式存储到单个可用区中。单区文件系统只能有一个挂载目标。此挂载目标必须位于创建文件系统的同一可用区中。  Note 单区域文件系统仅适用于某些可用区。 有关文件系统类型的更多信息，请参阅EFS 文件系统的可用性和持久性。
生命周期管理	推荐设置：30 天后转换为 IA 存储，90 天后转换为存档存储。不要转换为标准存储。 生命周期管理使用生命周期策略，根据访问模式自动将文件移入和移出成本较低的不频繁访问（IA）存储类别。使用创建文件系统时 AWS 管理控制台，文件系统的生命周期策略将使用以下默认设置进行配置：

设置	说明
	• 转换为 IA 设置为自上次访问后的 30 天。 • 转换为归档设置为自上次访问后的 90 天。 • 转换为标准设置为无。 使用 Amazon EFS API 或创建文件系统时 AWS SDKs，不能同时设置生命周期策略。AWS CLI 必须等到文件系统创建完毕，然后才能使用 PutLifecycleConfiguration API 操作更新生命周期策略。 有关生命周期管理的更多信息，请参阅 管理存储生命周期。
静态加密	推荐设置：已启用 默认情况下，Amazon EFS 使用您的 AWS Key Management Service (AWS KMS) <code>aws/elasticfilesystem</code> EFS 服务密钥 () 对静态数据进行加密。加密后，存储在文件系统上的所有数据及元数据均会被加密。创建 EFS 文件系统后，将无法更改其加密设置。若需将未加密文件系统转换为加密状态，必须创建新的文件系统。 有关 Amazon EFS 加密的更多信息，请参阅 Amazon EFS 中的数据加密。

设置	说明
吞吐量模式	推荐设置：弹性 可从以下吞吐量模式中进行选择： • 弹性 - 提供实时自动纵向扩展和缩减的吞吐量，以满足工作负载的性能需求。 • 预调配 - 指定您所需的吞吐量大小，与文件系统的大小无关。 • 突发 – 提供可随标准存储中的数据量而扩展的吞吐量。  Note 使用弹性和预调配吞吐量会产生额外费用。有关更多信息，请参阅 Amazon EFS 定价。 有关吞吐量模式的更多信息，请参阅吞吐量模式。

设置	说明
性能模式	推荐设置：通用 通用性能模式的每次操作延迟最低，建议用于所有文件系统。 最大 I/O 是上一代性能类型，专为高度并行化的工作负载而设计，与通用模式相比，可以容忍更高的延迟。One Zone 文件系统或使用弹性吞吐量的文件系统不支持最大 I/O 模式。  Important 由于最大 I/O 的每次操作延迟较高，因此我们建议对所有文件系统使用通用性能模式。 有关更多信息，请参阅 性能模式。

设置	说明
网络访问	推荐设置：在提供文件系统的每个可用区中创建的挂载目标 挂载目标设置如下： • VPC - 一次只能为一个 VPC 中的文件系统创建挂载目标。默认情况下，选择 EC2 实例的默认子网。若需从其他 VPC 访问文件系统，请更改 VPC。 如果要在创建文件系统之后更改 VPC，则需要先从当前 VPC 中删除挂载目标。有关更多信息，请参阅 更改挂载目标 VPC。 • 可用区和子网 ID - 您可以更改挂载目标的可用区或子网。 • IP 地址类型-默认情况下，Amazon EFS 使用一个 IPv4 地址来创建挂载目标。您也可以选择 IPv6 仅支持 IPv6 地址，或者选择双栈同时支持 IPv4 和 IPv6 地址。  Note IP 地址类型必须与子网的 IP 类型相匹配。此外，IP 地址类型会覆盖子网的 IP 寻址属性。例如，如果 IP 地址类型为 IPv4-only 且您的子网启用了 IPv6 寻址属性，则在子网中创建的网络接口将接收子网范围内 IPv4 的地址。有关更多信息，请参阅修改子网的 IP 寻址属性。 • IP 地址 - 默认情况下，Amazon EFS 会在指定子网的可用地址处创建挂载目标。或者，您也可以为挂载目标指定 IP 地址。

设置	说明
	在创建挂载目标后，您将无法更改其 IP 地址。要更改 IP 地址，必须删除挂载目标并使用新地址创建一个新目标。 安全组 - 默认使用分配给 VPC 目标的安全组。您可以添加或删除安全组。有关安全组的更多信息，请参阅 更改挂载目标安全组。 当您使用 AWS CLI、Amazon EFS API 或创建文件系统时 AWS SDKs，不能同时创建挂载目标。必须等到文件系统创建完毕，然后才能使用 CreateMountTarget API 操作创建挂载目标。 有关挂载目标的更多信息，请参阅管理挂载目标。

使用控制台快速创建

使用 Amazon EFS 控制台创建带推荐设置的 Amazon EFS 文件系统。如果要使用自定义配置创建文件系统，请参阅[使用控制台以自定义方式创建](#)。

快速创建具有推荐设置的 Amazon EFS 文件系统

1. 登录 AWS 管理控制台 并打开 Amazon EFS 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 选择创建文件系统以打开创建文件系统对话框。
3. （可选）输入文件系统的名称。
4. 对于虚拟私有云（VPC），请选择您的 VPC，或者将其设置为默认 VPC。
5. 选择创建以创建使用以下服务推荐设置的文件系统：
 - 区域可用性
 - 生命周期策略：文件系统创建 30 天后将其自动转换为 EFS 不频繁访问（IA）存储，90 天后将其转换为 EFS 存档存储，且不转换为 EFS 标准存储
 - 启用静态数据加密

- 弹性吞吐量模式
- 通用性能模式
- 使用指定子网 IPv4 上的可用挂载 AWS 区域 在创建文件系统的每个可用区中配置的目标

将出现文件系统页面，顶部有一个横幅，显示您创建的文件系统的状态。当文件系统可用时，横幅中会显示访问文件系统详细信息页面的链接。

有关文件系统的更多信息，请参阅[了解文件系统状态](#)。

使用控制台以自定义方式创建

本节介绍使用 Amazon EFS 控制台，而不是使用服务推荐的设置创建具有自定义设置的 EFS 文件系统的过程。有关使用推荐设置创建文件系统的更多信息，请参阅[使用控制台快速创建](#)。

使用控制台创建具有自定义设置的 EFS 文件系统包括四个步骤：

- 步骤 1 – 配置常规文件系统设置，包括存储类和吞吐量模式。
- 步骤 2 – 配置文件系统网络设置，包括虚拟私有云（VPC）和挂载目标。对于每个挂载目标，设置可用区、子网、IP 地址和安全组。
- 步骤 3 –（可选）创建文件系统策略以控制 NFS 客户端对文件系统的访问。
- 步骤 4 – 查看文件系统设置，进行任何更改，然后创建文件系统。

步骤 1：配置文件系统设置

1. 登录 AWS 管理控制台 并打开 Amazon EFS 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 选择创建文件系统以打开创建文件系统对话框。
3. 选择自定义以创建自定义文件系统，而不是使用推荐设置来创建文件系统。将打开文件系统设置页面。
4. 对于常规设置，执行以下操作。
 - a. （可选）输入文件系统的名称。
 - b. 对于文件系统类型，默认选择区域性。如果要创建在单个可用区内以冗余方式存储文件系统数据和元数据的文件系统，请选择单区。如果选择单区，请选择要在其中创建文件系统的可用区，或者保留默认值。

- c. 对于生命周期管理，如有必要，请更改生命周期策略。
- 转换为 IA - 根据自上次在标准存储中访问文件以来的时间，选择何时将文件转换为不频繁访问 (IA) 存储类。
 - 转换为归档 - 根据自上次在标准存储中访问文件以来的时间，选择何时将文件转换为存档存储类。
 - 转换为标准 - 选择是否将文件系统转换为存储类。

有关生命周期策略的更多信息，请参阅[管理存储生命周期](#)。

- d. 默认情况下，Amazon EFS 使用您的 AWS Key Management Service (AWS KMS) 的 `aws/elasticfilesystem` EFS 服务密钥 () 对静态数据进行加密。要选择其他 KMS 密钥用于加密，请展开自定义加密设置，然后从列表中选择密钥。或者，输入要使用的 KMS 密钥的 KMS 密钥 ID 或 Amazon 资源名称 (ARN)。

如果您需要创建新密钥，请选择创建 AWS KMS key 以启动 AWS KMS 控制台并创建新密钥。

可以通过清除复选框关闭对静态数据的加密。

创建文件系统后，将无法更改加密设置。有关更多信息，请参阅 [Amazon EFS 中的数据加密](#)。

5. 对于性能设置，请执行以下操作：

- a. 对于吞吐量模式，默认选择弹性模式。
- 要使用预调配吞吐量模式，请选择预调配，然后在预调配吞吐量 (MiB/s) 中输入要为文件系统请求预调配的吞吐量。最大读取吞吐量显示为您输入的吞吐量的三倍。
 - 要使用突发吞吐量，请选择突发。

选择吞吐量模式后，将显示文件系统每月成本的估计值。文件系统可用后，您可以更改吞吐量模式。

有关根据性能需求选择正确的吞吐量模式的更多信息，请参阅[吞吐量模式](#)。

- b. 对于性能模式，默认为通用。要更改性能模式，请展开其他设置，然后选择最大 I/O。

文件系统可用后，无法更改性能模式。有关更多信息，请参阅 [性能模式](#)。

 Important

由于最大 I/O 的每次操作延迟较高，因此我们建议对所有文件系统使用通用性能模式。

6. (可选) 向文件系统添加标签键值对。
7. 选择下一步以配置文件系统的网络访问。

步骤 2：配置网络访问

在步骤 2 中，您将配置文件系统的网络设置，包括 VPC 和挂载目标。

1. 选择希望 EC2 实例连接到文件系统的虚拟私有云 (VPC)。有关更多信息，请参阅 [管理挂载目标](#)。
2. 对于挂载目标，可以为文件系统创建一个或多个挂载目标。对于每个挂载目标，请设置以下属性：
 - 可用区 – 默认情况下，在 AWS 区域中的每个可用区中配置挂载目标。如果不想在特定可用区中安装挂载目标，请选择移除以删除该区域的挂载目标。在计划访问文件系统的每个可用区都创建一个挂载目标，此操作不收取费用。
 - 子网 ID – 从可用区中的可用子网中进行选择。已预先选择默认子网。
 - IP 地址类型-选择 IPv4 仅支持 IPv4 地址，IPv6 仅支持 IPv6 地址，或选择双栈以同时支持 IPv4 和 IPv6 地址。
 - IPv4 或 IPv6 地址-如果您知道要放置挂载目标的 IP 地址，则在与 IP 地址类型匹配的 IP 地址框中输入该地址。如果未指定值，Amazon EFS 会从指定的子网中选择未使用的 IP 地址。
 - 安全组 - 默认情况下，Amazon EFS 为 VPC 选择默认安全组。要更改安全组，请删除分配的组，然后从选择安全组列表中进行选择。可以为挂载目标指定一个或多个安全组。有关更多信息，请参阅 [使用 VPC 安全组](#)。
3. 选择添加挂载目标，为没有挂载目标的可用区创建挂载目标。如果为每个可用区配置了挂载目标，此选项不可用。
4. 选择下一步以设置文件系统策略。

步骤 3：创建文件系统策略 (可选)

也可以为文件系统创建文件系统策略。EFS 文件系统策略是一种 IAM 资源策略，用于控制 NFS 客户端对文件系统的访问。有关更多信息，请参阅 [使用 IAM 控制文件系统访问](#)。

1. 在策略选项中，可以选择预配置文件系统策略的任意组合：
 - 默认阻止根访问 – 此选项可从允许的 EFS 操作集中移除 ClientRootAccess。
 - 默认强制执行只读访问 – 此选项可从允许的 EFS 操作集中移除 ClientWriteAccess。
 - 防止匿名访问 – 此选项可从允许的 EFS 操作集中移除 ClientMount。
 - 对所有客户端强制执行传输中加密 – 此选项拒绝访问未加密的客户端。

选择预配置的策略时，策略 JSON 对象将显示在策略编辑器窗格中。

2. 使用授予额外权限向其他 IAM 委托人（包括另一个 AWS 账户）授予文件系统权限。选择添加，然后输入要向其授予权限的实体的主体 ARN。然后选择要授予的权限。其他权限将显示在策略编辑器中。
3. 可以使用策略编辑器自定义预配置策略或创建自己的文件系统策略。使用编辑器时，预配置策略选项将不可用。要清除当前文件系统策略并开始创建新策略，请选择清除。
4. 选择下一步以查看并创建文件系统。

步骤 4：审核并创建

1. 查看每个文件系统配置组。此时，可以通过选择编辑对每个组进行更改。
2. 选择创建文件系统来创建文件系统，并返回到文件系统页面。

顶部的横幅显示正在创建新文件系统。当文件系统可用时，横幅中会显示访问新文件系统详细信息页面的链接。

使用创建 AWS CLI

使用时 AWS CLI，您可以按顺序创建这些资源。首先，创建一个文件系统。然后，您可以使用相应的 AWS CLI 命令为文件系统创建挂载目标和任何其他可选标记。

以下示例使用 `adminuser` 作为 `--profile` 参数值。必须使用适当的用户配置文件来提供凭证。有关信息，请参阅《AWS Command Line Interface User Guide》中的 [Prerequisites to use the AWS CLI](#)。

- 要创建启用了自动备份功能的加密文件系统，请使用 Amazon EFS `create-file-system` CLI 命令（相应的操作是 [CreateFileSystem](#)），如下所示。

```
aws efs create-file-system \  
--creation-token creation-token \  
--encrypted \  

```



```
--backup \  
--performance-mode generalPurpose \  
--throughput-mode elastic \  
--region aws-region \  
--tags Key=key,Value=value Key=key1,Value=value1 \  
--profile adminuser
```

例如，以下 `create-file-system` 命令将使用弹性吞吐量在 us-west-2 AWS 区域中创建一个文件系统。该命令指定 `MyFirstFS` 作为创建令牌。有关可在 AWS 区域 何处创建 Amazon EFS 文件系统的列表，请参阅中的 [Amazon EFS 终端节点和配额](#)[Amazon Web Services 一般参考](#)。

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

成功创建文件系统后，Amazon EFS 会以 JSON 形式返回文件系统描述，如以下示例所示。

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

```
    "Key": "Name",
    "Value": "Test File System"
  }
]
```

- 以下示例将使用 `availability-zone-name` 属性在 `us-west-2a` 可用区中创建使用突增吞吐量的文件系统。

```
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
```

成功创建文件系统后，Amazon EFS 会以 JSON 形式返回文件系统描述，如以下示例所示。

```
{
  "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 还提供 `describe-file-systems` CLI 命令（相应 API 操作为 [DescribeFileSystems](#)），可以使用此命令在您的账户中检索文件系统列表，如下所示。

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

Amazon EFS 会返回您在指定区域中 AWS 账户 创建的文件系统的列表。

删除 EFS 文件系统

文件系统删除是一种无法撤消的破坏性操作。您将丢失文件系统及其包含的任何数据。从文件系统中删除的任何数据将会丢失，而无法还原该数据。当用户删除文件系统中的数据时，这些数据立即将呈现为不可用。EFS 将以最终的方式强制覆盖数据。

Note

无法删除属于复制配置的文件系统。必须先删除复制配置。有关更多信息，请参阅 [删除复制配置](#)。

Important

应始终在删除之前卸载文件系统。

使用控制台

删除文件系统

1. 打开 Amazon Elastic File System 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 选择要从文件系统页面中删除的文件系统。

3. 选择删除。
4. 在删除文件系统对话框中，输入显示的文件系统 ID，然后选择确认以确认删除。

控制台简化了文件系统删除操作。首先，它将删除关联的挂载目标，然后删除文件系统。

使用 AWS CLI

在使用 AWS CLI 命令删除文件系统之前，必须先删除为该文件系统创建的所有装载目标和接入点。

有关 AWS CLI 命令示例，请参见[步骤 4：清除](#)。

创建文件系统策略

可以使用 Amazon EFS 控制台或 AWS CLI 创建文件系统策略。您也可以直接使用 AWS SDKs 或 Amazon EFS API，以编程方式创建文件系统策略。EFS 文件系统策略有 2 万个字符的限制。有关使用 EFS 文件系统策略的更多信息和示例，请参阅[使用 IAM 控制文件系统访问](#)。

Note

Amazon EFS 文件系统策略更改可能需要几分钟才能生效。

使用控制台

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 选择 File Systems (文件系统)。
3. 在 File systems (文件系统) 页面上，选择要为其编辑或创建文件系统策略的文件系统。
4. 选择文件系统策略，然后选择编辑。
5. 在策略选项中，可以选择预配置文件系统策略的任意组合：
 - 默认阻止根访问 – 此选项可从允许的 EFS 操作集中移除 ClientRootAccess。
 - 默认强制执行只读访问 – 此选项可从允许的 EFS 操作集中移除 ClientWriteAccess。
 - 防止匿名访问 – 此选项可从允许的 EFS 操作集中移除 ClientMount。
 - 对所有客户端强制执行传输中加密 – 此选项拒绝访问未加密的客户端。

选择预配置的策略时，策略 JSON 对象将显示在策略编辑器窗格中。

6. 使用授予额外权限向其他 IAM 委托人（包括另一个 AWS 账户）授予文件系统权限。选择添加，然后输入要向其授予权限的实体的主体 ARN。然后选择要授予的权限。其他权限将显示在策略编辑器中。
7. 可以使用策略编辑器自定义预配置策略或创建自己的文件系统策略。使用编辑器时，预配置策略选项将不可用。要清除当前文件系统策略并开始创建新策略，请选择清除。

清除编辑器后，预配置策略将再次可用。

8. 编辑完策略后，选择保存。

使用 AWS CLI

在以下示例中，[put-file-system-policy](#) CLI 命令创建了一个文件系统策略，该策略允许对 EFS 文件系统进行指定的 AWS 账户 只读访问。等效的 API 命令是 [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"
      }
    }
  ]
}'
```

创建接入点

您可以使用 AWS 管理控制台、AWS Command Line Interface (AWS CLI) 和 Amazon EFS API 创建和删除 Amazon EFS 接入点，以及 SDKs。接入点一经创建就无法进行修改。一个文件系统最多可以有 10,000 个接入点。

 Note

如果快速连续发送多个在同一文件系统上创建接入点的请求，并且文件系统接近接入点上限，则可能会遇到对这些请求的节流响应。这是为了确保文件系统不超过规定的接入点配额。

有关 EFS 访问点的更多信息，请参阅[使用接入点工作](#)。

使用控制台

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 选择接入点以打开接入点窗口。
3. 选择创建接入点以显示创建接入点页面。

也可以通过选择文件系统来打开创建接入点页面。选择文件系统的名称或文件系统 ID，然后选择接入点和创建接入点，为该文件系统创建接入点。

a. 在详细信息面板中输入以下信息：

- 文件系统 – 输入文件系统名称或 ID，然后选择匹配的文件系统。也可以从选择输入字段时显示的列表中选择文件系统。
- (可选) 名称 – 输入接入点的名称。
- (可选) 根目录路径 – 可以为接入点指定根目录；默认接入点根目录为 /。要输入根目录路径，请使用格式 /foo/bar。有关更多信息，请参阅[使用接入点强制执行根目录](#)。

b. (可选) 在 POSIX 用户面板中，可以指定完整的 POSIX 身份，以用于使用接入点的 NFS 客户端对所有文件操作强制执行用户和组信息。有关更多信息，请参阅[使用接入点强制执行用户身份](#)。

- 用户 ID – 输入用户的数字 POSIX 用户 ID。
- 组 ID – 输入用户的数字 POSIX 组 ID。
- 辅助组 IDs-输入以逗号分隔的可选辅助组列表。IDs

c. (可选) 对于根目录创建权限，可以指定 Amazon EFS 创建根目录路径时使用的权限 (如果已指定，并且根目录尚不存在)。有关更多信息，请参阅[使用接入点强制执行根目录](#)。

 Note

如果未指定任何根目录所有权和权限，并且根目录尚不存在，EFS 将不会创建根目录。使用接入点挂载文件系统的任何尝试都将失败。

- 拥有者用户 ID – 输入用作根目录所有者的数字 POSIX 用户 ID。
- 拥有者组 ID – 输入用作根目录所有者组的数字 POSIX 组 ID。
- 权限 – 输入目录的 Unix 模式。一个常见配置是 755。确保为接入点用户设置了执行位，以便他们能够执行装载操作。

4. 选择创建接入点以使用此配置创建接入点。

使用 AWS CLI

在以下示例中，create-access-point CLI 命令将为 EFS 文件系统创建接入点。等效的 API 命令是 [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
```

如果请求成功，CLI 将使用接入点描述进行响应。

```
{
  "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,
    "OwnerUid": 11,
    "Permissions": "775"
  },
  "Path": "/efs/mobileapp/east",
},
"Tags": []
}
```

Note

如果快速连续发送多个在同一文件系统上创建接入点的请求，并且文件系统接近接入点上限，则可能会遇到对这些请求的节流响应。这是为了确保文件系统不超过规定的接入点配额。

删除接入点

删除接入点后，任何使用该接入点的客户端都将无法访问为其配置的 Amazon EFS 文件系统。

使用控制台

1. 打开 Amazon Elastic File System 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 在左侧导航窗格中，选择接入点以打开接入点页面。
3. 选择要删除的接入点。
4. 选择删除。
5. 选择确认以确认操作并删除接入点。

使用 AWS CLI

在以下示例中，delete-access-point CLI 命令将删除指定的接入点。等效的 API 命令是 [DeleteAccessPoint](#)。如果此命令成功，该服务会返回带有空 HTTP 正文的 HTTP 204 响应。

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


标记 EFS 资源

为了帮助您管理 EFS 资源，可以通过标签的形式为每个资源分配您自己的元数据。使用标签，您可以按不同的方式对 AWS 资源进行分类，例如按用途、所有者或环境进行分类。当您具有很多相同类型的资源时，这种分类会很有用 – 您可以根据分配的标签快速识别特定的资源。本主题介绍标签并演示如何创建标签。

标签基本知识

标签是您分配给 AWS 资源的标签。每个标签都包含定义的一个键和一个可选值。

标签使您能够以不同的方式对 AWS 资源进行分类，例如按用途、所有者或环境进行分类。例如，您可以为账户中的 Amazon EFS 文件系统定义一组标签，以帮助跟踪每个文件系统的所有者。

我们建议您针对每类资源设计一组标签，以满足您的需要。使用一组连续的标签键，管理资源时会更加轻松。您可以根据添加的标签搜索和筛选资源。

标签对 Amazon EFS 没有任何语义意义，应严格按字符串进行解析。同时，标签不会自动分配至您的资源。您可以修改标签的密钥和值，还可以随时删除资源的标签。您可以将标签的值设为空的字符串，但是不能将其设为空值。如果您添加的标签的值与该实例上现有标签的值相同，新的值就会覆盖旧值。如果您删除资源，则该资源的所有标签也会被删除。

标签限制

下面是适用于 标签的基本限制：

- 每个资源的标签数上限：50
- 对于每个资源，每个标签键都必须是唯一的，每个标签键只能有一个值。
- 最大键长度：128 个 Unicode 字符（采用 UTF-8 格式）
- 最大值长度：256 个 Unicode 字符（采用 UTF-8 格式）
- 虽然 Amazon EFS 允许在其标签中使用任何字符，但其他服务具有更严格的限制。允许在不同的服务中使用的字符包括：可以使用 UTF-8 表示的字母、数字和空格以及以下字符：`+ - = . _ : / @`。
- 标签键和值区分大小写。
- 该 `aws:` 前缀已保留供 AWS 使用。如果某个标签具有带有此标签键，则您无法编辑该标签的键或值。具有 `aws:` 前缀的标签不计入每个资源的标签数限制。

不能仅依据标签更新或删除资源，必须指定资源标识符。例如，要删除使用名为 DeleteMe 的标签键标记的文件系统，必须将 DeleteFileSystem 操作与文件系统的资源标识符（例如 fs-1234567890abcdef0）结合使用。

为公有或共享资源添加标签时，所分配的标签仅对您的 AWS 账户可用。没有其他人 AWS 账户 可以访问这些标签。要对共享资源进行基于标签的访问控制，每个共享资源都 AWS 账户 必须分配自己的标签集来控制对资源的访问权限。

可以为 Amazon EFS 文件系统和接入点资源添加标签。

使用标签进行访问控制

可以使用标签来控制对 Amazon EFS 资源的访问，并实现基于属性的访问权限控制 (ABAC)。

Note

复制不支持将标签用于基于属性的访问权限控制 (ABAC)。

标记资源

可以标记账户中已存在的 Amazon EFS 文件系统和接入点资源。

使用控制台

- 可以使用 Amazon EFS 控制台中资源详细信息屏幕上的标签选项卡将标签应用于现有资源。在 Amazon EFS 控制台中，可以在创建资源时为其指定标签。例如，可以添加具有键 Name 和指定值的标签。

在大多数情况下，控制台会在资源创建后（而不是在资源创建期间）立即应用标签。尽管控制台会根据 Name 标签对资源进行组织，但此标签对于 Amazon EFS 服务没有任何语义意义。

使用 AWS CLI

- 如果您使用的是 Amazon EFS API AWS CLI、或 AWS 软件开发工具包，则可以使用 TagResource EFS API 操作将标签应用于现有资源。此外，某些资源创建操作让您可以在创建资源时为其指定标签。

下表列出了用于管理标签的 AWS CLI 命令以及等效的 Amazon EFS API 操作。

CLI 命令	说明	等效的 API 操作
tag-resource	添加新标签或更新现有标签	TagResource
list-tags-for-resource	检索现有标签	ListTagsForResource
untag-resource	删除现有标签	UntagResource

教程：创建可写的每用户子目录

在您创建 EFS 文件系统并将其本地安装到您的 Amazon EC2 (EC2) 实例上之后，它会显示一个名为的 *file system root* 空目录。此文件系统根目录的一个常见使用案例是，为您在 EC2 实例上创建的每个用户创建一个“可写”子目录，并将该子目录挂载到用户的主目录上。用户在其主目录中创建的所有文件和子目录随后都会在 EFS 文件系统上创建。

Note

您可以按照[开始使用](#)练习创建一个 EFS 文件系统并将其挂载到 EC2 实例上。

在以下步骤中，您将创建一个用户，为该用户创建一个子目录，将用户设为该子目录的所有者，然后将 Amazon EFS 子目录挂载到该用户的主目录。

1. 创建用户 mike：

- 登录到您的 EC2 实例。使用根特权（这里使用 `sudo` 命令），创建用户并分配密码。

例如，以下命令创建用户 mike。

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

还会为该用户创建主目录。例如 `/home/mike`。

2. 在下 *EFsroot* 为用户创建一个子目录。

例如，以下命令在 mike 下 *EFsroot* 创建子目录。

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

你需要`EFSroot`用你的本地目录名替换。

- 根用户和根组都是子目录的所有者 (可以使用 `ls -l` 命令来验证)。要为用户授予对此子目录的完全权限，可授予对目录的所有权。

例如：

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

- 使用 `mount` 命令将子目录挂载到用户的主目录中。

例如：

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

该`mount-target-DNS`地址标识远程 EFS 文件系统的根目录。

如果卸载此挂载目标，用户将无法访问目录，除非重新挂载，而这需要根权限。

挂载 EFS 文件系统

我们建议使用 EFS 挂载帮助程序挂载 EFS 文件系统。EFS 挂载帮助程序可帮助您在运行受支持发行版的 EC2 Linux 和 Mac 实例上挂载 EFS 文件系统。挂载帮助程序是您在安装 Amazon EFS 客户端 (amazon-efs-utils) 时安装的开源工具集的一部分。有关 Amazon EFS 客户端和受支持发行版的更多信息，请参阅[安装 Amazon EFS 客户端](#)。

或者，您也可以使用标准 Linux NFS 客户端手动挂载 EFS 文件系统。在 Amazon EC2 实例上挂载文件系统时，Amazon EFS 支持网络文件系统版本 4.0 和 4.1 (NFSv4) 协议。

此外，您可以使用 EFS 挂载帮助程序或 NFS 将 EC2 实例配置为在实例启动时自动挂载 EFS 文件系统。

主题

- [Linux 的挂载注意事项](#)
- [使用 EFS 挂载帮助程序挂载 EFS 文件系统](#)
- [使用网络文件系统挂载 EFS 文件系统](#)
- [自动挂载 EFS 文件系统](#)
- [卸载文件系统](#)
- [教程：创建 EFS 文件系统并将其挂载到 EC2 实例上 AWS CLI](#)
- [教程：使用本地 Linux 客户端挂载](#)
- [教程：挂载来自其它 VPC 的文件系统](#)
- [解决挂载问题](#)

Linux 的挂载注意事项

我们建议在 Linux 上使用以下挂载选项值：

- `rsize=1048576` – 设置 NFS 客户端对每个网络 READ 请求可以接收的最大数据字节数。在从 EFS 文件系统上的文件读取数据时应用此值。我们建议您尽可能使用最大的大小（最多 1048576），以避免性能下降。
- `wsize=1048576` – 设置 NFS 客户端对每个网络 WRITE 请求可以发送的最大数据字节数。在将数据写入到 EFS 文件系统上的文件时应用此值。我们建议您尽可能使用最大的大小（最多 1048576），以避免性能下降。

- **hard** – 设置 NFS 客户端在 NFS 请求超时之后的恢复行为，以便 NFS 请求在服务器回复之前无限次重试。建议您使用硬挂载选项 (hard) 以确保数据完整性。如果您使用 soft 挂载，请将 timeo 参数至少设置为 150 分秒 (15 秒)。这样做可尽量减少源自软挂载的数据损坏风险。
- **timeo=600** – 将超时值设置为 600 分秒 (60 秒)，这是 NFS 客户端在重试 NFS 请求之前等待响应的的时间。如果您必须更改超时参数 (timeo)，我们建议您使用至少为 150 的值，这相当于 15 秒。这样做有助于避免性能下降。
- **retrans=2** – 将 NFS 客户端重试请求的次数设置为 2，超过此次数之后将尝试进一步的恢复操作。
- **noresvport** – 告知 NFS 客户端在重新建立网络连接时，使用新的非特权传输控制协议 (TCP) 源端口。这样做有助于确保 EFS 文件系统在网络恢复事件后具有不间断的可用性。
- **_netdev** – /etc/fstab 中存在此选项时，将阻止客户端尝试挂载 EFS 文件系统，直到启用了网络。

一般而言，避免设置任何其他不同于默认值的挂载选项，这会导致性能降低和其他问题。如果您不使用前面的默认值，请注意以下事项：

- 更改读或写缓冲区大小或禁用属性缓存会导致性能下降。
- Amazon EFS 会忽略源端口。如果您更改 Amazon EFS 源端口，则不会有任何影响。
- Amazon EFS 不支持任何 Kerberos 安全变体。例如，下面的挂载命令将失败。

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

- 我们建议您使用其 DNS 名称挂载文件系统。Amazon EFS 将此名称解析为与您的 Amazon EC2 实例位于同一可用区的 EFS 挂载目标的 IP 地址，而无需调用外部资源。如果您在与 EC2 实例不同的可用区中使用挂载目标，则会导致跨可用区发送数据的标准 EC2 费用。可能还会面临更高的文件系统操作延迟。
- 有关更多挂载选项和默认设置的详细说明，请参阅 Linux 文档。

Note

如果需要启动您的 EC2 实例而不考虑挂载的 EFS 文件系统状态，请将 nofail 选项添加到 /etc/fstab 文件的文件系统条目中。

使用 EFS 挂载帮助程序挂载 EFS 文件系统

安装 Amazon EFS 客户端 (`amazon-efs-utils`) 后，您可以使用 EFS 挂载帮助程序，在运行[受支持发行版](#)的 EC2 Linux 和 Mac 实例上挂载 EFS 文件系统。Amazon EFS 不支持从 EC2 Windows 实例进行挂载。

Important

我们建议您始终使用最新版本的 `amazon-efs-utils`，以确保挂载成功。例如，2.3 `amazon-efs-utils` 之前的版本不支持使用 IPv6 地址进行装载。

在挂载文件系统时，挂载帮助程序定义了新的网络文件系统类型（称为 `efs`），它与 Linux 中的标准 `mount` 命令完全兼容。挂载帮助程序还支持在 EC2 Linux 实例上使用 `/etc/fstab` 配置文件中的条目，在实例启动时自动挂载 EFS 文件系统。

Warning

请在自动挂载文件系统时使用 `_netdev` 选项，它用于指定网络文件系统。如果缺少 `_netdev`，您的 EC2 实例可能会停止响应。出现该结果是因为，需要在计算实例启动其网络后初始化网络文件系统。有关更多信息，请参阅[自动挂载失败，并且实例没有响应](#)。

可以通过指定下列属性之一挂载文件系统：

- 文件系统 DNS 名称 – 如果使用文件系统 DNS 名称，但挂载帮助程序无法解析它，例如，在不同 VPC 中挂载文件系统时，它将回退为使用挂载目标 IP 地址。有关更多信息，请参阅[从其他 AWS 账户或 VPC 挂载 EFS 文件系统](#)。
- 文件系统 ID – 如果使用文件系统 ID，挂载帮助程序无需调用外部资源即可将其解析为挂载目标弹性网络接口 (ENI) 的本地 IP 地址。
- 挂载目标 IP 地址 – 可以使用其中一个文件系统挂载目标的 IP 地址。

可以在 Amazon EFS 控制台中找到所有这些属性的值。文件系统 DNS 名称可在附加屏幕中找到。

将传输中数据加密声明为 EFS 文件系统的挂载选项时，挂载帮助程序将初始化客户端 `stunnel` 进程和名为 `amazon-efs-mount-watchdog` 的监管进程。`amazon-efs-mount-watchdog` 进程监控 TLS 挂载的运行状况，并在首次通过 TLS 挂载 EFS 文件系统时自动启动。如果您的客户端在

Linux 上运行，则此过程由 `upstart` 或 `systemd` 管理，具体取决于您的 Linux 发行版。对于在支持的 macOS 上运行的客户端，它由管理 `launchd`。

Stunnel 是一种开源多用途网络中继。客户端 `stunnel` 进程侦听本地端口的入站流量，挂载帮助程序将 NFS 客户端流量重定向到该本地端口。

挂载帮助程序使用 TLS 1.2 版与您的文件系统进行通信。使用 TLS 需要具有证书，并且这些证书需要由受信任的 Amazon 证书颁发机构进行签名。有关加密的工作方式的更多信息，请参阅[Amazon EFS 中的数据加密](#)。

主题

- [EFS 挂载帮助程序使用的挂载设置](#)
- [获取支持日志](#)
- [使用 EFS 挂载帮助程序的先决条件](#)
- [使用 EFS 挂载帮助程序在 EC2 Linux 实例上挂载](#)
- [使用 EFS 挂载帮助程序在 EC2 Mac 实例上挂载](#)
- [从其他服务器装载 EFS 文件系统 AWS 区域](#)
- [挂载单区文件系统](#)
- [使用 IAM 授权挂载](#)
- [使用 EFS 接入点进行挂载](#)
- [将 EFS 挂载到多个 EC2 实例](#)
- [从其他 AWS 账户 或 VPC 挂载 EFS 文件系统](#)

EFS 挂载帮助程序使用的挂载设置

Amazon EFS 挂载帮助程序客户端使用以下针对 Amazon EFS 进行了优化的挂载选项：

- `nfsvers=4.1` – 在 EC2 Linux 实例上安装时使用
`nfsvers=4.0` – 在运行 macOS Big Sur、Monterey 和 Ventura 的支持的 EC2 Mac 实例上挂载时使用
- `rsz=1048576` – 将 NFS 客户端可以为每个网络 READ 请求接收的最大数据字节数设置为 1048576 (最大可用字节数)，以避免性能下降。
- `wsz=1048576` – 将 NFS 客户端可以为每个网络 WRITE 请求发送的最大数据字节数设置为 1048576 (最大可用字节数)，以避免性能下降。

- `hard` – 设置 NFS 客户端在 NFS 请求超时之后的恢复行为，以便 NFS 请求在服务器回复之前无限次重试，从而确保数据完整。
- `timeo=600` – 将 NFS 客户端在重试 NFS 请求之前用于等待响应的超时值设置为 600 分秒（60 秒），以避免性能下降。
- `retrans=2` – 将 NFS 客户端重试请求的次数设置为 2，超过此次数之后将尝试进一步的恢复操作。
- `noresvport` – 告知 NFS 客户端在重新建立网络连接时，使用新的非特权传输控制协议（TCP）源端口。使用 `noresvport` 选项来帮助确保 EFS 文件系统在重新连接或网络恢复事件后保持不间断的可用性。
- `mountport=2049` – 仅在运行 macOS Big Sur、Monterey 和 Ventura 的 EC2 Mac 实例上挂载时使用。

获取支持日志

EFS 挂载帮助程序具有 EFS 文件系统的内置日志记录。您可以与 Su AWS pport 共享这些日志以进行故障排除。可以使用 EFS 挂载帮助程序查找存储在客户端上的 `/var/log/amazon/efs` 中的日志。这些日志适用于 EFS 挂载帮助程序、`stunnel` 进程（默认禁用），以及监控 `stunnel` 进程的 `amazon-efs-mount-watchdog` 进程。

Note

`amazon-efs-mount-watchdog` 进程确保每个挂载的 `stunnel` 进程正在运行，并在卸载 EFS 文件系统后停止 `stunnel` 进程。如果 `stunnel` 进程由于某种原因意外终止，`watchdog` 进程将重新启动该进程。

可以在 `/etc/amazon/efs/efs-utils.conf` 中更改日志配置。要使任何日志更改生效，需要使用 EFS 挂载帮助程序卸载并重新挂载文件系统。挂载帮助程序和 `watchdog` 日志的日志容量限制为 20 MiB。默认情况下，将禁用 `stunnel` 进程的日志。

Important

您可以为 `stunnel` 进程日志启用日志记录。但是，启用 `stunnel` 日志可能会用完您的文件系统上的宝贵空间量。

使用 EFS 挂载帮助程序的先决条件

可以使用 Amazon EFS 挂载帮助程序在 Amazon EC2 实例上挂载 EFS 文件系统。要使用挂载帮助程序，您需要具有：

- 要挂载的文件系统的文件系统 ID – EFS 挂载帮助程序将文件系统 ID 解析为挂载目标弹性网络接口 (ENI) 的本地 IP 地址，无需调用外部资源。
- EFS 挂载目标 - 在虚拟私有云 (VPC) 中创建挂载目标。如果您使用服务推荐设置在控制台中创建文件系统，则会在文件系统所在的每个可用区 AWS 区域 中创建一个挂载目标。有关创建挂载目标的说明，请参阅[管理挂载目标](#)。

Note

我们建议您在新创建的挂载目标的生命周期状态变为可用后等待 60 秒，然后再通过 DNS 挂载文件系统。这种等待可以让 DNS 记录在文件系统 AWS 区域 所在的位置完全传播。

如果您在与 EC2 实例不同的可用区中使用挂载目标，则会导致跨可用区发送数据的标准 EC2 费用。可能还会面临更高的文件系统操作延迟。

- 从不同的可用区挂载单区文件系统：
 - 文件系统可用区的名称 – 如果您挂载的 EFS 单区文件系统位于与 EC2 实例不同的可用区中。
 - 挂载目标 DNS 名称 – 或者，也可以指定挂载目标的 DNS 名称，而不是可用区。
- 运行受支持的 Linux 或 macOS 发行版之一的 EC2 实例 - 支持使用挂载帮助程序来挂载文件系统的发行版如下：
 - Amazon Linux 2
 - Amazon Linux 2023
 - Amazon Linux 2017.09 及更高版本
 - macOS Big Sur
 - Red Hat Enterprise Linux (和衍生产品，如 CentOS) 7 和更新版本
 - Ubuntu 16.04 LTS 和更新版本

Note

运行 macOS Big Sur 的 EC2 Mac 实例仅支持 NFS 4.0。

- EFS 挂载帮助程序安装在 EC2 实例上 - 挂载帮助程序是实用程序 amazon-efs-utils 程序包中的一个工具。有关安装 amazon-efs-utils 的信息，请参阅[安装 Amazon EFS 客户端](#)。
- EC2 实例在 VPC 中 - 连接的 EC2 实例必须位于基于 Amazon VPC 服务的虚拟私有云 (VPC) 中。还必须将其配置为使用提供的 DNS 服务器 AWS。有关 Amazon DNS 服务器的信息，请参阅《Amazon VPC 用户指南》中的[Amazon VPC 中的 DHCP 选项集](#)。
- VPC 已启用 DNS 主机名 - 连接的 EC2 实例的 VPC 必须启用了 DNS 主机名。有关更多信息，请参阅《Amazon VPC 用户指南》中的[VPC 的 DNS 属性](#)。
- 对于不同的 EC2 实例和文件系统 AWS 区域 — 如果 EC2 实例和您要挂载的文件系统位于不同的位置 AWS 区域，则需要编辑efs-utils.conf文件中的region属性。有关更多信息，请参阅[从其他服务器装载 EFS 文件系统 AWS 区域](#)。

使用 EFS 挂载帮助程序在 EC2 Linux 实例上挂载

此过程需要满足以下条件：

- 已在 Amazon EC2 实例上安装 amazon-efs-utils 软件包。有关更多信息，请参阅[手动安装 Amazon EFS 客户端](#)。
- 已为文件系统创建挂载目标。有关更多信息，请参阅[管理挂载目标](#)。

使用挂载帮助程序在 EC2 Linux 实例上挂载 EFS 文件系统

1. 通过 Secure Shell (SSH) 打开 EC2 实例上的终端窗口，然后使用正确的用户名登录。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 使用以下命令创建要用作文件系统装载点的目录 efs：

```
sudo mkdir efs
```

3. 运行以下命令之一来挂载文件系统。

Note

如果 EC2 实例和您要挂载的文件系统位于不同 AWS 区域，请参阅[从其他服务器装载 EFS 文件系统 AWS 区域](#)以编辑 efs-utils.conf 文件中的 region 属性。

- 使用文件系统 ID 装载：

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

使用您要安装的文件系统的 ID *file-system-id* 来代替 *efs-mount-point*。

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

或者，如果要使用传输中的数据加密，您可以使用以下命令挂载文件系统。

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

- 使用文件系统 DNS 名称挂载：

```
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/
```

- 使用挂载目标 IP 地址挂载：

```
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/
```

可以在附加对话框中查看和复制用于挂载文件系统的确切命令。

- 在 Amazon EFS 控制台中，选择要挂载的文件系统以显示其详细信息页面。
- 要显示用于此文件系统的挂载命令，请选择右上角的附加。

附加屏幕显示用于通过以下方式挂载文件系统的确切命令：

- （通过 DNS 挂载）将文件系统的 DNS 名称与 EFS 挂载帮助程序或 NFS 客户端一起使用。
- （通过 IP 挂载）使用 NFS 客户端的选定可用区中的挂载目标 IP 地址。

使用 EFS 挂载帮助程序在 EC2 Mac 实例上挂载

此过程需要满足以下条件：

- 已在 Amazon EC2 Mac 实例上安装 `amazon-efs-utils` 软件包。有关更多信息，请参阅 [在运行 macOS Big Sur、macOS Monterey 或 macOS Ventura 的 EC2 Mac 实例上安装 Amazon EFS 客户端](#)。
- 已为文件系统创建挂载目标。可以在创建文件系统时创建挂载目标，并将其添加到现有文件系统中。有关更多信息，请参阅 [管理挂载目标](#)。
- 您正在运行 macOS Big Sur、Monterey 或 Ventura 的 EC2 Mac 实例上挂载文件系统。不支持其他 macOS 版本。

Note

仅支持运行 macOS Big Sur、Monterey 和 Ventura 的 EC2 Mac 实例。不支持将其他 macOS 版本与 Amazon EFS 结合使用。

在运行 macOS Big Sur、Monterey 或 Ventura 的 EC2 Mac 实例上使用 EFS 挂载帮助程序挂载 EFS 文件系统

1. 通过 Secure Shell (SSH) 打开 EC2 Mac 实例上的终端窗口，然后使用正确的用户名登录。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 使用以下命令创建要用作文件系统装载点的目录：

```
sudo mkdir efs
```

3. 运行以下命令以挂载文件系统。

Note

默认情况下，无论您是否在挂载命令中使用 `tls` 选项，EFS 挂载帮助程序在 EC2 Mac 实例上挂载时都使用传输中加密。

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

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

挂载时还可以使用 `tls` 选项。

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

要在 EC2 Mac 实例上挂载文件系统而不使用传输中加密，请使用 `notls` 选项，如以下命令所示。

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

可以按照如下所述，在管理控制台的附加对话框中查看和复制挂载文件系统的确切命令。

- a. 在 Amazon EFS 控制台中，选择要挂载的文件系统以显示其详细信息页面。
- b. 要显示用于此文件系统的挂载命令，请选择右上角的附加。

附加屏幕显示用于通过以下方式挂载文件系统的确切命令：

- （通过 DNS 挂载）将文件系统的 DNS 名称与 EFS 挂载帮助程序或 NFS 客户端一起使用。
- （通过 IP 挂载）使用 NFS 客户端的选定可用区中的挂载目标 IP 地址。

从其他服务器装载 EFS 文件系统 AWS 区域

要从与文件系统不同 AWS 区域的 EC2 实例挂载 EFS 文件系统，必须编辑 `efs-utils.conf` 文件中的 `region` 属性值。

在 **`efs-utils.conf`** 中编辑 **`region`** 属性

1. 通过 Secure Shell (SSH) 访问您的 EC2 实例的终端，然后使用相应的用户名登录。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 找到 `/etc/amazon/efs/efs-utils.conf` 文件并使用首选编辑器打开。
3. 找到以下行：

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

- a. 取消注释此行。
 - b. 如果文件系统不在 us-east-1 区域中，请将 us-east-1 替换为文件系统所在的区域 ID。
 - c. 保存更改。
4. 为跨区域挂载添加主机条目。有关此操作的更多信息，请参阅[步骤 3：为挂载目标添加主机条目](#)。
 5. 使用适用于 [Linux](#) 或 [Mac](#) 实例的 EFS 挂载帮助程序挂载文件系统。

挂载单区文件系统

EFS 单区文件系统仅支持与文件系统位于同一可用区的单个挂载目标。无法添加其他挂载目标。本节介绍挂载单区文件系统时需要考虑的事项。

使用与文件系统挂载目标位于同一可用区的 Amazon EC2 计算实例访问 EFS 文件系统，可以避免在可用区之间产生数据传输费用并获得更好的性能。

本节中的过程需要满足以下条件：

- 已在 EC2 实例上安装 amazon-efs-utils package。有关更多信息，请参阅[安装 Amazon EFS 客户端](#)。
- 已为文件系统创建挂载目标。有关更多信息，请参阅[管理挂载目标](#)。

在不同可用区的 EC2 上挂载单区文件系统

如果要在位于不同可用区的 Amazon EC2 实例上挂载单区文件系统，必须在挂载帮助程序挂载命令中指定文件系统的可用区名称或文件系统挂载目标的 DNS 名称。

使用以下命令创建要用作文件系统挂载点的名为 efs 的目录：

```
sudo mkdir efs
```

使用以下命令，通过 EFS 挂载帮助程序挂载文件系统。此命令指定文件系统的可用区名称。

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

以下是使用示例值的此命令：

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

以下命令挂载文件系统，指定文件系统挂载目标的 DNS 名称。

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

以下是使用示例挂载目标 DNS 名称的此命令。

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

使用 EFS 挂载帮助程序在不同可用区中自动装载单区文件系统

如果要使用 `/etc/fstab` 在位于不同可用区的 EC2 实例上挂载 EFS 单区文件系统，必须在 `/etc/fstab` 条目中指定文件系统的可用区名称或文件系统挂载目标的 DNS 名称。

```
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
```

使用 NFS 自动挂载单区文件系统

如果要使用 `/etc/fstab` 在位于不同可用区的 EC2 实例上挂载使用单区存储的 EFS 文件系统，必须在 `/etc/fstab` 条目中使用文件系统的 DNS 名称指定文件系统的可用区名称。

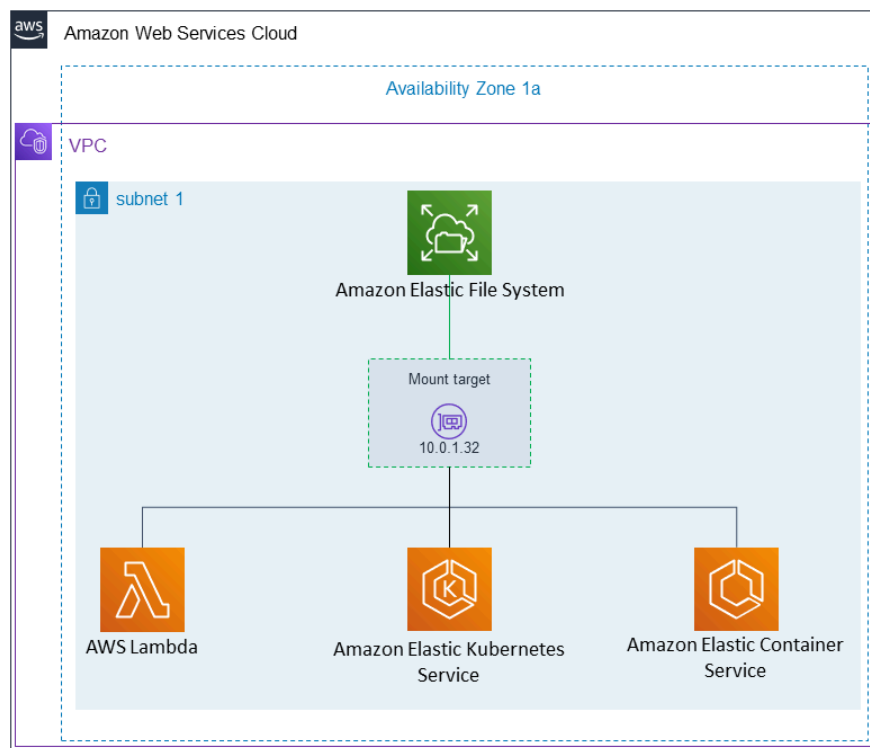
```
availability-zone-name.file-system-id.efs.aws-region.amazonaws.com:/ efs-mount-point  
nfs4  
nfsvers=4.1,rsiz=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0  
0
```

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

有关如何编辑 `/etc/fstab` 文件，以及此命令中使用的值的更多信息，请参阅[自动挂载 EFS 文件系统](#)。

在其他 AWS 计算实例上安装带有 One Zone 文件系统的文件系统

当您将单区域文件系统与亚马逊弹性容器服务、Amazon Elastic Kubernetes Service AWS Lambda或 Amazon Elastic Kubernetes Service 一起使用时，您需要将该服务配置为使用与 EFS 文件系统相同的可用区，如下所示，并在以下各节中进行介绍。



从 Amazon Elastic Container Service 连接

可以将 EFS 文件系统与 Amazon ECS 配合使用，以跨容器实例集共享文件系统数据，从而使您的任务无论位于哪个实例上，都可以访问相同的永久存储。要将 EFS 单区文件系统与 Amazon ECS 一起使用，应在启动任务时仅选择与文件系统位于同一可用区的子网。有关更多信息，请参阅《Amazon Elastic Container Service 开发人员指南》中的 [Amazon EFS 卷](#)。

从 Amazon Elastic Kubernetes Service 连接

在从 Amazon EKS 挂载单区文件系统时，可以使用支持 EFS 接入点的 Amazon EFS [容器存储接口](#) (CSI) 驱动程序，以在 Amazon EKS 或自行管理的 Kubernetes 集群中的多个容器组 (pod) 之间共享文件系统。Amazon EFS CSI 驱动程序安装在 Fargate 堆栈中。将 Amazon EFS CSI 驱动程序与 EFS 单区文件系统配合使用时，可以在启动容器组 (pod) 时使用 nodeSelector 选项来确保它与您的文件系统在同一个可用区内进行调度。

连接自 AWS Lambda

您可以将 Amazon EFS 与配合 AWS Lambda 使用，在函数调用之间共享数据，读取大型参考数据文件，并将函数输出写入永久存储和共享存储。Lambda 将函数实例安全地连接到位于同一可用区和子网中的 EFS 挂载目标。将 Lambda 与单区文件系统一起使用时，应将函数配置为只将调用启动到与您的文件系统位于同一可用区的子网中。

使用 IAM 授权挂载

要使用 AWS Identity and Access Management (IAM) 授权在 Linux 实例上挂载 EFS 文件系统，请使用 EFS 挂载帮助程序。有关 NFS 客户端的 IAM 授权的更多信息，请参阅[使用 IAM 控制文件系统访问](#)。

在以下几部分中，您需要创建一个目录作为文件系统挂载点。可使用以下命令创建挂载点目录 `efs`：

```
sudo mkdir efs
```

然后，可以将 *efs-mount-point* 的实例替换为 `efs`。

使用 EC2 实例配置文件通过 IAM 进行挂载

如果要通过 IAM 授权挂载到具有实例配置文件的 Amazon EC2 实例，请使用 `tls` 和 `iam` 挂载选项，如下所示。

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

要使用 IAM 授权自动挂载到具有实例配置文件的 EC2 实例，请将以下行添加到 EC2 实例上的 `/etc/fstab` 文件中。

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

使用命名配置文件通过 IAM 进行挂载

您可以使用证书文件或 AWS CLI 配置文件 `~/.aws/credentials` 中的 IAM 凭证通过 IAM 授权进行挂载 `~/.aws/config`。AWS CLI 如果未指定 `"awsprofile"`，则使用“默认”配置文件。

要使用凭证文件通过 IAM 授权挂载到 Linux 实例，请使用 `tls`、`awsprofile` 和 `iam` 挂载选项，如下所示。

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

要使用凭证文件通过 IAM 授权自动挂载到 Linux 实例，请将以下行添加到 EC2 实例上的 `/etc/fstab` 文件中。

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

使用 EFS 接入点进行挂载

只有使用 EFS 挂载帮助程序，才能使用 EFS 接入点挂载 EFS 文件系统。

Note

使用 EFS 接入点挂载文件系统时，必须为文件系统配置一个或多个挂载目标。

在使用访问点挂载文件系统时，除常规挂载选项外，挂载命令还包括 `access-point-id` 和 `tls` 挂载选项。下面显示了一个示例。

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

要使用访问点自动挂载文件系统，请将以下行添加到 EC2 实例上的 `/etc/fstab` 文件中。

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

有关 EFS 访问点的更多信息，请参阅[使用接入点工作](#)。

将 EFS 挂载到多个 EC2 实例

使用 AWS Systems Manager Run Command，可以远程安全地将 EFS 文件系统挂载到多个 Amazon EC2 实例，而无需登录这些实例。有关 AWS Systems Manager Run Command 的更多信息，请参阅《AWS Systems Manager 用户指南》中的[AWS Systems Manager Run Command](#)。使用此方法挂载 EFS 文件系统之前，需要满足以下先决条件：

1. 已使用包含 AmazonElasticFileSystemsUtils 权限策略的实例配置文件启动 EC2 实例。有关更多信息，请参阅[步骤 1：使用所需权限配置 IAM 实例配置文件](#)。
2. 在 EC2 实例上安装了 Amazon EFS 客户端 (amazon-efs-utils 软件包) 的 1.28.1 或更高版本。您可以使用 S AWS systems Manager 在您的实例上自动安装软件包。有关更多信息，请参阅[步骤 2：配置 State Manager 使用的关联](#)。

使用控制台将多个 EFS 文件系统挂载到多个 EC2 实例

1. 打开 AWS Systems Manager 控制台，网址为<https://console.aws.amazon.com/systems-manager/>。
2. 在导航窗格中，选择 Run Command。
3. 选择 Run a command (运行一个命令)。
4. 在命令搜索字段中输入 **AWS-RunShellScript**。
5. 选择 AWS-RunShellScript。
6. 在命令参数中，输入用于要挂载的每个 EFS 文件系统的装载命令。例如：

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

有关使用 Amazon EFS 客户端的 EFS 挂载命令的更多信息，请参阅[使用 EFS 挂载帮助程序在 EC2 Linux 实例上挂载](#)或[使用 EFS 挂载帮助程序在 EC2 Mac 实例上挂载](#)。

7. 选择要在其上运行命令的目标 AWS Systems Manager 托管 EC2 实例。
8. 根据需要进行任何其他设置。然后选择运行以运行此命令，并挂载命令中指定的 EFS 文件系统。

运行命令后，可以在命令历史记录中查看其状态。

从其他 AWS 账户 或 VPC 挂载 EFS 文件系统

可以使用 EFS 挂载帮助程序，通过 NFS 客户端和 EFS 接入点的 IAM 授权来挂载您的 EFS 文件系统。默认情况下，EFS 挂载帮助程序使用域名服务 (DNS) 来解析您的 EFS 挂载目标的 IP 地址。如果要从其他账户或 Virtual Private Cloud (VPC) 挂载文件系统，必须手动解析 EFS 挂载目标。

下面，您可以找到确定要用于您的 NFS 客户端的正确 EFS 挂载目标 IP 地址的说明。您还可以找到有关配置客户端以使用该 IP 地址挂载 EFS 文件系统的说明。

主题

- [从另一个挂载 EFS 文件系统 AWS 账户](#)
- [从另一个 VPC 挂载 EFS 文件系统](#)

从另一个挂载 EFS 文件系统 AWS 账户

使用共享 VPCs，您可以 AWS 账户 从另一个 Amazon EC2 实例中挂载一个人拥有的 EFS 文件系统 AWS 账户。有关设置共享 VPC 的更多信息，请参阅《Amazon VPC 对等连接指南》中的[与其它账户共享 VPC](#)。

设置 VPC 共享之后，EC2 实例可以使用域名系统 (DNS) 名称解析或者 EFS 挂载帮助程序来挂载 EFS 文件系统。我们建议使用 EFS 挂载帮助程序挂载 EFS 文件系统。

从另一个 VPC 挂载 EFS 文件系统

当您使用 VPC 对等连接或传输网关连接 VPC 时，一个 VPC 中的 Amazon EC2 实例可以访问另一个 VPC 中的 EFS 文件系统，即使 VPCs 它们属于不同的账户。

您不能对其他 VPC 中的 EFS 挂载点使用 DNS 名称解析。要挂载您的 EFS 文件系统，请使用对应可用区中挂载点的 IP 地址。

或者，您可以使用 Amazon Route 53 作为您的 DNS 服务。在 Route 53 中，您可以通过创建私有托管区和资源记录集来解析其他 VPC 中的 EFS 挂载目标 IP 地址。有关如何执行此操作的更多信息，请参阅《Amazon Route 53 开发人员指南》中的[私有托管区的使用](#)。

先决条件

在使用下面的过程之前，请执行以下步骤：

- 在要挂载 EFS 文件系统的计算实例上安装 Amazon EFS 客户端，这是实用程序 amazon-efs-utils 集的一部分。可以使用 amazon-efs-utils 中包含的 EFS 挂载帮助程序来挂载文件系统。有关安装 amazon-efs-utils 的说明，请参阅[安装 Amazon EFS 客户端](#)。
- 允许在 IAM 策略中对附加到实例的 IAM 角色执行 ec2:DescribeAvailabilityZones 操作。我们建议您将 AWS 托管策略附加 AmazonElasticFileSystemsUtils 到 IAM 实体，以便为该实体提供必要的权限。
- 从另一个文件系统挂载时 AWS 账户，请更新文件系统资源策略以允许对其他主体 ARN elasticfilesystem:DescribeMountTarget 执行操作。AWS 账户例如：

```
{
  "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"
  }
]
```

有关 EFS 文件系统资源策略的更多信息，请参阅[Amazon EFS 基于资源的策略](#)。

- 安装 `botocore`。在另一个 VPC 中挂载文件系统时，如果无法解析文件系统 DNS 名称，则 EFS 客户端会使用 `botocore` 检索挂载目标 IP 地址。有关更多信息，请参阅 `amazon-efs-utils` 自述文件中的[安装 `botocore`](#)。
- 设置 VPC 对等连接或 VPC 传输网关。

可以使用 VPC 对等连接或 VPC 传输网关连接，连接客户端的 VPC 和您的 EFS 文件系统的 VPC。当您使用 VPC 对等连接或传输网关连接 VPC 时，一个 VPC 中的 Amazon EC2 实例可以访问另一个 VPC 中的 EFS 文件系统，即使 VPCs 它们属于不同的账户。

传输网关是一个网络中转枢纽，可用于将您的网络 VPCs 和本地网络互连。有关使用 VPC 中转网关的信息，请参阅《Amazon VPC 中转网关指南》中的[中转网关入门](#)。

VPC 对等连接是两 VPCs 者之间的网络连接。这种类型的连接使您能够使用私有 Internet 协议版本 4 (IPv4) 或 Internet 协议版本 6 (IPv6) 地址在它们之间路由流量。您可以使用 VPC 对等互连 VPCs 在同一个 AWS 区域 或两者之间 AWS 区域进行连接。有关 VPC 对等的更多信息，请参阅《Amazon VPC 对等指南》中的[什么是 VPC 对等？](#)。

为了确保您的文件系统的高可用性，我们建议您始终使用与 NFS 客户端位于相同可用区的 EFS 挂载目标 IP 地址。如果要挂载其它账户中的 EFS 文件系统，请确保 NFS 客户端和 EFS 挂载目标位于相同的可用区 ID 中。此要求适用，因为可用区名称在账户之间可能会有所不同。

使用 IAM 或访问点在其他 VPC 中挂载 EFS 文件系统

1. 连接到您的 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 使用以下命令创建用于挂载文件系统的目录。

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

3. 要使用 IAM 授权来挂载文件系统，请使用以下命令：

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

有关将 IAM 授权与 EFS 结合使用的更多信息，请参阅 [使用 IAM 控制文件系统访问](#)。

要使用 EFS 访问点挂载文件系统，请使用以下命令：

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

有关 EFS 访问点的更多信息，请参阅[使用接入点工作](#)。

从其他服务器装载 EFS 文件系统 AWS 区域

如果您要从与文件系统不同的 AWS 区域 其他 VPC 挂载 EFS 文件系统，则需要编辑该 `efs-utils.conf` 文件。在 `/dist/efs-utils.conf` 中找到以下行：

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

取消该行的注释，如果文件系统不在 `us-east-1`，则替换文件系统所在区域的 ID 值。

使用网络文件系统挂载 EFS 文件系统

在下文中了解如何安装网络文件系统 (NFS) 客户端，以及如何在 Amazon EC2 实例上挂载 Amazon EFS 文件系统。还可以找到用于在 `mount` 命令中指定文件系统的域名系统 (DNS) 名称的 `mount` 命令和可用选项的说明。此外，您还可以了解如何使用 `fstab` 文件在任何系统重新启动后自动重新挂载您的文件系统。

Note

在本节中，您可以学习如何在没有 `amazon-efs-utils` 软件包的情况下挂载 Amazon EFS 文件系统。要使用文件系统加密传输中的数据，您必须使用传输层安全性 (TLS) 挂载文件系统。为此，我们建议使用该 `amazon-efs-utils` 软件包。有关更多信息，请参阅 [安装 Amazon EFS 客户端](#)。

主题

- [先决条件](#)
- [NFS 支持](#)
- [安装 NFS 客户端](#)
- [推荐的 NFS 挂载设置](#)
- [使用 DNS 名称在 Amazon EC2 上挂载](#)
- [使用 IP 地址挂载](#)

先决条件

在可以挂载文件系统之前，请确保满足以下要求：

- 创建、配置和启动相关 AWS 资源。有关说明，请参阅[开始使用 Amazon EFS](#)。
- 为您的 Amazon EC2 实例创建 VPC 安全组，并挂载具有所需入站和出站访问权限的目标。有关更多信息，请参阅[使用 VPC 安全组](#)。

NFS 支持

在 Amazon EC2 实例上挂载文件系统时，Amazon EFS 支持网络文件系统版本 4.0 和 4.1 (NFSv4) 协议。虽然支持 NFSv4.0，但我们建议您使用 NFSv4.1。在 Amazon EC2 实例上挂载 Amazon EFS 文件系统时，还需要使用支持所选的 NFSv4 协议的 NFS 客户端。运行 macOS Big Sur 的 Amazon EC2 Mac 实例仅支持 NFS v4.0。

Amazon EFS 不支持 `nconnect` 挂载选项。

Note

对于 Linux 内核版本 5.4.*，Linux NFS 客户端使用 128 KB 的默认 `read_ahead_kb` 值。我们建议将此值增加到 15 MB。有关更多信息，请参阅[优化 NFS read_ahead_kb 的大小](#)。

为获得最佳性能以及避免出现各种已知的 NFS 客户端错误，我们建议您使用最新的 Linux 内核。如果使用的是企业 Linux 发行版，我们建议您使用以下版本：

- Amazon Linux 2
- Amazon Linux 2017.09 或更高版本

- 红帽企业 Linux (以及诸如 CentOS 之类的衍生产品) 版本 8 及更高版本
- Ubuntu 16.04 LTS 和更新版本
- SLES 12 Sp2 或更高版本

如果使用其他发行版或自定义内核，我们建议您使用内核 4.3 或更高版本。要解决从 EC2 实例使用 Amazon EFS 时与某些 AMI 或内核版本相关的问题，请参阅[排查 AMI 和内核问题](#)。

Note

不支持使用运行 Microsoft Windows 的 Amazon EC2 实例挂载 EFS 文件系统。

安装 NFS 客户端

要在 Amazon EC2 实例上挂载 EFS 文件系统，首先需要安装 NFS 客户端。要连接到 EC2 实例并安装 NFS 客户端，您需要 EC2 实例的公有 DNS 名称和用户名称进行登录。实例的用户名通常为 `ec2-user`。

连接 EC2 实例和安装 NFS 客户端

1. 连接到 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。

密钥文件不能对 SSH 公开可见。您可以使用 `chmod 400 filename.pem` 命令设置这些权限。有关更多信息，请参阅[为您的 Amazon EC2 实例创建密钥对](#)。

2. (可选) 获取更新并重启。

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

3. 重启后，重新连接到您的 EC2 实例。
4. 安装 NFS 客户端。

如果您使用的是 Amazon Linux AMI 或 Red Hat Linux AMI，请使用以下命令安装 NFS 客户端。

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

如果您使用的是 Ubuntu Amazon EC2 AMI，请使用以下命令安装 NFS 客户端。

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

5. 使用以下命令启动 NFS 服务：

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

6. 验证 NFS 服务已启动，如下所示。

```
$ sudo service nfs-server status
Redirecting to /bin/systemctl status nfs.service
# nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor
   preset: disabled)
   Active: active (exited) since Wed 2019-10-30 16:13:44 UTC; 5s ago
   Process: 29446 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
   Process: 29441 ExecStartPre=/bin/sh -c /bin/kill -HUP `cat /run/gssproxy.pid` (code=exited, status=0/SUCCESS)
   Process: 29439 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
   Main PID: 29446 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nfs-server.service
```

如果您使用自定义内核（即构建自定义 AMI），则至少需要包含 NFSv4 .1 客户端内核模块和正确的 NFS4 用户空间挂载帮助程序。

Note

如果在启动 Amazon EC2 实例时选择 Amazon Linux AMI 2016.03.0 或 Amazon Linux AMI 2016.09.0，您不需要安装 `nfs-utils`，因为它已默认包含在 AMI 中。

下一步：挂载您的文件系统

使用以下过程之一挂载您的文件系统。

- [使用 DNS 名称在 Amazon EC2 上挂载](#)
- [使用 IP 地址挂载](#)
- [自动挂载 EFS 文件系统](#)

推荐的 NFS 挂载设置

我们建议在 Linux 上使用以下挂载选项值：

- `noresvport` – 告知 NFS 客户端在重新建立网络连接时，使用新的非特权传输控制协议 (TCP) 源端口。旧版 Linux 内核 (版本 v5.4 及更低版本) 中包含的 NFS 客户端软件包含一种行为，该行为会导致 NFS 客户端在断开连接时尝试在同一 TCP 源端口上重新连接。此行为不符合 TCP RFC 要求，并且可能会阻止这些客户端快速重新建立与 EFS 文件系统的连接。

使用 `noresvport` 选项有助于确保 NFS 客户端以透明的方式重新连接到您的 EFS 文件系统，从而在网络恢复事件后重新连接时保持不间断的可用性。

Important

我们强烈建议您使用 `noresvport` 挂载选项来帮助确保 EFS 文件系统在重新连接或网络恢复事件后保持不间断的可用性。

考虑使用 [EFS 挂载帮助程序](#) 挂载您的文件系统。EFS 挂载帮助程序使用针对 Amazon EFS 文件系统优化了的 NFS 挂载选项。

- `rsiz=1048576` – 设置 NFS 客户端对每个网络 READ 请求可以接收的最大数据字节数。在从 EFS 文件系统上的文件读取数据时应用此值。我们建议您尽可能使用最大的大小 (最多 1048576)，以避免性能下降。
- `wsiz=1048576` – 设置 NFS 客户端对每个网络 WRITE 请求可以发送的最大数据字节数。在将数据写入到 EFS 文件系统上的文件时应用此值。我们建议您尽可能使用最大的大小 (最多 1048576)，以避免性能下降。
- `hard` – 设置 NFS 客户端在 NFS 请求超时之后的恢复行为，以便 NFS 请求在服务器回复之前无限次重试。建议您使用硬挂载选项 (`hard`) 以确保数据完整性。如果您使用 `soft` 挂载，请将 `timeo` 参数至少设置为 150 分秒 (15 秒)。这样做可尽量减少源自软挂载的数据损坏风险。
- `timeo=600` – 将超时值设置为 600 分秒 (60 秒)，这是 NFS 客户端在重试 NFS 请求之前等待响应的的时间。如果您必须更改超时参数 (`timeo`)，我们建议您使用至少为 150 的值，这相当于 15 秒。这样做有助于避免性能下降。
- `retrans=2` – 将 NFS 客户端重试请求的次数设置为 2，超过此次数之后将尝试进一步的恢复操作。
- `_netdev` – `/etc/fstab` 中存在此选项时，将阻止客户端尝试挂载 EFS 文件系统，直到启用了网络。
- `nofail` – 如果需要启动您的 EC2 实例而不考虑挂载的 EFS 文件系统状态，请将 `nofail` 选项添加到 `/etc/fstab` 文件的文件系统条目中。

如果您不使用前面的默认值，请注意以下事项：

- 一般而言，避免设置任何其他不同于默认值的挂载选项，这会导致性能降低和其他问题。例如，更改读或写缓冲区大小或禁用属性缓存会导致性能下降。
- Amazon EFS 会忽略源端口。如果您更改 Amazon EFS 源端口，则不会有任何影响。
- Amazon EFS 不支持 `nconnect` 挂载选项。
- Amazon EFS 不支持任何 Kerberos 安全变体。例如，下面的挂载命令将失败。

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

- 我们建议您使用其 DNS 名称挂载文件系统。此名称解析为与您 Amazon EC2 实例位于相同可用区中的 Amazon EFS 挂载目标的 IP 地址。如果您在与 Amazon EC2 实例不同的可用区中使用挂载目标，则会对跨可用区发送的数据收取标准 EC2 费用。可能还会面临更高的文件系统操作延迟。
- 有关更多挂载选项和默认设置的详细说明，请参阅 Linux 文档。

使用 DNS 名称在 Amazon EC2 上挂载

Note

在挂载文件系统之前，您需要向挂载目标安全组添加一条规则，支持通过端口 2049 从 EC2 安全组进行入站 NFS 访问。有关更多信息，请参阅 [使用 VPC 安全组](#)。

- 文件系统 DNS 名称 – 使用文件系统的 DNS 名称是最简单的挂载方法。文件系统 DNS 名称自动解析为连接的 Amazon EC2 实例的可用区中的挂载目标 IP 地址。您可以从控制台中获取该 DNS 名称，或者，如果您具有文件系统 ID，可以使用以下约定构造该名称。

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

Note

文件系统 DNS 名称的 DNS 解析要求 Amazon EFS 文件系统在与客户端实例相同的可用区中具有挂载目标。

- 通过使用文件系统 DNS 名称，您可以使用以下命令在 Amazon EC2 Linux 实例上挂载文件系统。

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

- 通过文件系统 DNS 名称，您可以使用以下命令在运行支持的 macOS 版本 (Big Sur、Monterey、Ventura) 的 Amazon EC2 Mac 实例上挂载文件系统。

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

Important

在运行支持的 macOS 版本的 EC2 Mac 实例上挂载时，必须使用 mountport=2049 才能成功连接到 EFS 文件系统。

- 挂载目标 DNS 名称 – 2016 年 12 月，我们引入了文件系统 DNS 名称。我们继续为每个可用区挂载目标提供 DNS 名称以保持向后兼容。挂载目标 DNS 名称的通用形式如下所示。

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

Note

支持跨可用区的挂载目标 DNS 名称解析。

在某些情况下，您可能会删除挂载目标，然后在同一可用区中创建新的挂载目标。在这种情况下，该可用区中的新挂载目标的 DNS 名称与旧挂载目标的 DNS 名称相同。

可以在附加对话框中查看和复制用于挂载文件系统的确切命令。

查看文件系统的挂载命令

1. 在 Amazon EFS 控制台中，选择要挂载的文件系统以显示其详细信息页面。
2. 要显示用于此文件系统的挂载命令，请选择右上角的附加。

附加屏幕显示用于挂载文件系统的确切命令。

3. 默认的通过 DNS 挂载视图显示在使用 EFS 挂载帮助程序或 NFS 客户端挂载时使用文件系统的 DNS 名称挂载文件系统的命令。

有关支持 Amazon EFS 的 AWS 区域的列表，请参阅中的 [Amazon Elastic File System AWS 一般参考](#)。

要能够在 mount 命令中使用 DNS 名称，必须满足以下条件：

- 连接的 EC2 实例必须在 VPC 内，并且必须配置为使用 Amazon 提供的 DNS 服务器。有关 Amazon DNS 服务器的信息，请参阅《Amazon VPC 用户指南》中的 [Amazon VPC 中的 DHCP 选项集](#)。
- 连接的 EC2 实例的 VPC 必须启用 DNS Resolution (DNS 解析) 和 DNS Hostnames (DNS 主机名)。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 的 DNS 属性](#)。
- 连接的 EC2 实例必须位于与 EFS 文件系统相同的 VPC 内。有关从其他位置或不同的 VPC 访问和挂载文件系统的更多信息，请参阅[先决条件](#)和[教程：挂载来自其它 VPC 的文件系统](#)。

Note

在创建挂载目标后，我们建议您等待 90 秒，然后再挂载您的文件系统。这种等待可以让 DNS 记录在文件系统 AWS 区域 所在的位置完全传播。

使用 IP 地址挂载

作为使用 DNS 名称挂载 Amazon EFS 文件系统的替代方案，Amazon EC2 实例可使用挂载目标的 IP 地址来挂载文件系统。通过 IP 地址挂载适用于禁用 DNS 的环境，例如 VPCs 禁用 DNS 主机名的环境。

对于配置为默认使用 DNS 名称挂载文件系统的应用程序，您还可以将使用挂载目标 IP 地址挂载文件系统配置为回退选项。当连接到挂载目标 IP 地址时，EC2 实例应使用连接实例所在的同一可用区中的挂载目标 IP 地址进行挂载。

可以在附加对话框中查看和复制用于挂载文件系统的确切命令。

Note

在挂载文件系统之前，您需要向挂载目标安全组添加一条规则，支持通过端口 2049 从 EC2 安全组进行入站 NFS 访问。有关更多信息，请参阅 [使用 VPC 安全组](#)。

查看和复制使用挂载目标 IP 地址挂载 EFS 文件系统的确切命令

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 在 Amazon EFS 控制台中，选择要挂载的文件系统以显示其详细信息页面。
3. 要显示用于此文件系统的挂载命令，请选择右上角的附加。
4. 附加屏幕显示用于挂载文件系统的确切命令。

选择通过 IP 挂载，以显示使用 NFS 客户端选定可用区中的挂载目标 IP 地址挂载文件系统的命令。

- 在 mount 命令中使用挂载目标的 IP 地址，您可以使用以下命令在 Amazon EC2 Linux 实例上挂载文件系统。

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

- 在 mount 命令中使用挂载目标的 IP 地址，您可以通过以下命令在运行 macOS Big Sur 的 Amazon EC2 Mac 实例上挂载文件系统。

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

Important

在运行 macOS Big Sur 的 EC2 Mac 实例上挂载时，必须使用 mountport=2049 才能成功连接到 EFS 文件系统。

使用 IP 地址进行装载 AWS CloudFormation

您也可以使用 CloudFormation 模板中的 IP 地址挂载文件系统。有关更多信息，请参阅 [awsdocs/elastic-beanstalk-samples](#) 存储库中的 [storage-efs-mountfilesystem-ip-addr.config](#)，了解社区提供的配置文件。GitHub

自动挂载 EFS 文件系统

您可以使用 EFS 挂载帮助程序或 NFS 将 Amazon EC2 实例配置为在实例启动时自动挂载 EFS 文件系统。

- 使用 EFS 挂载帮助程序：
 - 在使用 EC2 启动实例向导创建新 EC2 Linux 实例时附加 EFS 文件系统。
 - 使用 EFS 文件系统条目更新 EC2 的 `/etc/fstab` 文件。
- 对于 EC2 Linux 和 Mac 实例，使用[没有 EFS 挂载帮助程序的 NFS](#)更新 EC2 `/etc/fstab` 文件。

Note

EFS 挂载帮助程序不支持在运行 macOS Big Sur 或 Monterey 的 EC2 Mac 实例上自动挂载。但是，您可以使用[NFS 在 EC2 Mac 实例上配置 `/etc/fstab` 文件](#)以自动挂载 EFS 文件系统。

主题

- [在新 EC2 Linux 实例上启用自动挂载](#)
- [在现有 EC2 Linux 实例上启用自动挂载](#)
- [使用 NFS 在 EC2 Linux 或 Mac 实例上启用自动挂载](#)

在新 EC2 Linux 实例上启用自动挂载

使用 Amazon EC2 启动实例向导创建新的 EC2 Linux 实例时，可以将该实例配置为自动挂载您的 Amazon EFS 文件系统。EC2 实例会在第一次启动时自动挂载文件系统，并且在重新启动时也会自动挂载文件系统。

此方法使用 EFS 挂载帮助程序挂载文件系统，以更新 EC2 实例上的 `/etc/fstab` 文件。挂载帮助程序是[amazon-efs-utils](#) 工具集的一部分。

Note

EFS 文件系统不支持在实例启动时在运行 macOS Big Sur 或 Monterey 的 EC2 Mac 实例上挂载。

 Note

不能将 Amazon EFS 与基于 Microsoft Windows 的 EC2 实例结合使用。

在启动并连接到 EC2 实例之前，需要创建一个密钥对。有关创建密钥对的更多信息，请参阅《Amazon EC2 用户指南》中的 [Amazon EC2 密钥对](#) 和 [Amazon EC2 实例](#)。

将 EC2 实例配置为在启动时自动挂载 EFS 文件系统

1. 打开位于 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 控制台。
2. 选择启动实例。
3. 在步骤 1：选择一个 Amazon 系统映像 (AMI) 中，在列表顶部找到一个 Amazon Linux AMI，然后选择选择。
4. 在步骤 2：选择一个实例类型中，选择下一步：配置实例详细信息。
5. 在 Step 3: Configure Instance Details (步骤 3：配置实例详细信息) 中，请提供以下信息：
 - 对于 Network (网络)，为您装载的 EFS 文件系统所在的同一 VPC 选择条目。
 - 对于 Subnet (子网)，在任何可用区中选择一个默认子网。
 - 对于 File system (文件系统)，选择要装载的 EFS 文件系统。文件系统 ID 旁边显示的路径是 EC2 实例将使用的装载点，您可以更改此装载点。
 - 在 Advanced Details (高级详细信息) 下，将自动生成 User data (用户数据)，并包括挂载您在 File systems (文件系统) 下指定的 EFS 文件系统所需的命令。
6. 选择下一步：添加存储。
7. 选择 Next: Add Tags。
8. 命名您的实例，然后选择下一步：配置安全组。
9. 在 Step 6: Configure Security Group (步骤 6：配置安全组) 中，将 Assign a security group (分配安全组) 设置为 Select an existing security group (选择现有安全组)。选择默认安全组以确保它能够访问您的 EFS 文件系统。

您不能使用该安全组通过安全外壳 (SSH) 访问您的 EC2 实例。对于通过 SSH 进行访问，您稍后可以编辑默认安全性并添加一个允许 SSH 的规则/新安全组。您可以使用以下设置：

- 类型：SSH
- 协议：TCP
- 端口范围：22

- 源：任何位置 0.0.0.0/0
10. 选择审核并启动。
 11. 选择启动。
 12. 选中您创建的密钥对的复选框，然后选择启动实例。

您的 EC2 实例现已配置为在启动或重新启动时装载 EFS 文件系统。

在现有 EC2 Linux 实例上启用自动挂载

/etc/fstab 文件包含有关文件系统的信息。命令 `mount -a` 在实例启动期间运行，用于挂载 /etc/fstab 中列出的所有文件系统。在此过程中，您将在 Amazon EC2 Linux 实例上手动更新 /etc/fstab，以便该实例在实例重启时使用 EFS 挂载帮助程序自动重新挂载 EFS 文件系统。

Note

EFS 文件系统不支持在运行 macOS Big Sur 或 Monterey 的 EC2 Mac 实例上使用 /etc/fstab 和 EFS 挂载帮助程序自动挂载。相反，可以[将 NFS 与 /etc/fstab 结合使用](#)，在运行 macOS Big Sur 和 Monterey 的 EC2 Mac 实例上自动挂载文件系统。

此方法使用 EFS 挂载帮助程序来挂载文件系统。挂载帮助程序是 amazon-efs-utils 工具集的一部分。

这些 amazon-efs-utils 工具可在亚马逊 Linux 和亚马逊 Linux 2 亚马逊机器映像 (AMIs) 上安装。有关 amazon-efs-utils 的更多信息，请参阅[安装 Amazon EFS 客户端](#)。如果您正在使用其他 Linux 发行版，例如 Red Hat Enterprise Linux (RHEL)，请手动构建并安装 amazon-efs-utils。有关更多信息，请参阅[在其他 Linux 发行版上安装 Amazon EFS 客户端](#)。

先决条件

在成功实施此过程之前，需要满足以下要求：

- 已经创建了想要自动重新挂载的 EFS 文件系统。有关更多信息，请参阅[使用控制台快速创建](#)。
- 已经创建了要配置为自动重新挂载 EFS 文件系统的 EC2 Linux 实例。
- EFS 挂载帮助程序已安装在 EC2 Linux 实例上。有关更多信息，请参阅[安装 Amazon EFS 客户端](#)。

更新 /etc/fstab 文件

执行以下步骤，更新 EC2 Linux 实例上的 /etc/fstab，以便该实例在实例重启时使用 EFS 挂载帮助程序自动重新挂载 EFS 文件系统。

更新 EC2 实例上的 /etc/fstab 文件

1. 连接到您的 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 在编辑器中打开 /etc/fstab 文件。
3. 对于使用 IAM 授权或 EFS 接入点的自动挂载：
 - 要使用 IAM 授权自动挂载到具有实例配置文件的 EC2 实例，请将以下行添加到 /etc/fstab 文件中。

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

- 要使用凭证文件通过 IAM 授权自动挂载到 Linux 实例，请将以下行添加到 /etc/fstab 文件中。

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

- 要使用 EFS 访问点自动挂载文件系统，请将以下行添加到 /etc/fstab 文件中。

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

Warning

请在自动挂载文件系统时使用 _netdev 选项，它用于指定网络文件系统。如果缺少 _netdev，您的 EC2 实例可能会停止响应。出现该结果是因为，需要在计算实例启动其网络后初始化网络文件系统。有关更多信息，请参阅[自动挂载失败，并且实例没有响应](#)。

有关更多信息，请参阅[使用 IAM 授权挂载](#)和[使用 EFS 接入点进行挂载](#)。

4. 保存对文件所做的更改。

Note

在某些情况下，无论挂载的 EFS 文件系统的状态如何，都可能需要启动您的 EC2 实例。在这些情况下，将 `nofail` 选项添加到 `/etc/fstab` 文件中的文件系统条目中。

您添加到 `/etc/fstab` 文件的代码行将执行以下操作。

字段	说明
<code>file-system-id</code> :/	您的 EFS 文件系统的 ID。您可以从控制台获取此 ID，也可以通过编程方式从 CLI 或 AWS SDK 中获取此 ID。
<code>efs-mount-point</code>	EFS 文件系统在 EC2 实例上的挂载点。
<code>efs</code>	文件系统的类型。在使用挂载帮助程序时，该类型始终为 <code>efs</code> 。
<code>mount options</code>	文件系统的挂载选项。这是一个逗号分隔列表，包含以下选项： <code>_netdev</code> – 该选项向操作系统指示文件系统位于需要网络访问的设备上。该选项禁止实例挂载文件系统，直到在客户端上启用了网络。 <code>noresvport</code> – 告知 NFS 客户端在重新建立网络连接时，使用新的传输控制协议 (TCP) 源端口。这样做有助于确保 EFS 文件系统在网络恢复事件后具有不间断的可用性。 <code>tls</code> – 启用传输中数据加密。 <code>iam</code> - 使用此选项可以使用 IAM 授权挂载到具有实例配置文件的 EC2 实例。使用 <code>iam</code> 挂载选项还需要使用 <code>tls</code> 选项。有关更多信息，请参阅 使用 IAM 控制文件系统访问。 <code>awsprofile= namedprofile</code> – 将此选项与 <code>iam</code> 和 <code>tls</code> 选项结合使用，以使用凭证文件通过 IAM 授权挂载到 Linux 实例。有关 EFS 访问点的更多信息，请参阅 使用 IAM 控制文件系统访问。 <code>accesspoint= access-point-id</code> – 将此选项与 <code>tls</code> 选项结合使用以通过 EFS 接入点进行挂载。有关 EFS 访问点的更多信息，请参阅 使用接入点工作。
<code>0</code>	非零值表示应由 <code>dump</code> 备份文件系统。对于 EFS，该值应为 <code>0</code> 。

字段	说明
0	fsck 在启动时检查文件系统的顺序。对于 EFS 文件系统，该值应为 0，表示 fsck 不应在启动时运行。

使用 NFS 在 EC2 Linux 或 Mac 实例上启用自动挂载

对于 EC2 Linux 和 Mac 实例，使用没有 EFS 挂载帮助程序的 NFS 更新 Amazon EC2 `/etc/fstab` 文件。

更新 EC2 实例上的 `/etc/fstab` 文件

1. 连接到您的 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 在编辑器中打开 `/etc/fstab` 文件。
3. 要使用 NFS 而不是 EFS 挂载帮助程序自动挂载文件系统，请在 `/etc/fstab` 文件中添加以下行。
 - `file_system_id` 替换为要挂载的文件系统的 ID。
 - `aws-region` 替换 AWS 区域 为文件系统所在的，例如 `us-east-1`。
 - 将 `mount_point` 替换为文件系统的挂载点。

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

您添加到 `/etc/fstab` 文件的代码行将执行以下操作。

字段	说明
<code>file-system-id</code> <code>:</code>	您的 EFS 文件系统的 ID。您可以从控制台获取此 ID，也可以通过编程方式从 CLI 或 AWS SDK 中获取此 ID。
<code>efs-mount-point</code>	EFS 文件系统在 EC2 实例上的挂载点。

字段	说明
nfs4	指定文件系统类型。
mount options	逗号分隔的文件系统的挂载选项列表： • <code>nfsvers=4.1</code> – 指定使用 NFS v4.1。 • <code>rsiz=1048576</code> – 为了提高性能，设置从 EFS 文件系统上的文件读取数据时，NFS 客户端可以为每个网络读取请求接收的最大数据字节数。1048576 是可能的最大数量。 • <code>wsiz=1048576</code> – 为了提高性能，设置向 EFS 文件系统上的文件写入数据时，NFS 客户端可以为每个网络写入请求发送的最大数据字节数。1048576 是可能的最大数量。 • <code>hard</code> – 设置 NFS 客户端在 NFS 请求超时之后的恢复行为，以便 NFS 请求在服务器回复之前无限次重试。建议您使用硬挂载选项 (<code>hard</code>) 以确保数据完整性。如果您使用 <code>soft</code> 挂载，请将 <code>timeo</code> 参数至少设置为 150 分秒（15 秒）。这样做可尽量减少源自软挂载的数据损坏风险。 • <code>timeo=600</code> – 将超时值设置为 600 分秒（60 秒），这是 NFS 客户端在重试 NFS 请求之前等待响应的的时间。如果您必须更改超时参数 (<code>timeo</code>)，我们建议您使用至少为 150 的值，这相当于 15 秒。这样做有助于避免性能下降。 • <code>retrans=2</code> – 将 NFS 客户端重试请求的次数设置为 2，超过此次数之后将尝试进一步的恢复操作。 • <code>noresvport</code> – 告知 NFS 客户端在重新建立网络连接时，使用新的传输控制协议（TCP）源端口。这样做有助于确保 EFS 文件系统在网络恢复事件后具有不间断的可用性。 • <code>_netdev</code> – 禁止客户端尝试挂载 EFS 文件系统，直到启用了网络。
0	指定 dump 值；0 告诉 dump 实用程序不备份文件系统。
0	告诉 fsck 实用程序不在启动时运行。

卸载文件系统

在删除文件系统之前，建议您从该文件系统连接到的每个 Amazon EC2 实例卸载文件系统。可以通过在 Amazon EC2 实例上运行 `umount` 命令来从该实例上卸载文件系统。您无法通过 AWS CLI、或通过任何 AWS 软件开发工具包卸载 EFS 文件系统。AWS 管理控制台要卸载连接到运行 Linux 的 EC2 实例的 EFS 文件系统，请使用 `umount` 命令，如下所示：

```
umount /mnt/efs
```

建议您不要指定任何其他 `umount` 选项。避免设置不同于默认值的任何其他 `umount` 选项。

可以通过运行 `df` 命令，验证 EFS 文件系统是否已卸载。此命令显示当前挂载在基于 Linux 的 Amazon EC2 实例上的文件系统的磁盘使用统计信息。如果在 `df` 命令输出中没有列出要卸载的 EFS 文件系统，这意味着已卸载该文件系统。

Example- 标识 EFS 文件系统的挂载状态并卸载该文件系统

```
$ 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% /
```

教程：创建 EFS 文件系统并将其挂载到 EC2 实例上 AWS CLI

创建加密的 EFS 文件系统，将其挂载到 VPC 中的 EC2 实例上，然后使用 AWS CLI 测试设置。

Note

在[开始使用](#)教程中，您使用控制台创建 Amazon EC2 和 EFS 资源。在本教程中，您将使用 AWS CLI 来做同样的事情，主要是为了熟悉 Amazon EFS API。

在本教程中，您将在自己的账户中创建以下 AWS 资源：

- Amazon EC2 资源：
 - 两个安全组（一个用于 EC2 实例，一个用于 EFS 文件系统）。

您可以将规则添加到这些安全组中，以授权适当的入站/出站访问。这样您的 EC2 实例就可以使用标准的 NFSv4 .1 TCP 端口通过挂载目标连接到文件系统。

- 您的 VPC 中的 EC2 实例。
- Amazon EFS 资源：
 - 文件系统。
 - 文件系统的挂载目标。

为了将文件系统挂载到 EC2 实例上，需要在您的 VPC 中创建一个挂载目标。您可以在 VPC 中的每个可用区分别创建一个挂载目标。有关更多信息，请参阅 [Amazon EFS 的工作原理](#)。

然后，在 EC2 实例上测试文件系统。教程结束时的清理步骤提供了删除这些资源的信息。

本教程在美国西部（俄勒冈州）区域（us-west-2）创建所有这些资源。无论 AWS 区域您使用哪种方式，请务必始终如一地使用它。您的所有资源（您的 VPC、EC2 资源和 EFS 资源）必须位于同一 AWS 区域中。

主题

- [先决条件](#)
- [设置 AWS CLI](#)
- [步骤 1：创建 EC2 资源](#)
- [步骤 2：创建 EFS 资源](#)
- [步骤 3：将文件系统挂载到 EC2 实例上并测试](#)
- [步骤 4：清除](#)

先决条件

- 您可以使用您的根凭证 AWS 账户登录控制台并尝试入门练习。但是，AWS Identity and Access Management (IAM) 建议您不要使用您的根证书 AWS 账户。而是在您的账户中创建一个管理员用户，并使用这些凭证来管理您的账户中的资源。有关更多信息，请参阅[用户指南中的为 IAM Identity Cent AWS IAM Identity Center 用户分配 AWS 账户访问权限](#)。

- 您可以使用默认 VPC，也可以使用在您的账户中创建的自定义 VPC。对于本演练，可以使用默认的 VPC 配置。但是，如果您使用自定义 VPC，请验证以下情况：
 - 已启用 DNS 主机名。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 中的 DNS 属性](#)。
 - Internet 网关已连接到您的 VPC。有关更多信息，请参阅 Amazon VPC 用户指南中的 [使用互联网网关连接到互联网](#)。
 - 已配置 VPC 子网来为 VPC 子网中启动的实例请求公有 IP 地址。有关更多信息，请参阅 Amazon VPC 用户指南中的 [您的 VPCs 和子网的 IP 地址](#)。
 - VPC 路由表包含一个规则，以将 Internet 范围的所有流量发送到 Internet 网关。
- 您需要设置 AWS CLI 并添加管理员用户配置文件。

设置 AWS CLI

按照以下说明设置 AWS CLI 和用户配置文件。

要设置 AWS CLI

1. 下载并配置 AWS CLI。有关说明，请参阅《AWS Command Line Interface User Guide》中的 [Get started with the AWS CLI](#)。
2. 设置配置文件。

您将用户凭据存储在 AWS CLI config 文件中。本教程中的示例 CLI 命令指定 adminuser 配置文件。在 config 文件中创建 adminuser 配置文件。也可以在 config 文件中将管理员用户配置文件设置为默认配置文件，如下所示。

```
[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
```

前面的配置文件也设置了默认值 AWS 区域。如果没有在 CLI 命令中指定区域，则假定为 us-west-2 区域。

3. 在命令提示符处输入以下命令来验证设置。两个命令都没有显式提供凭证，所以将使用默认配置文件的凭证。

- 尝试 help 命令

您也可以通过添加 `--profile` 参数来显式指定用户配置文件。

```
aws help
```

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

步骤 1：创建 EC2 资源

在此步骤中，您将执行以下操作：

- 创建两个安全组。
- 在安全组中添加规则以授权额外访问。
- 启动一个 EC2 实例。在下一步中，您将创建一个 EFS 文件系统并将其挂载到该实例上。

步骤 1.1：创建两个安全组

在本部分中，您将在 VPC 中为 EC2 实例和 EFS 挂载目标创建安全组。在教程的稍后部分，您要将这些安全组分配给 EC2 实例和 EFS 挂载目标。有关安全组的信息，请参阅 [Linux 实例的 Amazon EC2 安全组](#)。

创建安全组

1. 使用 `create-security-group` CLI 命令创建两个安全组：
 - a. 为您的 EC2 实例创建一个安全组 (`efs-walkthrough1-ec2-sg`) 并提供您的 VPC ID。

```
$ 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
```

记下安全组 ID。以下为响应示例。

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

您可以使用以下命令查找 VPC ID：

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

- b. 为 EFS 挂载目标创建安全组 (`efs-walkthrough1-mt-sg`)。您需要提供 VPC ID。

```
$ 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
```

记下安全组 ID。以下为响应示例。

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

2. 验证安全组。

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

两个安全组都应当只有一条允许所有出站流量的出站规则。

在下一节中，您将授权额外访问，以便：

- 您能够连接到 EC2 实例。
- 启用 EC2 实例与 EFS 挂载目标之间的流量 (在本教程的稍后部分，您要将这些安全组与它们关联)。

步骤 1.2：向安全组添加规则以授权 inbound/outbound 访问

在该步骤中，您将在安全组中添加规则以授权入站/出站访问。

添加规则

1. 授权与 EC2 实例安全组 (efs-walkthrough1-ec2-sg) 的传入安全 Shell (SSH) 连接，以便可以从任何主机使用 SSH 连接到 EC2 实例。

```
$ 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
```

验证安全组具有您添加的入站和出站规则。

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

2. 授权到 EFS 挂载目标安全组 (efs-walkthrough1-mt-sg) 的入站访问。

在命令提示符处，使用管理员用户配置文件运行以下 AWS CLI `authorize-security-group-ingress` 命令来添加入站规则。

```
$ 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. 确认两个安全组现在都授权了入站访问。

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

```
--region us-west-2
```

步骤 1.3：启动 EC2 实例

在该步骤中，您将启动 EC2 实例。

启动 EC2 实例

1. 收集在启动 EC2 实例时需要提供的以下信息：

- 密钥对名称。有关创建密钥对的说明，请参阅《Amazon EC2 用户指南》中的[为您的 Amazon EC2 实例创建密钥对](#)。
- 要启动的 Amazon 系统映像 (AMI) 的 ID。

用于启动 EC2 实例的 AWS CLI 命令需要将要部署的 Amazon 系统映像 (AMI) 的 ID 作为参数。本练习使用 Amazon Linux HVM AMI。

Note

您可以使用大多数基于 Linux 的通用型 AMIs。如果您使用其他 Linux AMI，请确保使用分发管理器在实例上安装 NFS 客户端。此外，您可能还需要添加一些软件包。

对于亚马逊 Linux HVM AMI，您可以在[亚马逊 Linux AMI IDs](#) 上找到最新版本。您可以按如下方式从亚马逊 Linux AMI IDs 表中选择 ID 值：

- 选择美国西部（俄勒冈）区域。本演练假定您将在美国西部（俄勒冈州）区域（us-west-2）创建所有资源。
- 选择 EBS 支持的 HVM 64 位类型（因为您在 CLI 命令中指定 t2.micro 实例类型，它不支持实例存储）。
- 您为 EC2 实例创建的安全组的 ID。
- AWS 区域。本演练使用 us-west-2 区域。
- 您要在其中启动实例的 VPC 子网的 ID。可以使用 describe-subnets 命令获取子网列表。

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

```

```
--profile adminuser
```

选择子网 ID 后，记下 describe-subnets 结果中的以下值：

- 子网 ID – 创建挂载目标时需要此值。在本练习中，您将在启动了 EC2 实例的同一子网中创建挂载目标。
- 子网的可用区 – 构建挂载目标 DNS 名称时需要此值，用于将文件系统挂载到 EC2 实例上。

2. 运行以下 AWS CLI run-instances 命令启动 EC2 实例。

```
$ 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. 记下 run-instances 命令返回的实例 ID。

4. 您创建的 EC2 实例必须有公有 DNS 名称，以便用来连接 EC2 实例并向其中挂载文件系统。公有 DNS 名称的形式为：

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

运行以下 CLI 命令，并记下公有 DNS 名称。

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

如果未找到公有 DNS 名称，则检查您在其中启动了 EC2 实例的 VPC 的配置。有关更多信息，请参阅 [先决条件](#)。

5. (可选) 将一个名称分配给您创建的 EC2 实例。为此，请添加一个标签，其中键名和值设置为要分配给实例的名称。您可以通过运行以下 AWS CLI create-tags 命令来完成此操作。

```
$ aws ec2 create-tags \  

```

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

步骤 2：创建 EFS 资源

在此步骤中，您将执行以下操作：

- 创建加密 EFS 文件系统。
- 启用生命周期管理
- 在启动了 EFS 实例的可用区创建挂载目标。

步骤 2.1：创建 EFS 文件系统

在该步骤中，您将创建一个 EFS 文件系统。记下 FileSystemId，以便稍后在下一步中为文件系统创建挂载目标时使用。

创建文件系统

- 创建文件系统并添加可选的 Name 标签。
 - a. 在命令提示符处，运行以下 AWS CLI create-file-system 命令。

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

您将收到以下响应。

```
{  
  "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. 记下 FileSystemId 的值。在[步骤 2.3：创建挂载目标](#)中为该文件系统创建挂载目标时需要该值。

步骤 2.2：启用生命周期管理

在该步骤中，您将在文件系统上启用生命周期管理，以便使用 EFS 不频繁访问（IA）存储类。要了解更多信息，请参阅[管理存储生命周期](#)和[EFS 存储类](#)。

启用生命周期管理

- 在命令提示符处，运行以下 AWS CLI put-lifecycle-configuration 命令。

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

您将收到以下响应。

```
{
  "LifecyclePolicies": [
```



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

步骤 2.3：创建挂载目标

在该步骤中，您将在启动了 EC2 实例的可用区中为文件系统创建一个挂载目标。

1. 确保您已获得以下信息：

- 您为其创建挂载目标的文件系统 (例如 fs-example) 的 ID。
- 您在[步骤 1：创建 EC2 资源](#)中启动了 EC2 实例的 VPC 子网 ID。

在本教程中，您在启动了 EC2 实例的同一子网中创建挂载目标，因此您需要子网 ID (例如，subnet-example)。

- 在上一步中您为挂载目标创建的安全组的 ID。

2. 在命令提示符处，运行以下 AWS CLI create-mount-target 命令。

```
$ 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
```

您将收到以下响应。

```
{
  "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. 您还可以使用 `describe-mount-targets` 命令来获取为文件系统创建的挂载目标的描述。

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

步骤 3：将文件系统挂载到 EC2 实例上并测试

在此步骤中，您将执行以下操作：

- 收集所需信息。
- 在 EC2 实例上安装 NFS 客户端。
- 在 EC2 实例上挂载文件系统并测试。

主题

- [步骤 3.1：收集信息](#)
- [步骤 3.2：在 EC2 实例上安装 NFS 客户端](#)
- [步骤 3.3：将文件系统挂载到 EC2 实例上并测试](#)

步骤 3.1：收集信息

在执行本节的步骤时，确保您获取以下信息：

- EC2 实例的公有 DNS 名称，格式如下：

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

- 文件系统的 DNS 名称。您可以使用以下通用形式构建此 DNS 名称：

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

您使用挂载目标在其中挂载文件系统的 EC2 实例可以将文件系统的 DNS 名称解析为挂载目标的 IP 地址。

 Note

Amazon EFS 不要求您的 EC2 实例具有公有 IP 地址或公有 DNS 名称。前面列出的要求仅针对本演练示例，目的是确保您可以使用 SSH 从 VPC 外部连接到实例。

步骤 3.2：在 EC2 实例上安装 NFS 客户端

您可以从运行 Windows、Linux、macOS X 或任何其他 Unix 变体的计算机连接到您的 EC2 实例。

安装 NFS 客户端

1. 连接到您的 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 在 EC2 实例上通过使用 SSH 会话执行以下命令：
 - a. (可选) 获取更新并重启。

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

重启后，重新连接到您的 EC2 实例。

- b. 安装 NFS 客户端。

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

 Note

如果在启动 EC2 实例时选择 Amazon Linux AMI 2016.03.0 Amazon Linux AMI，则不需要安装 `nfs-utils`，因为它已默认包含在此 AMI 中。

步骤 3.3：将文件系统挂载到 EC2 实例上并测试

现在，将文件系统挂载到 EC2 实例上。

1. 创建目录 ("efs-mount-point")。

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

2. 挂载 EFS 文件系统。

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

EC2 实例可以将挂载目标的 DNS 名称解析为 IP 地址。您也可以直接指定挂载目标的 IP 地址。

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

3. 您已经将 EFS 文件系统挂载到 EC2 实例上，接下来就可以创建文件了。

a. 更改目录。

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

b. 列出目录的内容。

```
$ ls -al
```

它应该是空的。

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

c. 刚创建的文件系统的根目录由根用户拥有并且只能由根用户写入，因此您需要更改权限以添加文件。

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

现在，如果您尝试 `ls -al` 命令，可以看到权限已更改。

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

- d. 创建 文本文件。

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

- e. 列出目录的内容。

```
$ ls -l
```

现在，您已成功创建一个 EFS 文件系统并将其挂载到您的 VPC 中的 EC2 实例上。

重启后挂载的文件系统将不复存在。为了自动重新挂载目录，可以使用 `fstab` 文件。如果您使用 Auto Scaling 组来启动 EC2 实例，则也可以在启动配置中设置脚本。

步骤 4：清除

如果不再需要使用创建的资源，应将其删除。可以使用 CLI 删除。

- 移除 EC2 资源（EC2 实例和两个安全组）。当您删除挂载目标时，Amazon EFS 会删除网络接口。
- 删除 EFS 资源（文件系统、挂载目标）。

删除在本演练中创建的 AWS 资源

1. 终止为本教程创建的 EC2 实例。

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

您还可以使用控制台删除 EC2 资源。有关说明，请参阅[终止实例](#)。

2. 删除挂载目标。

只有在删除为文件系统创建的挂载目标后才能删除文件系统。可以使用 `describe-mount-targets` CLI 命令获得挂载目标列表。

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

然后，使用 `delete-mount-target` CLI 命令删除挂载目标。

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

3. (可选) 删除您创建的两个安全组。创建安全组不需要支付费用。

必须先删除挂载目标的安全组，然后再删除 EC2 实例的安全组。挂载目标的安全组包含一个引用 EC2 安全组的规则。因此，不能先删除 EC2 实例的安全组。

有关说明，请参阅《Amazon EC2 User Guide》中的 [Delete your security group](#)。

4. 通过 `delete-file-system` CLI 命令删除文件系统。可以使用 `describe-file-systems` CLI 命令获得文件系统列表。可以从响应中获得文件系统 ID。

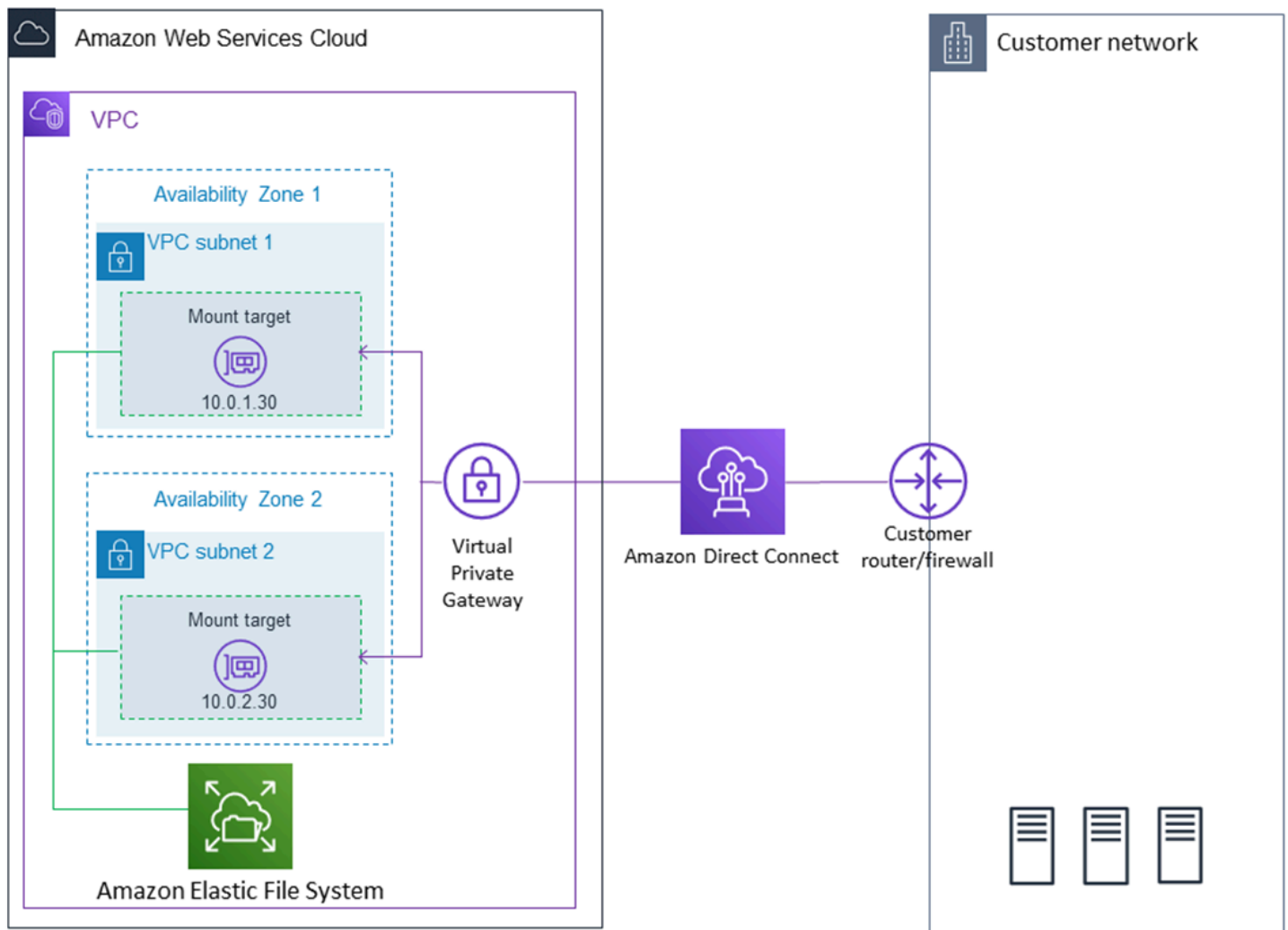
```
aws efs describe-file-systems \  
--profile adminuser \  
--region aws-region
```

通过提供文件系统 ID 删除文件系统。

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

教程：使用本地 Linux 客户端挂载

当通过 Direct Connect 或 VPN 连接到 Amazon VPC 时，您可以将 EFS 文件系统挂载到本地数据中心服务器上。下图显示了从本地安装 EFS 文件系统 AWS 服务 所需的高级示意图。



Note

不支持将 Amazon EFS 与基于 Microsoft Windows 的客户端结合使用。

主题

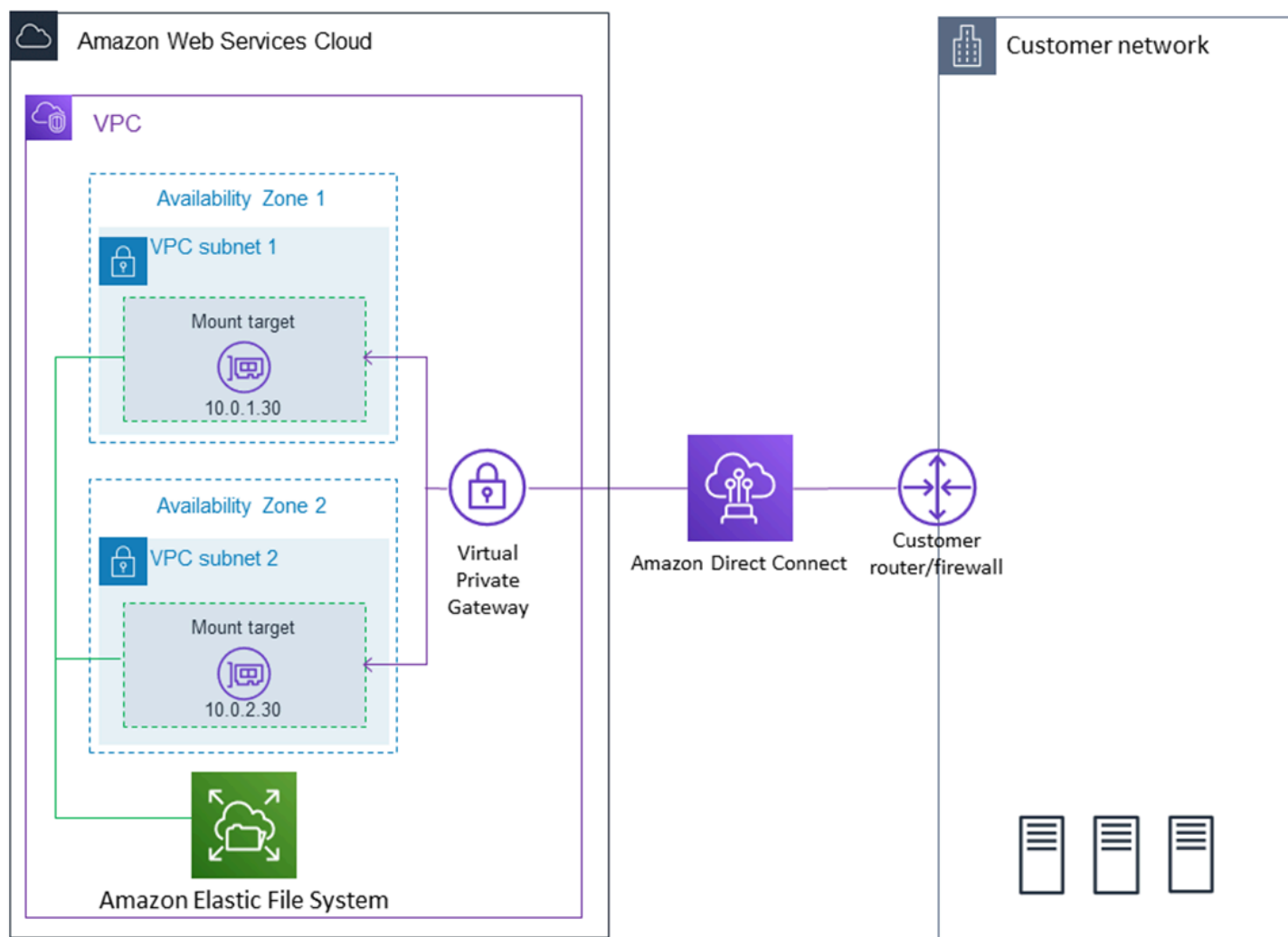
- [先决条件](#)
- [步骤 1：创建您的 EFS 资源](#)
- [步骤 2：安装 NFS 客户端](#)
- [步骤 3：在本地客户端上挂载 EFS 文件系统](#)
- [第 4 步：清理资源并保护您的 AWS 账户](#)
- [可选：加密传输中的数据](#)

先决条件

确保您已经有 Direct Connect 或 VPN 连接。有关 Direct Connect 的更多信息，请参阅 [Direct Connect 用户指南](#)。有关设置 VPN 连接的更多信息，请参阅《Amazon VPC 用户指南》中的[使用 AWS Virtual Private Network 将 VPC 连接到远程网络](#)。

建立 Direct Connect 或 VPN 连接后，在您的 Amazon VPC 中创建一个 EFS 文件系统和一个挂载目标。之后，您就可以下载并安装这些 amazon-efs-utils 工具。接下来，您从本地客户端中测试文件系统。最后，本演练结束时的清理步骤提供了删除这些资源的信息。

此演练在美国西部（俄勒冈州）区域（us-west-2）创建所有这些资源。无论 AWS 区域您使用哪种方式，请务必始终如一地使用它。您的所有资源（您的 VPC、挂载目标和 EFS 文件系统）都必须位于 AWS 区域同一位置，如下图所示。



Note

在某些情况下，您的本地应用程序可能需要知道 EFS 文件系统是否可用。在这些情况下，您的应用程序应能在第一个挂载点暂时不可用时指向其他挂载点 IP 地址。在这种情况下，我们建议您将两个本地客户端通过不同的可用区 (AZs) 连接到您的文件系统，以获得更高的可用性。

您可以使用您的根凭证 AWS 账户 登录控制台并尝试本练习。但是，AWS Identity and Access Management (IAM) 最佳实践建议您不要使用自己的根证书 AWS 账户。而是在您的账户中创建一个管理员用户，并使用这些凭证来管理您的账户中的资源。有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[单点登录访问 AWS 账户](#)。

您可以使用默认 VPC，也可以使用在您的账户中创建的自定义 VPC。对于本演练，可以使用默认的 VPC 配置。但是，如果您使用自定义 VPC，请验证以下情况：

- Internet 网关已连接到您的 VPC。有关更多信息，请参阅《Amazon VPC 用户指南》中的[使用互联网网关启用 VPC 互联网访问](#)。
- VPC 路由表包含一个规则，以将 Internet 范围的所有流量发送到 Internet 网关。

步骤 1：创建您的 EFS 资源

在此步骤中，您将创建 EFS 文件系统和挂载目标。

创建 EFS 文件系统

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 选择创建文件系统。
3. 从 VPC 列表中选择您的默认 VPC。
4. 选中所有可用区对应的复选框。确保它们全都选择了默认子网、自动 IP 地址和默认安全组。这些是您的挂载目标。有关更多信息，请参阅[管理挂载目标](#)。
5. 选择下一步。
6. 命名您的文件系统，选择通用型以作为您的默认性能模式，然后选择下一步。
7. 选择创建文件系统。
8. 从列表中选择您的文件系统，并记下安全组值。在下一个步骤中，您需要用到此值。

您刚刚创建的文件系统具有挂载目标。每个挂载目标都具有一个关联的安全组。该安全组充当虚拟防火墙以控制网络流量。如果您在创建挂载目标时未提供安全组，Amazon EFS 会将 VPC 的默认安全组与之关联。如果您完全按照上述步骤进行操作，则挂载目标使用默认安全组。

下一步，您将向挂载目标的安全组添加一条规则，以允许入站流量进入网络文件系统 (NFS) 端口 (2049)。您可以使用将规则 AWS 管理控制台 添加到您的 VPC 中挂载目标的安全组中。

允许入站流量进入 NFS 端口

1. 登录 AWS 管理控制台 并打开 Amazon EC2 控制台，网址为 <https://console.aws.amazon.com/ec2/>。
2. 在网络与安全下面，选择安全组。
3. 选择与您的文件系统关联的安全组。您在 [步骤 1：创建您的 EFS 资源](#) 的结尾记录了该值。
4. 在安全组列表下面显示的分页窗格中，选择入站选项卡。
5. 选择编辑。
6. 选择添加规则，然后选择以下类型的规则：
 - 类型 – NFS
 - 源 – 任何位置

我们建议您仅使用任何位置源进行测试。您可以创建一个设置为本地客户端 IP 地址的自定义源，或者从客户端本身中使用控制台并选择我的 IP。

Note

您不需要添加出站规则，因为默认出站规则允许所有出站流量。如果没有该默认出站规则，请添加一个出站规则以在 NFS 端口 2049 上打开 TCP 连接，从而将挂载目标安全组指定为目标。

步骤 2：安装 NFS 客户端

在此步骤中，安装 NFS 客户端。

在本地服务器上安装 NFS 客户端

Note

如果您需要在传输中加密数据，请使用 Amazon EFS 挂载帮助程序 `amazon-efs-utils` 而不是 NFS 客户端。有关安装的信息 `amazon-efs-utils`，请参阅“可选：加密传输中的数据”部分。

1. 访问本地客户端的终端。
2. 安装 NFS。

如果您使用的是 Red Hat Linux，请使用以下命令安装 NFS。

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

如果您使用的是 Ubuntu，请使用以下命令安装 NFS。

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

步骤 3：在本地客户端上挂载 EFS 文件系统

创建挂载目录

1. 使用以下命令为挂载点创建目录。

Example

```
mkdir ~/efs
```

2. 选择可用区中的挂载目标的所需 IP 地址。您可以从本地 Linux 客户端中测量延迟。为此，请针对不同可用区中的 EC2 实例的 IP 地址使用基于终端的工具（如 `ping`）以查找具有最低延迟的实例。
- 运行挂载命令以使用挂载目标的 IP 地址挂载文件系统。

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

现已挂载 EFS 文件系统，您可以使用以下过程对其进行测试。

测试 EFS 文件系统连接

1. 使用以下命令将目录更改为您创建的新目录。

```
$ cd ~/efs
```

2. 创建一个子目录，并将该子目录的所有权更改为您的 EC2 实例用户。接下来，使用以下命令导航到该新目录。

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

3. 使用以下命令创建一个文本文件。

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

4. 使用以下命令列出目录内容。

```
$ ls -al
```

这样，将会创建以下文件。

```
-rw-rw-r-- 1 username username 0 Nov 15 15:32 test-file.txt
```

Warning

请在自动挂载文件系统时使用 `_netdev` 选项，它用于指定网络文件系统。如果缺少 `_netdev`，您的 EC2 实例可能会停止响应。出现该结果是因为，需要在计算实例启动其网络后初始化网络文件系统。有关更多信息，请参阅 [自动挂载失败，并且实例没有响应](#)。

第 4 步：清理资源并保护您的 AWS 账户

完成本演练后，或者如果您不想探索这些演练，则应执行如下步骤以清理您的资源并保护您的 AWS 账户。

清理资源并保护您的 AWS 账户

1. 使用以下命令卸载 EFS 文件系统。

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

2. 单击 [Amazon EFS 控制台](#)，打开 Amazon EFS 控制台。
3. 选择要从文件系统列表中删除的 EFS 文件系统。
4. 对于操作，选择删除文件系统。
5. 在永久删除文件系统对话框中，键入要删除的 EFS 文件系统的文件系统 ID，然后选择删除文件系统。
6. 打开位于 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 控制台。
7. 在导航窗格中，选择安全组。
8. 选择您针对本演练向其中添加了规则的安全组的名称。

Warning

不要删除您的 VPC 的默认安全组。

9. 对于操作，请选择编辑入站规则。
10. 选择在添加的入站规则末尾的 X，然后选择保存。

可选：加密传输中的数据

要对传输中的数据进行加密，请使用 Amazon EFS 挂载帮助程序代替 NFS 客户端。amazon-efs-utils

amazon-efs-utils 软件包是一个开源 Amazon EFS 工具集。该 amazon-efs-utils 集合附带挂载助手和工具，可以更轻松地加密传输给 Amazon EFS 的数据。有关此软件包的更多信息，请参阅 [安装 Amazon EFS 客户端](#)。该软件包可从免费下载 GitHub，您可以通过克隆软件包的存储库来获取。

要 amazon-efs-utils 从中进行克隆 GitHub

1. 访问本地客户端的终端。

2. 在终端中，使用以下命令将 amazon-efs-utils 工具克隆 GitHub 到您选择的目录中。

```
git clone https://github.com/aws/efs-utils
```

现已具有该软件包，您可以开始进行安装了。该安装是以不同方式处理的，具体取决于本地客户端的 Linux 发行版。支持以下发行版：

- Amazon Linux 2
- Amazon Linux
- 红帽企业 Linux (以及诸如 CentOS 之类的衍生产品) 版本 8 及更高版本
- Ubuntu 16.04 LTS 和更新版本

amazon-efs-utils 作为 RPM 软件包进行编译和安装

1. 在您的客户端上打开终端，然后导航到包含克隆 amazon-efs-utils 软件包的 GitHub 目录。
2. 使用以下命令构建该软件包。

```
make rpm
```

Note

如果尚未安装 rpm-builder 软件包，请使用以下命令进行安装。

```
sudo yum -y install rpm-build
```

3. 使用以下命令安装 软件包。

```
sudo yum -y install build/amazon-efs-utils*.rpm
```

amazon-efs-utils 作为 deb 软件包进行构建和安装

1. 在您的客户端上打开终端，然后导航到包含克隆 amazon-efs-utils 软件包的 GitHub 目录。
2. 使用以下命令构建该软件包。

```
./build-deb.sh
```

3. 使用以下命令安装 软件包。

```
sudo apt-get install build/amazon-efs-utils*deb
```

安装软件包后，配置 amazon-efs-utils 为在 with Direct Connect 或 VPN 中使用。AWS 区域

配置 amazon-efs-utils 为在您的 AWS 区域

1. 通过 Secure Shell (SSH) 访问您的 EC2 实例的终端，然后使用相应的用户名登录。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 使用所选的文本编辑器打开 `/etc/amazon/efs/efs-utils.conf` 文件。
3. 查找 `"dns_name_format = {fs_id}.efs.{region}.amazonaws.com"` 行。
4. 使用您的 AWS 区域的 ID 更改 `{region}`，例如，`us-west-2`。

要在本地客户端上挂载 EFS 文件系统，请先在本地 Linux 客户端上打开终端。要挂载系统，您需要使用文件系统 ID、其中一个挂载目标的 IP 地址，以及文件系统的 AWS 区域。如果您为文件系统创建了多个挂载目标，则可选择其中任一项。

在具有该信息时，您可以使用三个步骤挂载文件系统：

创建挂载目录

1. 使用以下命令为挂载点创建目录。

Example

```
mkdir ~/efs
```

2. 选择可用区中的挂载目标的所需 IP 地址。您可以从本地 Linux 客户端中测量延迟。为此，请针对不同可用区中的 EC2 实例的 IP 地址使用基于终端的工具（如 ping）以查找具有最低延迟的实例。

更新 `/etc/hosts`

- 在本地 `/etc/hosts` 文件中添加一个具有文件系统 ID 和挂载目标 IP 地址的条目，格式如下所示。

```
mount-target-IP-Address file-system-ID.efs.region.amazonaws.com
```

Example

```
192.0.2.0 fs-12345678.efs.us-west-2.amazonaws.com
```

创建挂载目录

- 使用以下命令为挂载点创建目录。

Example

```
mkdir ~/efs
```

- 运行 `mount` 命令以挂载文件系统。

Example

```
sudo mount -t efs fs-12345678 ~/efs
```

如果要使用传输中的数据加密，`mount` 命令类似于以下内容。

Example

```
sudo mount -t efs -o tls fs-12345678 ~/efs
```

教程：挂载来自其它 VPC 的文件系统

在此教程中，您将设置一个 EC2 实例来挂载位于其它虚拟私有云 (VPC) 中的 EFS 文件系统。您可以使用 EFS 挂载帮助程序执行该操作。挂载帮助程序是 `amazon-efs-utils` 工具集的一部分。有关 `amazon-efs-utils` 的更多信息，请参阅[安装 Amazon EFS 客户端](#)。

必须使用 VPC 对等连接或 VPC 传输网关连接客户端的 VPC 和 EFS 文件系统的 VPC。当您使用 VPC 对等连接或传输网关连接 VPC 时，一个 VPC 中的 EC2 实例可以访问另一个 VPC 中的 EFS 文件系统，即使 VPCs 它们属于不同的账户。

Note

不支持将 Amazon EFS 与基于 Microsoft Windows 的客户端结合使用。

主题

- [先决条件](#)
- [步骤 1：确定挂载目标的可用区 ID](#)
- [步骤 2：确定挂载目标 IP 地址](#)
- [步骤 3：为挂载目标添加主机条目](#)
- [步骤 4：使用 EFS 挂载帮助程序挂载您的文件系统](#)
- [第 5 步：清理资源并保护您的 AWS 账户](#)

先决条件

要完成本教程，您必须满足以下条件：

- 在使用此过程之前，将在 EC2 实例上安装 amazon-efs-utils 工具集。有关安装 amazon-efs-utils 的说明，请参阅[安装 Amazon EFS 客户端](#)。
- 下列情况之一：
 - EFS 文件系统所在的 VPC 与 EC2 实例所在的 VPC 之间的 VPC 对等连接。VPC 对等连接是两 VPCs 者之间的网络连接。这种类型的连接使您能够使用私有 Internet 协议版本 4 (IPv4) 或 Internet 协议版本 6 (IPv6) 地址在它们之间路由流量。您可以使用 VPC 对等互连 VPCs 在同一个 AWS 区域 或两者之间 AWS 区域进行连接。有关更多信息，请参阅《Amazon VPC 对等连接指南》中的[创建并接受 VPC 对等连接](#)。
 - EFS 文件系统所在的 VPC 与 EC2 实例所在的 VPC 之间的中转网关连接。传输网关是一个网络中转枢纽，可用于将您的网络 VPCs 和本地网络互连。有关更多信息，请参阅《Amazon VPC Transit Gateway 指南》中的[开始使用 Amazon VPC 中转网关](#)。

步骤 1：确定挂载目标的可用区 ID

为了确保您的文件系统的高可用性，我们建议您始终使用与 NFS 客户端位于相同可用区的 EC2 挂载目标 IP 地址。如果要挂载其他账户中的 EFS 文件系统，请确保 NFS 客户端和 EFS 挂载目标位于相同的可用区 ID 中。此要求适用，因为可用区名称在账户之间可能会有所不同。

确定 EC2 实例的可用区 ID

1. 连接到您的 EC2 实例。有关更多信息，请参阅《Amazon EC2 用户指南》中的[连接到您的 EC2 实例](#)。
2. 按以下所示，使用 `describe-availability-zones` CLI 命令确定 EC2 实例所在的可用区 ID。

```
[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"
    }
  ]
}
```

可用区 ID 将在 `ZoneId` 属性 `use2-az2` 中返回。

步骤 2：确定挂载目标 IP 地址

您已知道 EC2 实例的可用区 ID，现在可以检索位于同一可用区 ID 中的挂载目标的 IP 地址。

在同一可用区 ID 中确定挂载目标 IP 地址

- 按以下所示，使用 `describe-mount-targets` CLI 命令在 `use2-az2` 可用区 ID 中检索文件系统的挂载目标 IP 地址。

```
$ 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"
    }
  ]
}
```

use2-az2 可用区 ID 中的挂载目标的 IP 地址为 10.0.2.153。

步骤 3：为挂载目标添加主机条目

您现在可以在 EC2 实例上的 `/etc/hosts` 文件中生成一个条目，将挂载目标 IP 地址映射到您的 EFS 文件系统的主机名。

为挂载目标添加主机条目

1. 将挂载目标 IP 地址行添加到 EC2 实例的 `/etc/hosts` 文件中。该条目使用 `mount-target-IP-Address file-system-ID.efs.region.amazonaws.com` 格式。使用以下命令，将该行添加到文件中。

```
echo "10.0.2.153 fs-01234567.efs.us-east-2.amazonaws.com" | sudo tee -a /etc/hosts
```

2. 确保 EC2 实例和挂载目标的 VPC 安全组具有允许根据需要访问 EFS 文件系统的规则。有关更多信息，请参阅 [使用 VPC 安全组](#)。

步骤 4：使用 EFS 挂载帮助程序挂载您的文件系统

要挂载 EFS 文件系统，必须先在 EC2 实例上创建挂载目录。然后，使用 EFS 挂载助手，您可以通过 AWS Identity and Access Management (IAM) 授权或 EFS 接入点挂载文件系统。有关更多信息，请参阅 [使用 IAM 控制文件系统访问](#) 和 [使用接入点工作](#)。

创建挂载目录

- 使用以下命令创建用于挂载文件系统的目录。

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

使用 IAM 授权挂载文件系统

- 使用以下命令，通过 IAM 授权挂载文件系统。

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

使用 EFS 访问点挂载文件系统

- 使用以下命令，通过 EFS 访问点挂载文件系统。

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

现已挂载 EFS 文件系统，您可以使用以下过程对其进行测试。

测试 EFS 文件系统连接

1. 使用以下命令将目录更改为您创建的新目录。

```
$ cd ~/mnt/efs
```

2. 创建一个子目录，并将该子目录的所有权更改为您的 EC2 实例用户。接下来，使用以下命令导航到该新目录。

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

3. 使用以下命令创建一个文本文件。

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

4. 使用以下命令列出目录内容。

```
$ ls -al
```

这样，将会创建以下文件。

```
-rw-rw-r-- 1 username username 0 Nov 15 15:32 test-file.txt
```

您也可以在 `/etc/fstab` 文件中添加条目以自动挂载文件系统。有关更多信息，请参阅 [在现有 EC2 Linux 实例上启用自动挂载](#)。

Warning

请在自动挂载文件系统时使用 `_netdev` 选项，它用于指定网络文件系统。如果缺少 `_netdev`，您的 EC2 实例可能会停止响应。出现该结果是因为，需要在计算实例启动其网络后初始化网络文件系统。有关更多信息，请参阅 [自动挂载失败，并且实例没有响应](#)。

第 5 步：清理资源并保护您的 AWS 账户

完成本教程后，请执行以下步骤以清理资源并保护您的 AWS 账户。

清理资源并保护您的 AWS 账户

1. 使用以下命令卸载 EFS 文件系统。

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

2. 打开 Amazon Elastic File System 控制台，网址为 <https://console.aws.amazon.com/efs/>。
3. 选择要从文件系统列表中删除的 EFS 文件系统。
4. 对于操作，选择删除文件系统。
5. 在永久删除文件系统对话框中，键入要删除的 EFS 文件系统的文件系统 ID，然后选择删除文件系统。
6. 打开位于 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 控制台。
7. 在导航窗格中，选择安全组。
8. 选择您针对本教程向其中添加了规则的安全组的名称。

 Warning

不要删除您的 VPC 的默认安全组。

9. 对于操作，请选择编辑入站规则。
10. 选择在添加的入站规则末尾的 X，然后选择保存。

解决挂载问题

您可以在下文中找到有关排查 EFS 文件系统挂载问题的信息。

在 Windows 实例上挂载文件系统失败

在 Microsoft Windows 上的 Amazon EC2 实例上挂载文件系统失败。

要采取的操作

请勿将 Amazon EFS 与 Windows EC2 实例一起使用，不支持该配置。

服务器拒绝访问

文件系统挂载失败，并显示以下消息：

```
/efs mount.nfs4: access denied by server while mounting 127.0.0.1:/
```

如果您的 NFS 客户端没有挂载文件系统的权限，则可能会出现此问题。

要采取的操作

如果您尝试使用 IAM 挂载文件系统，请确保您在挂载命令中使用了 `-o iam` 或 `-o tls` 选项。这会告诉 EFS 挂载帮助程序将您的凭证传递给 EFS 挂载目标。如果您仍然没有访问权限，请检查您的文件系统策略和身份策略，以确保没有适用于您的连接的 DENY 子句，并且至少有一个适用于连接的 ALLOW 子句。有关更多信息，请参阅[使用 IAM 控制文件系统访问](#)和[创建文件系统策略](#)。

自动挂载失败，并且实例没有响应

如果在实例上自动挂载文件系统，并且未声明 `_netdev` 选项，则可能会出现该问题。如果缺少 `_netdev`，您的 EC2 实例可能会停止响应。出现该结果是因为，需要在计算实例启动其网络后初始化网络文件系统。

要采取的操作

如果出现此问题，请联系 [Supp AWS ort](#)。

在 /etc/fstab 中挂载多个 Amazon EFS 文件系统失败

如果实例使用的 systemd 初始化系统在 `/etc/fstab` 中具有两个或更多 Amazon EFS 条目，有时可能会没有挂载其中的部分或全部条目。在这种情况下，`dmesg` 输出显示类似于以下内容的一行或多行。

```
NFS: nfs4_discover_server_trunking unhandled error -512. Exiting with error EIO
```

要采取的操作

在这种情况下，我们建议您在 `/etc/systemd/system/mount-nfs-sequentially.service` 中创建新的 systemd 服务文件。文件中包含的代码取决于您是手动挂载文件系统，还是使用 Amazon EFS 挂载帮助程序进行挂载。

- 如果要手动装载文件系统，则该 `ExecStart` 命令必须指向网络文件系统 (NFS4)。在此文件中包含以下代码：

```
[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
```

- 如果您使用的是 Amazon EFS 挂载帮助程序，则该ExecStart命令必须指向 EFS，而不是使用传输层安全 (TLS)。NFS4 在此文件中包含以下代码：

```
[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
```

创建此文件后，运行以下两个命令：

1. `sudo systemctl daemon-reload`
2. `sudo systemctl enable mount-nfs-sequentially.service`

然后，重新启动您的 Amazon EC2 实例。将按需挂载文件系统，通常在一秒内。

挂载命令失败，并显示“错误的 fs 类型”错误消息

挂载命令失败，并显示如下错误消息。

```
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.
```

要采取的操作

如果收到该消息，请安装 `nfs-utils`（或 Ubuntu 上的 `nfs-common`）软件包。有关更多信息，请参阅 [安装 NFS 客户端](#)。

挂载命令失败，并显示“不正确的挂载选项”错误消息

挂载命令失败，并显示如下错误消息。

```
mount.nfs: an incorrect mount option was specified
```

要采取的操作

此错误消息很可能意味着您的 Linux 发行版不支持网络文件系统版本 4.0 和 4.1 (NFSv4)。要确认是否属于这种情况，您可以运行以下命令。

```
$ grep CONFIG_NFS_V4_1 /boot/config*
```

如果上述命令返回 `# CONFIG_NFS_V4_1 is not set`，则表明您的 Linux 发行版不支持 NFSv4.1。有关 NFSv4 支持 .1 的亚马逊弹性计算云 (Amazon EC2) 的亚马逊系统映像 (AMI) 列表，请参阅 [NFS 支持](#)。

使用接入点挂载失败

使用接入点进行挂载时，挂载命令失败，并显示以下错误消息：

```
mount.nfs4: mounting access_point failed, reason given by server: No such file or directory
```

要采取的操作

此错误消息表示指定的 EFS 路径不存在。确保您提供接入点根目录的所有权和权限。EFS 将使用此信息创建根目录。有关更多信息，请参阅 [使用接入点工作](#)。

如果未指定任何根目录所有权和权限，并且根目录尚不存在，EFS 将不会创建根目录。发生这种情况时，使用接入点挂载文件系统的任何尝试都将失败。

在创建文件系统后文件系统挂载立即失败

在创建域名服务 (DNS) 记录的挂载目标后，可能需要长达 90 秒的时间才能在整个 AWS 区域中传播。

要采取的操作

如果您以编程方式创建和挂载文件系统（例如使用 CloudFormation 模板），我们建议您实现等待条件。

文件系统挂载挂起，然后失败，并显示超时错误

文件系统挂载命令挂起一两分钟，然后失败，并显示超时错误。下面的代码显示了一个示例。

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

[2+ minute wait here]
mount.nfs: Connection timed out
$
```

要采取的操作

出现该错误的原因可能是，Amazon EC2 实例或挂载目标安全组的配置不正确。确保挂载目标安全组具有支持使用端口 2049 从 EC2 安全组进行 NFS 访问的入站规则。有关更多信息，请参阅 [使用 VPC 安全组](#)。

请验证您所指定的挂载目标 IP 地址是否有效。如果指定的 IP 地址不正确，并且在该 IP 地址中没有任何其他内容以拒绝挂载，则可能会遇到该问题。

使用 NFS 通过 DNS 名称挂载文件系统失败

尝试使用 NFS 客户端（不使用 amazon-efs-utils 客户端）通过文件系统的 DNS 名称挂载文件系统失败，如以下示例所示：

```
$ sudo mount -t nfs -o
  nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport file-
system-id.efs.aws-region.amazonaws.com:/ mnt
mount.nfs: Failed to resolve server file-system-id.efs.aws-region.amazonaws.com:
  Name or service not known.

$
```

要采取的操作

请检查您的 VPC 配置。如果使用自定义 VPC，请确保已启用 DNS 设置。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 的 DNS 属性](#)。此外，文件系统和挂载目标 DNS 名称还无法从它们存在的 VPC 外部进行解析。

您必须先执行以下操作，然后才能在 mount 命令中使用文件系统的 DNS 名称来挂载文件系统：

- 确保 Amazon EC2 实例所在的同一可用区中有一个 Amazon EFS 挂载目标。
- 确保在与 Amazon EC2 实例相同的 VPC 中有一个挂载目标。否则，不能对位于其他 VPC 中的 EFS 挂载目标使用 DNS 名称解析。有关更多信息，请参阅 [从其他 AWS 账户 或 VPC 挂载 EFS 文件系统](#)。
- 在配置为使用由 Amazon 提供的 DNS 服务器的 Amazon VPC 内连接至您的 Amazon EC2 实例。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [Amazon VPC 中的 DHCP 选项集](#)。
- 确保连接 Amazon EC2 实例的 Amazon VPC 已启用 DNS 主机名。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 中的 DNS 属性](#)。

文件系统挂载失败，并显示错误“nfs 未响应”

Amazon EFS 文件系统挂载因传输控制协议 (TCP) 重新连接事件失败，并返回错误 "nfs: server_name still not responding"。

要采取的操作

请使用 noresvport 挂载选项，以确保在重新建立网络连接时，NFS 客户端将使用新的 TCP 源端口。这样做有助于确保在网络恢复事件后具有不间断的可用性。

挂载目标生命周期状态停滞

挂载目标生命周期停滞在正在创建或正在删除状态。

要采取的操作

重试 CreateMountTarget 或 DeleteMountTarget 调用。

挂载目标生命周期状态显示错误

挂载目标生命周期状态显示为错误。

要采取的操作

如果虚拟私有云 (VPC) 的托管区相互冲突，Amazon EFS 无法为新的文件系统挂载目标创建必要的域名系统 (DNS) 记录。Amazon EFS 无法在客户拥有的托管区内创建新记录。如果您需要维护具有冲突的 `efs.<region>.amazonaws.com` DNS 范围的托管区，请在单独的 VPC 中创建托管区。有关 VPC 的 DNS 注意事项的更多信息，请参阅[您的 VPC 的 DNS 属性](#)。

要解决此问题，请从 VPC 中删除冲突的 `efs.<region>.amazonaws.com` 主机，然后重新创建挂载目标。有关删除挂载目标的更多信息，请参阅[管理挂载目标](#)。

挂载没有响应

Amazon EFS 挂载看起来没有响应。例如，`ls` 等命令挂起。

要采取的操作

如果另一个应用程序正在将大量数据写入文件系统，则可能会出现该错误。在该操作完成前，可能会阻止对正在被写入的文件的访问。一般来说，尝试访问正在被写入的文件的任何命令或应用程序均可能会显示为挂起状态。例如，`ls` 命令可能会在访问正在被写入的文件时挂起。出现该结果是因为，某些 Linux 发行版在 `ls` 命令中使用别名，以便检索文件属性以及列出目录内容。

要解决该问题，请验证另一个应用程序是否正在将文件写入 Amazon EFS 挂载，并验证它是否处于 `Uninterruptible sleep (D)` 状态，如下面的示例所示：

```
$ 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
```

在已验证确属这种情况之后，您可以通过等待其他写入操作完成或通过实施一种变通解决办法来解决问题。在 `ls` 示例中，您可以直接使用 `/bin/ls` 命令，而不是使用别名。这样做可以继续执行命令，而不会在写入的文件处挂起。通常，如果写入数据的应用程序可能会定期强制执行数据刷新（可能使用 `fsync(2)`），这样做可能有助于提高文件系统对其他应用程序的响应能力。但是，在应用程序写入数据时，这种改善可能会牺牲性能。

挂载的客户端断开连接

挂载到 Amazon EFS 文件系统的客户端偶尔会由于多种原因而断开连接。NFS 客户端旨在在出现中断时自动重新连接，从而最大限度地减少例行断开连接对应用程序性能和可用性的影响。在大多数情况下，客户端会在几秒钟内以透明的方式重新连接。

但是，旧版 Linux 内核（版本 v5.4 及更低版本）中包含的 NFS 客户端软件包含一种行为，该行为会导致 NFS 客户端在断开连接时尝试在同一 TCP 源端口上重新连接。此行为不符合 TCP RFC 要求，并且可能会阻止这些客户端快速重新建立与其 EFS 服务器（在本例中为 EFS 文件系统）的连接。

要解决此问题，我们强烈建议您使用 Amazon EFS 挂载帮助程序来挂载 EFS 文件系统。EFS 挂载帮助程序使用针对 Amazon EFS 文件系统优化了的挂载设置。有关 EFS 客户端和挂载帮助程序的更多信息，请参阅[安装 Amazon EFS 客户端](#)。

如果您无法使用 EFS 挂载帮助程序，我们强烈建议您使用 `noresvport` NFS 挂载选项，该选项会指示 NFS 客户端使用新的 TCP 源端口重新建立连接，以避免出现此问题。有关更多信息，请参阅[推荐的 NFS 挂载设置](#)。

对新挂载的文件系统的操作返回“坏文件句柄”错误

针对新挂载的文件系统执行的操作返回 bad file handle 错误。

如果 Amazon EC2 实例连接到了一个文件系统和一个具有指定 IP 地址的挂载目标，然后该文件系统和挂载目标被删除，则可能会出现该错误。如果您创建新的文件系统和挂载目标，以连接到具有相同挂载目标 IP 地址的 Amazon EC2 实例，则可能会发生该问题。

要采取的操作

您可以卸载文件系统，然后在 Amazon EC2 实例上重新挂载文件系统以解决该问题。有关卸载您的 Amazon EFS 文件系统的更多信息，请参阅[卸载文件系统](#)。

卸载文件系统失败

如果文件系统繁忙，则无法将其卸载。

要采取的操作

您可以通过以下方法解决该问题：

- 使用延迟卸载 `umount -l`，它会在运行时将文件系统从文件系统层次结构中分离出来，然后在文件系统不再忙碌时立即清理对文件系统的所有引用。
- 等待所有读取和写入操作完成，然后再次尝试执行 `umount` 命令。
- 使用 `umount -f` 命令强制卸载。

Warning

强制卸载将会中断当前为文件系统执行的任何数据读取或写入操作。有关使用此选项的更多信息和指导，请参阅[卸载手册页](#)。

将数据传入和传出 Amazon EFS

您可以使用 AWS DataSync 和 AWS Transfer Family 将数据传入和传出您的 Amazon EFS 文件系统。AWS DataSync 是一种在线数据传输服务，可以在网络文件系统 (NFS)、服务器消息块 (SMB) 文件服务器、自我管理的对象存储之间以及服务之间复制数据。AWS 有关 DataSync 与 Amazon EFS 配合使用的更多信息，请参阅[AWS DataSync 用于传输数据](#)。

AWS Transfer Family 是一项完全托管的 AWS 服务，您可以使用它通过安全文件传输协议 (SFTP)、文件传输协议 (FTP) 和基于安全套接字层 (FTPS) 的 FTP 协议，将文件传入和传出 Amazon EFS 文件系统。使用 Transfer Family，您可以为业务合作伙伴提供访问存储在 Amazon EFS 文件系统中的文件的权限，用于数据分发、供应链、内容管理和网络服务应用程序等用例。有关将 Transfer Family 与 Amazon EFS 结合使用的更多信息，请参阅[AWS Transfer Family 用于传输数据](#)。

主题

- [AWS DataSync 用于传输数据](#)
- [AWS Transfer Family 用于传输数据](#)

AWS DataSync 用于传输数据

AWS DataSync 是一项在线数据传输服务，可简化、自动化和加速本地存储系统之间以及存储服务之间的数据移动和复制。AWS DataSync 可以在网络文件系统 (NFS)、服务器消息块 (SMB) 文件服务器、自我管理的对象存储、A AWS Snowball Edge mazon S3 存储桶、EFS 文件系统 FSx 和 Windows 文件服务器文件系统之间复制数据。

您还可以使用 DataSync 在两个 EFS 文件系统之间传输文件，包括不同文件系统中的文件系统和不同 AWS 区域的 EF AWS 账户 S 拥有的文件系统。使用 DataSync 在 EFS 文件系统之间复制数据，您可以执行一次性数据迁移，为分布式工作负载定期摄取数据，并自动复制以实现数据保护和恢复。

有关更多信息，请参阅 [开始使用 Amazon EFS](#) 和 [AWS DataSync 用户指南](#)。

AWS Transfer Family 用于传输数据

AWS Transfer Family 是一项完全托管的 AWS 服务，您可以使用它通过以下协议将文件传入和传出 Amazon EFS 文件系统：

- Secure Shell (SSH) 文件传输协议 (SFTP) (AWS Transfer for SFTP)
- 安全文件传输协议 (FTPS) (AWS Transfer for FTPS)

- 文件传输协议 (FTP) (AWS Transfer for FTP)

使用 Transfer Family，您可以安全地让第三方（例如您的供应商、合作伙伴或客户）通过支持的协议在全球范围内大规模访问您的文件，而无需管理任何基础架构。此外，您现在还可以使用 SFTP、FTPS 和 FTP 客户端从 Windows、macOS 和 Linux 环境轻松访问您的 EFS 文件系统。这有助于将数据的可访问性扩展到 NFS 客户端和接入点之外，覆盖多个环境中的用户。

使用 Transfer Family 在 Amazon EFS 文件系统中传输数据的核算方式与其他客户端使用量的核算方式相同。有关更多信息，请参阅[吞吐量模式](#)和[Amazon EFS 限额](#)。

要了解更多信息 AWS Transfer Family，请参阅《[AWS Transfer Family 用户指南](#)》。

Note

对于那些拥有 2021 年 1 月 6 日之前创建的 Amazon EFS 文件系统的允许公开访问的策略，则默认情况下会禁用将 Transfer Family 与 Amazon EFS 配合使用。要允许使用 Transfer Family 访问您的文件系统，请联系 支持。

主题

- [AWS Transfer Family 与 Amazon EFS 一起使用的先决条件](#)
- [将 EFS 文件系统配置为 AWS Transfer Family](#)

AWS Transfer Family 与 Amazon EFS 一起使用的先决条件

要使用 Transfer Family 访问您的 Amazon EFS 文件系统中的文件，您的配置必须满足以下条件：

- Transfer Family 服务器和您的 Amazon EFS 文件系统位于同一 AWS 区域中。
- Transfer Family 使用的 IAM 角色必须附加 Amazon EFS 权限（例如 `elasticfilesystem:ClientMount` 和 `elasticfilesystem:ClientWrite`）。有关策略和权限的更多信息，请参阅《AWS Transfer Family 用户指南》中的[创建 IAM 角色和策略](#)。
- （可选）如果 Transfer Family 服务器归其他账户所有，请启用跨账户存取。
 - 确保您的文件系统策略不允许公有访问。有关更多信息，请参阅[屏蔽对 EFS 文件系统的公共访问权限](#)。
 - 修改文件系统策略以启用跨账户存取。有关更多信息，请参阅[配置 Transfer Family 的跨账户存取](#)。

将 EFS 文件系统配置为 AWS Transfer Family

配置 Amazon EFS 文件系统以使用 Transfer Family 需要执行以下步骤：

- 步骤 1：获取分配给 Transfer Family 用户的 POSIX IDs 列表。
- 步骤 2：使用 IDs 分配给 Transfer Family 用户的 POSIX，确保 Transfer Family 用户可以访问文件系统的目录。
- 第 3 步：配置 IAM 以允许访问 Transfer Family 使用的 IAM 角色。

为 Transfer Family 用户设置文件和目录权限

确保 Transfer Family 用户可以访问您的 EFS 文件系统上的必要文件和目录。使用分配给 Transfer Family 用户的 POSIX 列表为目录 IDs 分配访问权限。在此示例中，用户在 Amazon EFS 挂载点下创建一个名为 `transferFam` 的目录。根据您的使用情况，创建目录是可选的。如果需要，您可以在 EFS 文件系统上选择其名称和位置。

为 Transfer Family 的 POSIX 用户分配文件和目录权限

1. 连接到 Amazon EC2 实例。Amazon EFS 仅支持通过基于 Linux 的 EC2 实例进行挂载。
2. 如果 EFS 文件系统尚未挂载在 EC2 实例上，则挂载该文件系统。有关更多信息，请参阅 [挂载 EFS 文件系统](#)。
3. 以下示例将在 EFS 文件系统上创建目录，并将其组更改为 Transfer Family 用户的 POSIX 组 ID，在本示例中为 1101。
 - a. 使用以下命令创建目录 `efs/transferFam`。实际上，您可以在所选文件系统上使用名称和位置。

```
[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. 使用以下命令将 `efs/transferFam` 组更改为分配给 Transfer Family 用户的 POSIX GID。

```
[ec2-user@ip-192-0-2-0 ~]$ sudo chown :1101 efs/transferFam/
```


c. 确认更改。

```
[ec2-user@ip-192-0-2-0 ~]$ ls -l efs
total 0
drwxr-xr-x 2 root 1101 6 Jan  6 15:58 transferFam
```

启用对 Transfer Family 使用的 IAM 角色的访问

在 Transfer Family 中，您可以创建基于资源的 IAM 策略和 IAM 角色，用于定义用户对 EFS 文件系统的访问权限。有关更多信息，请参阅《AWS Transfer Family 用户指南》中的[创建 IAM 角色和策略](#)。您必须使用 IAM 身份策略或文件系统策略向该 Transfer Family IAM 角色授予对您的 EFS 文件系统的访问权限。

以下是授予 IAM 角色 EFS-role-for-transfer ClientMount（读取）和 ClientWrite 访问权限的文件系统策略示例。

JSON

```
{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-8698b356-4212-4d30-901e-ad2030b57762",
  "Statement": [
    {
      "Sid": "Grant-transfer-role-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/EFS-role-for-transfer"
      },
      "Action": [
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/*"
    }
  ]
}
```

有关创建文件系统策略的更多信息，请参阅[创建文件系统策略](#)。有关使用基于身份的 IAM 策略管理对 EFS 资源的访问的更多信息，请参阅[Amazon EFS 基于身份的策略](#)。

配置 Transfer Family 的跨账户存取

如果用于访问您的文件系统的 Transfer Family 服务器属于其他服务器 AWS 账户，则必须授予该帐户访问您的文件系统的权限。此外，您的文件系统策略还必须是非公开的。有关阻止对文件系统的公有访问的更多信息，请参阅[屏蔽对 EFS 文件系统的公共访问权限](#)。

您可以在文件系统策略中授予对文件系统的不同 AWS 账户 访问权限。在 Amazon EFS 控制台中，使用文件系统策略编辑器的“授予额外权限”部分来指定您授予的文件系统访问权限 AWS 账户 和级别。有关创建或编辑文件系统策略的更多信息，请参阅[创建文件系统策略](#)。

您可以使用账户 ID 或账户 Amazon 资源名称 (ARN) 指定账户。有关更多信息 ARNs，请参阅 [IAM 用户指南 ARNs 中的 IAM](#)。

以下是一个非公开文件系统策略示例，此策略授予对文件系统的跨账户存取权限。它包含以下两个声明：

1. 第一个声明 NFS-client-read-write-via-fsmt 向使用文件系统挂载目标访问文件系统的 NFS 客户端授予读取、写入和根权限。
2. 第二条语句仅授予对 AWS 账户 111122223333 的读写权限，该账户拥有 Transfer Family 服务器，需要在您的账户中访问此 EFS 文件系统。Grant-cross-account-access

```
{
  "Statement": [
    {
      "Sid": "NFS-client-read-write-via-fsmt",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ],
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    }
  ],
}
```

```

    {
      "Sid": "Grant-cross-account-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ]
    }
  ]
}

```

以下文件系统策略添加了一条声明，授予对 Transfer Family 使用的 IAM 角色的访问权限。

```

{
  "Statement": [
    {
      "Sid": "NFS-client-read-write-via-fsmt",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ],
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    },
    {
      "Sid": "Grant-cross-account-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ]
    }
  ],
}

```

```
{
  "Sid": "Grant-transfer-role-access",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/EFS-role-for-transfer"
  },
  "Action": [
    "elasticfilesystem:ClientWrite",
    "elasticfilesystem:ClientMount"
  ]
}
```

管理 EFS 文件系统

文件系统管理任务包括使用挂载目标管理文件系统的网络可访问性、更改文件系统吞吐量模式、更新文件系统生命周期策略以及管理文件系统加密。

您可以使用执行这些文件系统管理任务 AWS 管理控制台，也可以使用 AWS Command Line Interface (AWS CLI) 或 API 以编程方式执行这些文件系统管理任务，如以下各节所述。

主题

- [了解文件系统状态](#)
- [管理挂载目标](#)
- [管理文件系统吞吐量](#)
- [管理存储生命周期](#)

了解文件系统状态

可以使用 Amazon EFS 控制台或 AWS CLI 查看 Amazon EFS 文件系统的状态。Amazon EFS 文件系统可以具有下表中所述的状态值之一。

文件系统状态	说明
AVAILABLE	文件系统处于正常状态，可以访问并可供使用。
CREATING	Amazon EFS 正在创建新文件系统。
DELETING	Amazon EFS 正在删除文件系统，以响应用户发起的删除请求。有关更多信息，请参阅 删除 EFS 文件系统 。
DELETED	Amazon EFS 已删除文件系统，以响应用户发起的删除请求。有关更多信息，请参阅 删除 EFS 文件系统 。
UPDATING	文件系统正在进行更新，以响应用户发起的更新请求。
ERROR	适用于单区文件系统，包括复制配置中的文件系统。

文件系统状态	说明
	文件系统处于故障状态，但可恢复。要访问文件系统数据，请将相关文件系统的备份还原到新文件系统。有关更多信息，请参阅 在 Amazon EFS 中备份和复制数据 。

管理挂载目标

您可以使用为文件系统创建的一个或多个挂载目标，将文件系统挂载到 Amazon EC2 或虚拟私有云 (VPC) 中的其他 AWS 计算实例上。您可以在创建文件系统时或创建文件系统之后创建挂载目标。

为文件系统创建挂载目标后，您可以创建其他挂载目标、删除挂载目标以及修改挂载目标的安全组。如果要修改挂载目标的 VPC，则需要先删除现有的挂载目标。

Note

您不能更改现有挂载目标的 IP 地址。要更改 IP 地址，需要删除挂载目标并使用新地址创建一个新的挂载目标。

主题

- [挂载目标和可用区](#)
- [创建挂载目标](#)
- [删除挂载目标](#)
- [更改挂载目标 VPC](#)
- [更改挂载目标安全组](#)

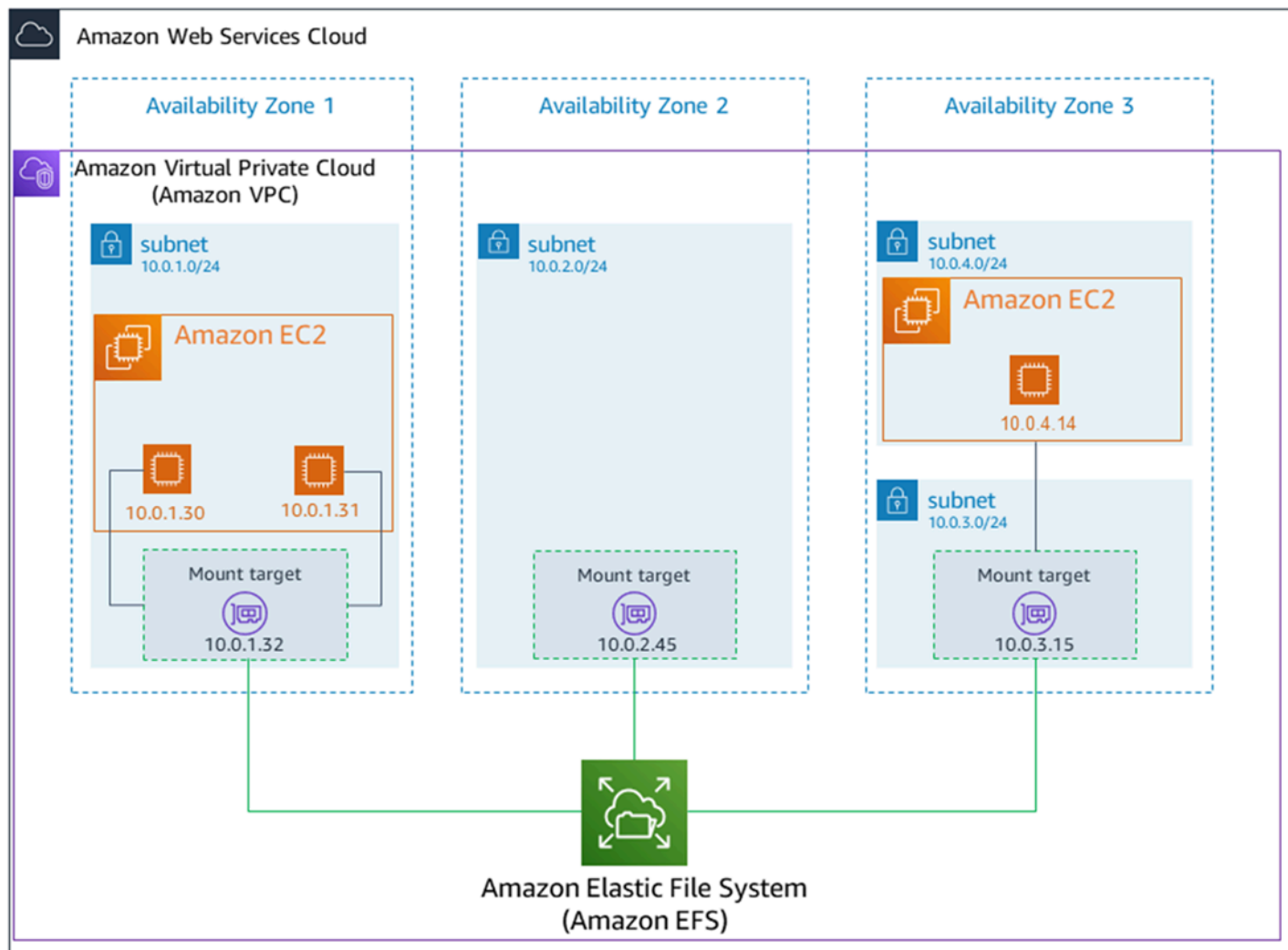
挂载目标和可用区

对于使用区域性存储类别的 EFS 文件系统，可以在 AWS 区域的每个可用区中创建挂载目标。

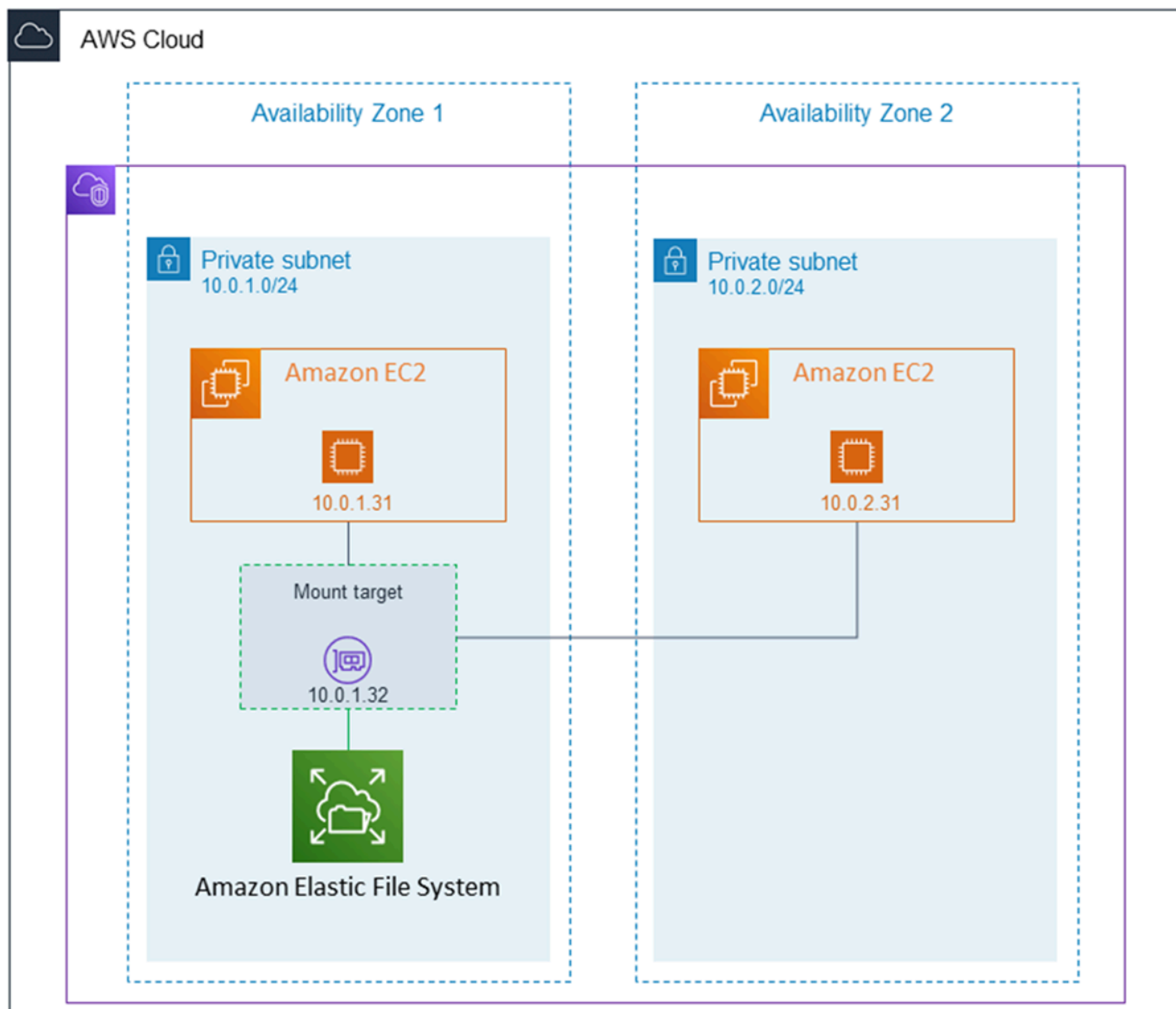
对于单区文件系统，只能在与文件系统相同的可用区中创建单个挂载目标。然后，您可以将文件系统挂载到计算实例上，包括 Amazon EC2、Amazon ECS 和 AWS Lambda 您的虚拟私有云 (VPC)。

下图显示了一个区域性文件系统，带有在 VPC 中的所有可用区中创建的挂载目标。该图显示了在访问 EFS 文件系统的不同 VPC 子网中启动的三个 EC2 实例。该图还显示了每个可用区中的一个挂载目标（不考虑每个可用区中的子网数）。

每个可用区只能创建一个挂载目标。如果可用区具有多个子网，如图中的其中一个区域所示，则只能在其中一个子网中创建挂载目标。只要您在可用区中有一个挂载目标，在其任一子网中启动的 EC2 实例就可以共享该同一挂载目标。



下图显示一个单区文件系统，在与文件系统相同的可用区中创建了单个挂载目标。使用 us-west2c 可用区中的 EC2 实例访问文件系统会产生数据访问费用，因为该实例与挂载目标位于不同的可用区中。



挂载目标安全组充当控制流量的虚拟防火墙。例如，它决定哪些客户端可以访问文件系统。本节介绍以下内容：

- 管理挂载目标安全组和启用流量。
- 将文件系统挂载到客户端上。
- NFS 级权限注意事项。

最初，只有 Amazon EC2 实例上的根用户拥有文件系统的 read-write-execute 权限。本主题讨论 NFS 级权限并提供显示如何在常见场景中授予权限的示例。有关更多信息，请参阅 [网络文件系统 \(NFS\) 级别用户、组和权限](#)。

创建挂载目标

要访问 VPC 中的 EFS 文件系统，您需要为文件系统创建挂载目标。

对于 EFS 文件系统，必须满足以下条件：

- 一次只能为一个 VPC 中的文件系统创建挂载目标。如果要从另一个 VPC 访问文件系统，则需要从当前 VPC 中删除挂载目标，然后在另一个 VPC 中创建新的挂载目标。有关更多信息，请参阅 [更改挂载目标 VPC](#)。
- 如果 VPC 在可用区中有多个子网，则您只能在其中一个子网中创建挂载目标。可用区中的所有 EC2 实例可以共享单个挂载目标。
- 至少应在您希望从中访问文件系统的每个可用区中创建一个挂载目标。

Note

使用在一个可用区中创建的挂载目标在另一个可用区中的 EC2 实例上挂载文件系统时，需要考虑成本。有关更多信息，请参阅 [Amazon EFS 定价](#)。此外，通过始终使用实例可用区本地的挂载目标，可以消除部分故障情况。如果挂载目标的区域发生故障，则无法通过该挂载目标访问文件系统。

您可以使用、为文件系统创建挂载目标 AWS 管理控制台 AWS CLI，也可以使用以编程方式为文件系统创建挂载目标。AWS SDKs 在控制台中，您可以在创建文件系统时或创建文件系统之后创建挂载目标。有关在创建文件系统时创建挂载目标的说明，请参阅 [使用控制台以自定义方式创建](#)。

使用控制台

按照以下过程，为现有 EFS 文件系统添加挂载目标。

在 EFS 文件系统上创建挂载目标

1. 打开 Amazon Elastic File System 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 在左侧导航窗格中，选择文件系统，然后选择要更改 VPC 的文件系统。
3. 选择网络，然后选择管理以显示文件系统的挂载目标。
4. 选择要为其添加挂载目标的文件系统的名称或文件系统 ID 来选择该文件系统。

Note

对于单区文件系统，只能创建单个挂载目标，它位于与文件系统相同的可用区中。

5. 对于使用 EFS 区域存储类别的文件系统，请为要为文件系统创建的每个挂载目标选择添加挂载目标。
6. 定义挂载目标设置：
 - a. 为挂载目标选择可用区和子网 ID。
 - b. 对于 IP 地址类型，请选择 IPv4 仅支持 IPv4 地址，IPv6 仅支持 IPv6 地址，或者选择双栈以同时支持 IPv4 和 IPv6 地址。

Note

IP 地址类型必须与子网的 IP 类型相匹配。此外，IP 地址类型会覆盖子网的 IP 寻址属性。例如，如果 IP 地址类型为 IPv4-only 且您的子网启用了 IPv6 寻址属性，则在子网中创建的网络接口将接收子网范围内 IPv4 的地址。有关更多信息，请参阅[修改子网的 IP 寻址属性](#)。

- c. 如果您知道要放置挂载目标的 IP 地址，请在与 IP 地址类型匹配的 IP 地址框中输入该地址。如果未指定值，Amazon EFS 会从指定的子网中选择未使用的 IP 地址。

Note

在创建挂载目标后，您将无法更改其 IP 地址。要更改 IP 地址，需要删除挂载目标并使用新地址创建一个新的挂载目标。

7. 至少选择一个要与挂载目标关联的安全组。您可以稍后[修改安全组](#)。
8. 选择保存。

使用 AWS CLI

本节提供了在 AWS CLI 中使用 `create-mount-target` 命令创建复制配置的示例。等效的 API 命令是 [CreateMountTarget](#)。

- 如果您没有为挂载目标指定 IP 地址类型，则使用 IPv4-only。
- 如果您没有为挂载目标指定 IP 地址，Amazon EFS 将分配指定子网中可用的地址。

- IP 地址类型会覆盖子网的 IP 寻址属性。例如，如果 IP 地址类型为 IPv4-only 且您的子网启用了 IPv6 寻址属性，则在子网中创建的网络接口将接收子网范围内 IPv4 的地址。有关更多信息，请参阅[修改子网的 IP 寻址属性](#)。

Note

对于单区文件系统，只能创建单个挂载目标，它位于与文件系统相同的可用区中。

示例：在子网上的可用 IPv4 地址创建挂载目标

以下命令为挂载目标指定文件系统、子网和安全组。目标是在指定子网的可用 IPv4 地址上创建的。

```
$ 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 \  

```

以下示例显示了带有示例数据的命令。

```
$ aws efs create-mount-target \  
--file-system-id fs-0123456789abcdef1 \  
--subnet-id subnet-b3983dc4 \  
--security-group sg-01234567 \  
--region us-east-2 \  

```

成功创建挂载目标后，Amazon EFS 以 JSON 形式返回挂载目标描述，如以下示例所示。

```
{  
  "OwnerId": "111122223333"  
  "MountTargetId": "fsmt-f9a14450",  
  "FileSystemId": "fs-0123456789abcdef1",  
  "SubnetId": "subnet-b3983dc4",  
  "LifecycleState": "available",  
  "IpAddress": "10.0.1.24",  
  "NetworkInterfaceId": "eni-3851ec4e",  
  "AvailabilityZoneId": "use2-az1",  
  "AvailabilityZoneName": "us-east-2a",  
  "VpcId": "vpc-3c39ef57"  
}
```

```
}
```

示例：在特定 IPv4 地址创建挂载目标

以下命令指定要用于挂载目标的文件系统、子网、安全组 and IPv4 地址。目标是在指定子网的指定 IPv4 地址创建的。

```
$ aws efs create-mount-target \  
--file-system-id file-system-id \  
--subnet-id subnet-id \  
--security-group ID-of-the-security-group-created-for-mount-target \  
--ip-address IPv4-address \  
--region aws-region \  

```

以下示例显示了带有示例数据的命令。

```
$ aws efs create-mount-target \  
--file-system-id fs-0123456789abcdef1 \  
--subnet-id subnet-b3983dc4 \  
--security-group sg-01234567 \  
--ip-address 10.0.1.24 \  
--region us-east-2 \  

```

成功创建挂载目标后，Amazon EFS 以 JSON 形式返回挂载目标描述，如以下示例所示。

```
{  
  "OwnerId": "111122223333"  
  "MountTargetId": "fsm-t-f9a14450",  
  "FileSystemId": "fs-0123456789abcdef1",  
  "SubnetId": "subnet-b3983dc4",  
  "LifeCycleState": "available",  
  "IpAddress": "10.0.1.24",  
  "NetworkInterfaceId": "eni-3851ec4e",  
  "AvailabilityZoneId": "use2-az1",  
  "AvailabilityZoneName": "us-east-2a",  
  "VpcId": "vpc-3c39ef57"  
}
```

示例：在特定 IPv6 地址创建挂载目标

以下命令指定要用于挂载目标的文件系统、子网、安全组 and IPv6 地址。目标是在指定子网的指定 IPv6 地址创建的。

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id subnet-id \
--security-group ID-of-the-security-group-created-for-mount-target \
--ip-address-type IP-address-type \
--ipv6-address IPv6-address \
--region aws-region \
```

以下示例显示了带有示例数据的命令。

```
$ aws efs create-mount-target \
--file-system-id fs-0123456789abcdef1 \
--subnet-id subnet-b3983dc4 \
--security-group sg-01234567 \
--ip-address-type IPV6_ONLY \
--ipv6-address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 \
--region us-east-2 \
```

成功创建挂载目标后，Amazon EFS 以 JSON 形式返回挂载目标描述，如以下示例所示。

```
{
  "OwnerId": "111122223333"
  "MountTargetId": "fsm-t-f9a14450",
  "FileSystemId": "fs-0123456789abcdef1",
  "SubnetId": "subnet-b3983dc4",
  "LifecycleState": "available",
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "NetworkInterfaceId": "eni-3851ec4e",
  "AvailabilityZoneId": "use2-az1",
  "AvailabilityZoneName": "us-east-2a",
  "VpcId": "vpc-3c39ef57"
}
```

示例：在双栈子网的可用 IPv4 地址上创建挂载目标

该命令为挂载目标指定文件系统、子网、安全组、双栈 IP IPv6 地址类型和地址。目标是在双栈子网上的可用 IPv4 地址和指定 IPv6 地址创建的。

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id subnet-id \
--security-group ID-of-the-security-group-created-for-mount-target \
```

```
--ip-address-type IP-address-type
--ipv6-address IPv6-address \
--region aws-region \
```

以下示例显示了带有示例数据的命令。

```
$ aws efs create-mount-target \
--file-system-id fs-0123456789abcdef1 \
--subnet-id subnet-b3983dc4 \
--security-group sg-01234567 \
--ip-address-type DUAL_STACK \
--ipv6-address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 \
--region us-east-2 \
```

成功创建挂载目标后，Amazon EFS 以 JSON 形式返回挂载目标描述，如以下示例所示。

```
{
  "OwnerId": "111122223333"
  "MountTargetId": "fsm-t-f9a14450",
  "FileSystemId": "fs-0123456789abcdef1",
  "SubnetId": "subnet-b3983dc4",
  "LifeCycleState": "available",
  "IpAddress": "10.0.1.24",
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "NetworkInterfaceId": "eni-3851ec4e",
  "AvailabilityZoneId": "us-east-2a",
  "AvailabilityZoneName": "us-east-2a",
  "VpcId": "vpc-3c39ef57"
}
```

删除挂载目标

删除挂载目标时，则操作将强制中断文件系统的任何挂载，这可能会中断使用这些挂载的实例或应用程序。为避免应用程序中断，请在删除挂载目标之前停止应用程序并卸载文件系统。有关更多信息，请参阅 [卸载文件系统](#)。

您可以使用、删除文件系统的挂载目标 AWS 管理控制台 AWS CLI，也可以使用以编程方式删除文件系统的挂载目标。AWS SDKs

使用控制台

按照以下过程，为现有 EFS 文件系统删除挂载目标。

删除 EFS 文件系统上的挂载目标

1. 卸载文件系统。有关说明，请参阅[卸载文件系统](#)。
2. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
3. 在左侧导航窗格中，选择文件系统，然后选择要删除挂载目标的文件系统。
4. 选择网络，然后选择管理以显示文件系统的挂载目标。
5. 对于要删除的每个挂载目标，请选择移除。
6. 选择保存。

使用 AWS CLI

要删除现有的挂载目标，请使用 `delete-mount-target` AWS CLI 命令（相应的操作是 [DeleteMountTarget](#)），如下所示。

Note

在删除挂载目标之前，先卸载文件系统。

```
$ aws efs delete-mount-target \  
--mount-target-id mount-target-ID-to-delete \  
--region aws-region-where-mount-target-exists
```

以下是使用示例数据的示例：

```
$ aws efs delete-mount-target \  
--mount-target-id fsmt-5751852e \  
--region us-east-2 \  

```

更改挂载目标 VPC

一次只能在一个 VPC 中基于 Amazon VPC 服务使用 EFS 文件系统。也就是说，您在 VPC 中为文件系统创建挂载目标，并使用这些挂载目标提供对该文件系统的访问权限。

您可以从这些目标挂载 EFS 文件系统：

- 同一 VPC 中的 Amazon EC2 实例

- VPC 中通过 VPC 对等连接的 EC2 实例
- 本地服务器，使用 AWS Direct Connect
- 使用 Amazon VPC 通过 AWS 虚拟专用网络 (VPN) 实现本地服务器

VPC 对等连接是两者之间的网络连接 VPCs，使您能够在它们之间路由流量。该连接可以使用私有互联网协议版本 4 (IPv4) 或版本 6 (IPv6)。有关 Amazon EFS 如何使用 VPC 对等连接的更多信息，请参阅[从其他 AWS 账户 或 VPC 挂载 EFS 文件系统](#)。

更改文件系统 VPC

以下步骤可用于更改 EFS 文件系统网络配置的 VPC。

1. 删除分配给文件系统的挂载目标。有关说明，请参阅[删除挂载目标](#)。
2. 当每个挂载目标的挂载目标状态为已删除时，请分配新的 VPC 并为文件系统创建新的挂载目标。有关说明，请参阅[创建挂载目标](#)。

更改挂载目标安全组

安全组定义入站和出站访问。当您更改与挂载目标相关联的安全组时，请确保您授权必要的入站和出站访问。这样做可以使您的 EC2 实例与文件系统进行通信。有关安全组的更多信息，请参阅[使用 VPC 安全组](#)。

您可以使用 AWS 管理控制台、AWS CLI 或以编程方式使用为文件系统的挂载目标添加或删除安全组。AWS SDKs

使用控制台

修改挂载目标的安全组

按照以下过程，为现有 EFS 文件系统添加或移除挂载目标安全组。

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 在左侧导航窗格中，选择文件系统，然后选择要管理挂载目标的文件系统。
3. 选择网络，然后选择管理以显示文件系统的挂载目标。
4. 要从挂载目标中移除安全组，请选择安全组 ID 旁边的 X。
5. 要向挂载目标添加安全组，请从安全组列表中选择安全组。

6. 选择保存。

使用 AWS CLI

要修改对挂载目标有效的安全组，请使用 `modify-mount-target-security-group` AWS CLI 命令（对应的操作是 [ModifyMountTargetSecurityGroups](#)）替换任何现有的安全组，如下所示。

```
$ 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
```

以下是使用示例数据的示例：

```
$ aws efs modify-mount-target-security-groups \  
--mount-target-id fsmt-5751852e \  
--security-groups sg-1004395a sg-1114433a \  
--region us-east-2
```

管理文件系统吞吐量

弹性吞吐量是 Amazon EFS 控制台中默认的吞吐量模式，建议用于大多数使用案例。使用弹性吞吐量，性能可以自动纵向扩展或缩减，以满足工作负载活动的需要。但是，如果您知道工作负载的特定访问模式（包括吞吐量、延迟和存储需求），可以选择更改吞吐量模式。

Note

虽然 Elastic 吞吐量旨在根据您的吞吐量弹性扩展，但我们建议您通过使用 CloudWatch（计量 IOBytes）监控指标和使用情况警报来实施适当的治理，以此作为最佳运营实践的一部分。这有助于保持资源的最优利用率，并确保始终保持在计划的操作参数范围内。有关更多信息，请参阅 [使用 Amazon 监控指标 CloudWatch](#)。

可以选择的其他吞吐量模式包括：

- 预调配吞吐量 – 可以指定文件系统可以驱动的吞吐量级别，不受文件系统大小或突增点数余量影响。
- 突增吞吐量 – 吞吐量可随文件系统中的存储量而扩展，每天最多支持 12 小时突增到更高水平。

有关 Amazon EFS 吞吐量模式的更多信息，请参阅[吞吐量模式](#)。

Note

文件系统可用后，您可以更改吞吐量模式和预调配吞吐量。更改吞吐量模式不会导致应用程序停机。但是，每当您将文件系统更改为预调配吞吐量模式或增加预调配吞吐量时，都必须等待至少 24 小时，才能再次更改吞吐量模式或减少预调配吞吐量。

可以使用 Amazon EFS 控制台、AWS Command Line Interface (AWS CLI) 和 Amazon EFS API 来管理文件系统吞吐量模式。

使用控制台

管理文件系统吞吐量

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 在左侧导航窗格中，选择文件系统以显示账户中的 EFS 文件系统列表。
3. 选择希望为其更改吞吐量模式的文件系统。
4. 在文件系统详细信息页面的常规部分中，选择编辑。将显示编辑页面。
5. 修改吞吐量模式设置。

- 要使用弹性吞吐量或预调配吞吐量，请选择增强型，然后选择弹性或预调配。

如果选择预调配，然后在预调配吞吐量 (MiB/s) 中输入要为文件系统请求预调配的吞吐量。最大读取吞吐量显示为您输入的吞吐量的三倍。EFS 文件系统以其他请求三分之一的速率计量读取请求。输入吞吐量后，将显示文件系统每月成本的估计值。

Note

文件系统可用后，您可以更改吞吐量模式和预配置吞吐量。但是，每当您将文件系统吞吐量更改为预置或增加预置吞吐量时，都必须等待至少 24 小时，才能再次更改吞吐量模式或减少预置吞吐量。

- 要使用突发吞吐量，请选择突发。

有关根据性能需求选择正确的吞吐量模式的更多信息，请参阅[吞吐量模式](#)。

6. 选择保存更改以实施您的更改。

使用 AWS CLI

使用 [update-file-system](#) CLI 命令或 [UpdateFileSystem](#) API 操作更改文件系统的吞吐量模式。

管理存储生命周期

您可以管理文件系统，使其在整个生命周期中经济高效地存储。使用生命周期管理，根据文件系统的生命周期配置，自动在存储类之间转换数据。生命周期配置由您为文件系统设置的三个生命周期策略组成。

生命周期策略指示生命周期管理何时将文件转入和转出 EFS 不频繁访问（IA）存储类和 EFS 存档存储类。转换时间基于上次访问标准存储类中的文件的时间。为了确定标准存储类中的上次访问时间，内部计时器会跟踪上次访问文件的时间（而不是可公开查看的 POSIX 文件系统属性）。每次访问标准存储中的文件时，都会重置生命周期管理计时器。

生命周期策略适用于整个 EFS 文件系统。

EFS 生命周期策略为：

- 转换为 IA - 指示生命周期管理何时将文件移至不频繁访问（IA）存储类别，该存储针对每个季度仅访问几次的数据进行了成本优化。默认情况下，标准存储类别中在 30 天内未访问的文件会转换为 IA。
- 转换为“归档”- 指示生命周期管理何时将标准或 IA 存储类别中的文件移至存档存储类别，该存储类针对每年仅访问几次或更少次的数据进行了成本优化。默认情况下，标准存储类别中在 90 天内未访问的文件会转换为存档存储类别。
- 转换为“标准”- 指示生命周期管理是否在 IA 或存档存储类别中的文件被访问时，将文件从 IA 或归档转换回标准存储类别。默认情况下，在访问文件时，文件不会移回标准存储类别，而是保留为 IA 或存档存储类别。

对于需要最快延迟性能的性能敏感型用例（例如处理大量小文件的应用程序），请选择第一次访问时将文件转换为标准存储。

有关为文件系统配置生命周期策略的更多信息，请参阅[配置生命周期策略](#)。

生命周期管理的文件系统操作

生命周期管理的文件系统操作的优先级低于 EFS 文件系统工作负载操作的优先级。将文件转入和转出 IA 和存档存储所需的时间取决于文件大小和文件系统工作负载。例如，转换数百万个小文件所花费的时间可能比转换数量较少、总存储大小相同的大型文件所花费的时间更长。

文件元数据（包括文件名、所有权信息和文件系统目录结构）始终存储在“标准”中，以帮助确保一致的元数据性能。

对 IA 或存档存储中的文件系统执行的元数据操作（例如列出目录内容等），不算文件访问。在将文件的内容转换为 IA 或存档存储类的过程中，文件将存储在标准存储类中，并按该存储费率计费。

对文件系统的 IA 或存档存储类中文件的所有写入操作都将首先写入标准存储类，然后在 24 小时后有资格转换到适用的存储类。

配置生命周期策略

使用创建具有推荐设置的 EFS 文件系统时 AWS 管理控制台，文件系统会自动使用以下默认生命周期配置进行配置：

- 转换为 IA 设置为自上次访问后的 30 天。
- 转换为归档设置为自上次访问后的 90 天。
- 转换为标准设置为无。

在使用使用自定义设置创建文件系统时，AWS 管理控制台 或者使用创建文件系统时，您可以更改默认生命周期策略 AWS CLI。或者，您可以在创建文件系统之后更改策略，如以下过程所述。

使用控制台

您可以使用 AWS 管理控制台 为现有文件系统设置生命周期策略。

1. 登录 AWS 管理控制台 并打开 Amazon EFS 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 选择文件系统以显示账户中的文件系统列表。
3. 选择要修改生命周期策略的文件系统。
4. 在文件系统详细信息页面的常规部分中，选择编辑。将显示编辑页面。
5. 对于生命周期管理，请配置生命周期策略：
 - 将转换为 IA 设置为可用选项之一。要停止将文件移入 IA 存储，请选择无。
 - 将转换为“归档”设置为可用选项之一。要停止将文件移入存档存储，请选择无。
 - 将转换为标准设置为第一次访问时，以在对 IA 存储中的文件进行非元数据操作访问时将其移动到标准存储。

要停止在第一次访问时将文件从 IA 或归档移到标准存储，请选择无。

6. 选择保存更改以保存您的更改。

使用 AWS CLI

您可以使用 AWS CLI 来设置或修改文件系统的生命周期策略。

- 运行 [put-lifecycle-configuration](#) AWS CLI 命令或 [PutLifecycleConfiguration](#) API 命令，指定要管理生命周期管理的文件系统的文件系统 ID。

```
$ 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
```

您将收到以下响应。

```
{
  "LifecyclePolicies": [
    {
      "TransitionToIA": "AFTER_60_DAYS"
    },
    {
      "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
    },
    {
      "TransitionToArchive": "AFTER_90_DAYS"
    }
  ]
}
```

为现有文件系统停止生命周期管理 (CLI)

- 运行 `put-lifecycle-configuration` 命令，指定要停止生命周期管理的文件系统的文件系统 ID。将 `--lifecycle-policies` 属性留空。

```
$ aws efs put-lifecycle-configuration \
```

```
--file-system-id File-System-ID \  
--lifecycle-policies \  
--region us-west-2 \  
--profile adminuser
```

您将收到以下响应。

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

监控 Amazon EFS

监控是维护 Amazon EFS 和您的 AWS 解决方案的可靠性、可用性和性能的重要组成部分。我们建议您从 AWS 解决方案的所有部分收集监控数据，以便在出现多点故障时可以更轻松地进行调试。不过，在开始监控 Amazon EFS 之前，应制定一个监控计划并在计划中回答下列问题：

- 监控目的是什么？
- 您将监控哪些资源？
- 监控这些资源的频率如何？
- 您将使用哪些监控工具？
- 谁负责执行监控任务？
- 出现错误时应通知谁？

下一步，通过在不同时间和不同负载条件下测量性能，在您的环境中建立正常 Amazon EFS 性能的基准。监控 Amazon EFS 时，请考虑存储历史监控数据。此存储数据将为您提供与当前性能数据进行比较的基准，确定正常性能模式和性能异常，以及设计解决问题的方法。

例如，使用 Amazon EFS，您可以监控网络吞吐量、I/O 读取、写入和元数据操作、客户端连接以及文件系统的突发信用余额。如果性能低于您设定的基准，可能需要更改文件系统的大小或连接的客户端数量，以便针对您的工作负载优化文件系统。

要建立基准，您至少应监控以下各项：

- 文件系统的网络吞吐量。
- 文件系统的客户端连接数量。
- 每个文件系统操作的字节数，包括数据读取、数据写入和元数据操作。

主题

- [监控工具](#)
- [Amazon EFS 如何报告文件系统和对象大小](#)
- [查看存储类大小](#)
- [使用 Amazon 监控指标 CloudWatch](#)
- [使用记录 Amazon EFS API 调用 AWS CloudTrail](#)

监控工具

AWS 提供了可用于监控 Amazon EFS 的各种工具。可以配置其中的一些工具来为您执行监控任务，但有些工具需要手动干预。建议您尽可能实现监控任务自动化。

自动监控工具

可使用以下自动化监控工具来监控 Amazon EFS，并在出现错误时进行报告：

- A CloudWatch Amazon Alarms — 在您指定的时间段内观察单个指标，并根据该指标在多个时间段内相对于给定阈值的值执行一项或多项操作。该操作是发送到亚马逊简单通知服务 (Amazon SNS) Simple Notification Scaling 主题或亚马逊 EC2 Auto Scaling 策略的通知。CloudWatch 警报不会仅仅因为它们处于特定状态而调用操作；该状态必须已更改并保持了指定的时间段。有关更多信息，请参阅 [使用 Amazon 监控指标 CloudWatch](#)。
- Amazon CloudWatch Logs — 监控、存储和访问来自 AWS CloudTrail 或其他来源的日志文件。有关更多信息，请参阅[什么是 Amazon CloudWatch 日志？](#) 在《亚马逊 CloudWatch 用户指南》中。
- Amazon CloudWatch Events — 匹配事件并将其路由到一个或多个目标函数或流，以进行更改、捕获状态信息并采取纠正措施。有关更多信息，请参阅《[亚马逊 CloudWatch 用户指南](#)》中的[什么是亚马逊 CloudWatch 活动](#)。
- AWS CloudTrail 日志监控-在账户之间共享日志文件，通过将 CloudTrail 日志文件发送到“日志”来实时监控 CloudWatch 日志文件，用 Java 编写日志处理应用程序，并验证您的日志文件在传送后是否未更改 CloudTrail。有关更多信息，请参阅《AWS CloudTrail 用户指南》中的[使用 CloudTrail 日志文件](#)。

手动监控工具

监控 Amazon EFS 的另一个重要部分涉及手动监控 Amazon CloudWatch 警报未涵盖的项目。Amazon EFS 和其他 AWS 管理控制台 控制面板提供您的 AWS 环境状态 at-a-glance 视图。CloudWatch 建议您还要查看文件系统的日志文件。

- 可以从 Amazon EFS 控制台找到文件系统的以下项目：
 - 当前计量大小
 - 挂载目标的数量
 - 生命周期状态
- CloudWatch 主页显示：

- 当前告警和状态
- 告警和资源图表
- 服务运行状况

此外，您还可以使用 CloudWatch 执行以下操作：

- 创建[自定义控制面板](#)以监控您使用的服务。
- 绘制指标数据图，以排除问题并弄清楚趋势。
- 搜索并浏览您的所有 AWS 资源指标。
- 创建和编辑告警接收有关问题的通知。

Amazon EFS 如何报告文件系统和对象大小

下面的章节介绍 Amazon EFS 如何报告文件系统大小、文件系统内对象的大小以及文件系统吞吐量。

计量 EFS 文件系统对象

您可以在 EFS 文件系统中查看的对象包括常规文件、目录、符号链接和特殊文件（FIFOs 和套接字）。其中的每个对象按照 2 千位二进制字节 (KiB) 元数据（对于其 inode）以及一个或多个 4 KiB 数据增量进行计量。以下列表说明了不同类型的文件系统对象的计量数据大小：

- 常规文件 – 常规文件的计量数据大小是舍入到下一个 4 KiB 增量的文件逻辑大小，但稀疏文件可能较小。

稀疏文件 具有这样一种特点：在达到其逻辑大小之前，不会将数据写入文件的全部位置。对于稀疏文件，在某些情况下，使用的实际存储小于舍入到下一个 4 KiB 增量的逻辑大小。在这些情况下，Amazon EFS 将使用的实际存储报告为计量的数据大小。

- 目录 – 目录的计量数据大小是用于目录条目和保存这些条目的数据结构的实际存储，舍入到下一个 4 KiB 增量。计量的数据大小不包含文件数据使用的实际存储。
- 符号链接和特殊文件 – 这些对象的计量数据大小始终为 4 KiB。

当 Amazon EFS 通过 NFSv4.1 `space_used` 属性报告对象使用的空间时，它包括该对象的当前计量数据大小，但不包括其元数据大小。您可以使用以下两个实用程序测量文件的磁盘使用情况：`du` 和 `stat` 实用程序。下例说明了如何对空文件使用 `du` 实用程序，利用 `-k` 选项以返回以千字节为单位的输出。

```
$ du -k file
```

```
4      file
```

下例说明了如何对空文件使用 stat 实用程序来返回文件的磁盘使用情况。

```
$ /usr/bin/stat --format="%b*%B" file | bc
4096
```

要测量目录的大小，请使用 stat 实用程序。找到 Blocks 值，然后将该值乘以块大小。下面是如何对空目录使用 stat 实用程序的示例：

```
$ /usr/bin/stat --format="%b*%B" . | bc
4096
```

EFS 文件系统的计量大小

EFS 文件系统的计量大小包括所有 EFS 存储类别中所有当前对象的大小之和。每个对象的大小根据计量小时（例如上午 8:00 到上午 9:00）内的对象大小的代表性取样计算得出。

空文件对文件系统计量大小贡献 6 KiB（2 KiB 元数据 + 4 KiB 数据）。在创建时，文件系统有一个空的根目录，因此计量大小为 6 KiB。

特定文件系统的计量大小定义这一小时内针对该文件系统对所有者账户计费的使用量。

Note

计算的计量大小不表示文件系统在该小时内的任何特定时间的一致快照。相反，它表示每小时内的不同时间（也可能是前一小时）在文件系统中存在的对象的大小。这些大小的总和确定该小时的文件系统计量大小。因此，文件系统的计量大小最终与没有在文件系统中写入内容时存储的对象的计量大小一致。

可通过以下方式查看 EFS 文件系统的计量大小：

- 使用 [describe-file-systems](#) AWS CLI 命令和 [DescribeFileSystem](#) API 操作，响应包括以下内容：

```
"SizeInBytes":{
  "Timestamp": 1403301078,
  "Value": 29313744866,
  "ValueInIA": 675432,
```

```
"ValueInStandard": 29312741784
"ValueInArchive": 327650
}
```

其中，计量的大小还 `ValueInStandard` 用于确定 I/O 吞吐量基准和使用突增吞吐量的文件系统的突发速率。

- 查看 `StorageBytes CloudWatch` 指标，该指标显示每个存储类中计量的数据总大小。有关 `StorageBytes` 指标的更多信息，请参阅 [CloudWatch 亚马逊 EFS 的指标](#)。
- 在 Linux 中，可在 EC2 实例的终端提示符下运行 `df` 命令。

不要在文件系统的根目录上使用 `du` 命令进行存储计量，因为响应不会反映用于计量文件系统的完整数据。

Note

`ValueInStandard` 的计量大小还用于确定您的 I/O 吞吐量基准值和突发速率。有关更多信息，请参阅 [突增吞吐量](#)。

计量不频繁访问和存档存储类

EFS 不频繁访问（IA）和归档存储类以 4KiB 为增量进行计量，且对于每个文件按最低 128KiB 收取账单费用。IA 和归档文件元数据（每个文件 2KiB）始终存储在标准存储类中并在标准存储类中计量。对小于 128KiB 的文件的支持仅适用于太平洋时间 2023 年 11 月 26 日中午 12:00 或之后更新的生命周期策略。IA 和归档存储的数据访问以 128KiB 为增量进行计量。

您可以使用该 `StorageBytes CloudWatch` 指标来查看每个存储类中计量的数据大小。该指标还显示了在 IA 和存档存储类中，为了向上舍入小文件而消耗的总字节数。有关查看 `CloudWatch` 指标的更多信息，请参阅 [访问 Amazon EFS 的 CloudWatch 指标](#)。有关 `StorageBytes` 指标的更多信息，请参阅 [CloudWatch 亚马逊 EFS 的指标](#)。

计量吞吐量

Amazon EFS 以其他文件系统 I/O 操作速率的三分之一来计量读取请求的吞吐量。例如，如果您每秒驱动读取和写入吞吐量 30 MB (MiBps)，则读取部分计为有效吞吐量的 10 MiBps，写入部分计为 30 MiBps，计量总吞吐量为 40。MiBps 根据消耗率调整后的总吞吐量反映在 `MeteredIOBytes CloudWatch` 指标中。

弹性吞吐量计量

为文件系统启用弹性吞吐量模式后，只需为在文件系统中读取或写入的元数据和数据量付费。使用弹性吞吐量模式的 EFS 文件系统将元数据读取计费为读取操作，将元数据写入计费为写入操作。元数据操作在前 4 KiB 之后以 1 KiB 为增量进行计量。数据操作在前 32 KiB 之后以 1 KiB 为增量进行计量。

Note

虽然 Elastic 吞吐量旨在根据您的吞吐量弹性扩展，但我们建议您通过使用 CloudWatch（计量 IOBytes）监控指标和使用情况警报来实施适当的治理，以此作为最佳运营实践的一部分。这有助于保持资源的最优利用率，并确保始终保持在计划的操作参数范围内。有关更多信息，请参阅 [使用 Amazon 监控指标 CloudWatch](#)。

预调配吞吐量计量

对于使用预置吞吐量模式的文件系统，只需为启用吞吐量的时间量付费。Amazon EFS 对启用预置吞吐量模式的文件系统每小时进行一次计量。对于预置吞吐量模式设置为少于一小时的计量，Amazon EFS 使用毫秒精度计算平均时间。

查看存储类大小

您可以使用 Amazon EFS 控制台、或 EFS API 查看文件系统的每个存储类别中 AWS CLI 存储了多少数据。

使用控制台

文件系统详细信息页面上的计量大小选项卡以字节的二进制倍数（千字节、兆字节、千兆字节和太字节）显示文件系统的当前计量大小。该指标每 15 分钟发布一次，允许您查看文件系统在一段时间内的计量大小。计量大小显示文件系统存储大小的以下信息：

- 总大小是存储在文件系统的数据的大小（以二进制字节为单位），包括所有存储类。
- 标准版的大小是存储在 EFS 标准存储类中的数据的大小（以二进制字节为单位）。
- IA 版的大小是存储在 EFS 不频繁访问存储类中的数据的大小（以二进制字节为单位）。小于 128KiB 的文件向上舍入到 128KiB。
- 归档版的大小是存储在 EFS 存档存储类中的数据的大小（以二进制字节为单位）。小于 128KiB 的文件向上舍入到 128KiB。

也可以在 Amazon EFS 控制台的文件系统详细信息页面上的监控选项卡上查看 Storage bytes 指标。有关更多信息，请参阅 [访问 Amazon EFS 的 CloudWatch 指标](#)。

使用 AWS CLI

您可以使用 AWS CLI 或 EFS API 查看文件系统的每个存储类别中存储了多少数据。通过调用 describe-file-systems CLI 命令查看数据存储详细信息（相应的 API 操作是 [DescribeFileSystems](#)）。

```
$ aws efs describe-file-systems \
  --region us-west-2 \
  --profile adminuser
```

在响应中，ValueInIA 显示文件系统的不频繁访问存储类中上次计量的大小（以字节为单位）。ValueInStandard 显示标准存储类中上次计量的大小（以字节为单位）。ValueInArchive 显示存档存储类中上次计量的大小（以字节为单位）。这三个值的总和等于整个文件系统的大小，如 Value 中所示。

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

有关查看和测量磁盘使用情况的其他方法，请参阅[计量 EFS 文件系统对象](#)。

使用 Amazon 监控指标 CloudWatch

您可以使用 Amazon 监控文件系统 CloudWatch，Amazon 会收集来自 Amazon EFS 的原始数据并将其处理为可读的近乎实时的指标。这些统计数据会保存 15 个月，这样您就可以更好地了解 Web 应用程序或服务的运行情况。

默认情况下，Amazon EFS 指标数据以 1 分钟为间隔自动发送到 CloudWatch，除非某些单独指标另有说明。Amazon EFS 控制台根据来自亚马逊的原始数据显示一系列图表 CloudWatch。根据您的需求，您可能更喜欢从控制台中获取文件系统的数据，CloudWatch 而不是从控制台中的图表中获取数据。

有关亚马逊的更多信息 CloudWatch，请参阅[什么是亚马逊 CloudWatch？](#) 在《亚马逊 CloudWatch 用户指南》中。

Amazon EFS CloudWatch 指标以原始字节形式报告。字节数不会舍入到十进制或二进制单位倍数。

主题

- [CloudWatch 亚马逊 EFS 的指标](#)
- [访问 Amazon EFS 的 CloudWatch 指标](#)
- [使用 Amazon EFS 的 CloudWatch 指标](#)
- [将公制数学与 CloudWatch 指标配合使用](#)
- [监控挂载尝试的成功与失败](#)
- [创建 CloudWatch 警报以监控 Amazon EFS](#)

CloudWatch 亚马逊 EFS 的指标

Amazon EFS 指标使用 EFS 命名空间。AWS/EFS 命名空间包括以下指标。除 TimeSinceLastSync 外，所有指标均针对单一维度 FileSystemId。可以在 Amazon EFS 管理控制台中找到文件系统 ID，该 ID 采用 fs-abcdef0123456789a 的格式。

TimeSinceLastSync

显示自复制配置中上次成功同步到目标文件系统以来经过的时间。成功复制 TimeSinceLastSync 值之前对源文件系统上的数据所做的任何更改。在 TimeSinceLastSync 之后发生的对源文件系统上数据所做的任何更改都可能无法完全复制。

该指标使用两个维度：

- `FileSystemId` 维度 - 复制配置中目标文件系统的 ID。
- `DestinationFileSystemId` 维度 - 复制配置中目标文件系统的 ID。

单位：秒

有效统计数据：Minimum、Maximum、Average

PercentIOLimit

显示文件系统离达到通用性能模式 I/O 极限的距离有多近。

单位：百分比

有效统计数据：Minimum、Maximum、Average

BurstCreditBalance

文件系统具有的突增额度。利用突增额度，文件系统可以突增到高于不同时段内文件系统基线水平的吞吐量级别。

Minimum 统计数据是该时段内任何一分钟的最小突增点数余额。Maximum 统计数据是该时段内任何一分钟的最大突增点数余额。Average 统计数据是该时段内的平均突增点数余额。

单位：字节

有效统计数据：Minimum、Maximum、Average

PermittedThroughput

文件系统可以驱动的最大吞吐量。

- 对于使用弹性吞吐量的文件系统，此值反映了文件系统的最大写入吞吐量。
- 对于使用预置吞吐量的文件系统，如果 EFS 标准存储类中存储的数据量允许您的文件系统驱动比您预置的吞吐量更高的吞吐量，则此指标反映的是更高的吞吐量，而不是预置的吞吐量。
- 对于采用突增吞吐量的文件系统，此值是文件系统大小与 `BurstCreditBalance` 的函数。

Minimum 统计数据是该时段内任何一分钟允许的最小吞吐量。Maximum 统计数据是该时段内任何一分钟允许的最大吞吐量。Average 统计数据是该时段内允许的平均吞吐量。

Note

读取操作以其他操作三分之一的速率计量。

单位：字节/秒

有效统计数据：Minimum、Maximum、Average

MeteredIOBytes

每个文件系统操作（包括数据读取、数据写入和元数据操作）的计量字节数，读取操作的折扣基于吞吐量限制。

您可以创建与进行比较MeteredIOBytes的[CloudWatch 指标数学表达式](#) PermittedThroughput。如果这些值相等，则说明您正在使用分配给文件系统的全部吞吐量。在这种情况下，可以考虑更改文件系统的吞吐量模式以获得更高的吞吐量。

Sum 统计数据是与所有文件系统操作关联的总计量字节数。Minimum 统计数据是该时段内的最小操作的大小。Maximum 统计数据是该时段内的最大操作的大小。Average 统计数据是该时段内的操作的平均大小。SampleCount 统计数据提供了所有操作数。

单位：

- Minimum、Maximum、Average 和 Sum 统计数据的单位是字节。
- SampleCount 的数量。

有效统计数据：Minimum、Maximum、Average、Sum、SampleCount

总计 IOBytes

Amazon EFS 处理的每个文件系统操作的实际字节数，没有任何读取折扣。该数字可能与您的应用程序请求的实际数量不同，因为它包含了最低数量。此数字也可能高于 PermittedThroughput 中显示的数字。

数据操作以 32 KiB 计量，其他操作以 4 KiB 计量。在超过最小值之后，所有操作均按每 KiB 计量。

Sum 统计数据是与所有文件系统操作关联的总字节数。Minimum 统计数据是该时段内的最小操作的大小。Maximum 统计数据是该时段内的最大操作的大小。Average 统计数据是该时段内的操作的平均大小。SampleCount 统计数据提供了所有操作数。

Note

要计算某个时段内的每秒平均操作数，请将 SampleCount 统计数据除以该时段的秒数。
要计算某个时段内的平均吞吐量（每秒字节数），请将 Sum 统计数据除以该时段的秒数。

单位：

- Minimum、Maximum、Average 和 Sum 统计数据单位是字节。
- SampleCount 的数量。

有效统计数据：Minimum、Maximum、Average、Sum、SampleCount

DataReadIOBytes

每个文件系统读取操作的实际字节数。

Sum 统计数据是与读取操作关联的总字节数。Minimum 统计数据是该时段内的最小读取操作的大小。Maximum 统计数据是该时段内的最大读取操作的大小。Average 统计数据是该时段内的读取操作的平均大小。SampleCount 统计数据提供了读取操作数。

单位：

- 对于 Minimum、Maximum、Average 和 Sum，单位为字节。
- SampleCount 的数量。

有效统计数据：Minimum、Maximum、Average、Sum、SampleCount

DataWriteIOBytes

每个文件系统写入操作的实际字节数。

Sum 统计数据是与写入操作关联的总字节数。Minimum 统计数据是该时段内的最小写入操作的大小。Maximum 统计数据是该时段内的最大写入操作的大小。Average 统计数据是该时段内的写入操作的平均大小。SampleCount 统计数据提供了写入操作数。

单位：

- Minimum、Maximum、Average 和 Sum 统计数据单位是字节。
- SampleCount 的数量。

有效统计数据：Minimum、Maximum、Average、Sum、SampleCount

MetadataIOBytes

每个元数据操作的实际字节数。

Sum 统计数据是与元数据操作关联的总字节数。Minimum 统计数据是该时段内的最小元数据操作的大小。Maximum 统计数据是该时段内的最大元数据操作的大小。Average 统计数据是该时段内的平均元数据操作的大小。SampleCount 统计数据提供了元数据操作数。

单位：

- Minimum、Maximum、Average 和 Sum 统计数据的单位是字节。
- SampleCount 的数量。

有效统计数据：Minimum、Maximum、Average、Sum、SampleCount

MetadataReadIOBytes

每个元数据读取操作的实际字节数。

Sum 统计数据是与元数据读取操作关联的总字节数。Minimum 统计数据是该时段内的最小元数据读取操作的大小。Maximum 统计数据是该时段内的最大元数据读取操作的大小。Average 统计数据是该时段内的元数据读取操作的平均大小。SampleCount 统计数据提供了元数据读取操作数。

单位：

- Minimum、Maximum、Average 和 Sum 统计数据的单位是字节。
- SampleCount 的数量。

有效统计数据：Minimum、Maximum、Average、Sum、SampleCount

MetadataWriteIOBytes

每个元数据写入操作的实际字节数。

Sum 统计数据是与元数据写入操作关联的总字节数。Minimum 统计数据是该时段内的最小元数据写入操作的大小。Maximum 统计数据是该时段内的最大元数据写入操作的大小。Average 统计数据是该时段内的元数据写入操作的平均大小。SampleCount 统计数据提供了元数据写入操作数。

单位：

- Minimum、Maximum、Average 和 Sum 统计数据的单位是字节。
- SampleCount 的数量。

有效统计数据：Minimum、Maximum、Average、Sum、SampleCount

ClientConnections

文件系统的客户端连接数量。使用标准客户端时，每个挂载的 Amazon EC2 实例使用一个连接。

Note

要计算超过一分钟的时段的 ClientConnections 平均值，请将 Sum 统计数据除以该时段的分钟数。

单位：客户端连接数量

有效统计数据：Sum

StorageBytes

文件系统的大小（以字节为单位），包括存储在 EFS 存储类中的数据量。该指标 CloudWatch 每 15 分钟发出一次。

StorageBytes 指标具有以下维度：

- Total 是存储在文件系统的数据（所有存储类）的计量大小（以字节为单位）。对于 EFS 不频繁访问（IA）和 EFS 存档存储类，小于 128KiB 的文件向上舍入到 128KiB。
- Standard 是存储在 EFS 标准存储类别中的数据的计量大小（以字节为单位）。
- IA 是存储在 EFS 不频繁访问存储类别中的数据的实际大小（以字节为单位）。
- IASizeOverhead 是 EFS 不频繁访问存储类中的实际数据大小（在 IA 维度中指示）与将小文件向上舍入到 128KiB 后的该存储类计量大小之间的差异（以字节为单位）。
- Archive 是存储在 EFS 存档存储类中的数据的实际大小（以字节为单位）。
- ArchiveSizeOverhead 是 EFS 存档存储类中的实际数据大小（在 Archive 维度中指示）与将小文件向上舍入到 128KiB 后的该存储类计量大小之间的差异（以字节为单位）。

单位：字节

有效统计数据：Minimum、Maximum、Average

Note

StorageBytes 显示在 Amazon EFS 控制台的文件系统指标页面上，使用 1024 个基本单位（千字节、兆字节、千兆字节和太字节）。

访问 Amazon EFS 的 CloudWatch 指标

您可以通过多种方式查看 Amazon EFS CloudWatch 的指标：

- 在 Amazon EFS 控制台中
- 在 CloudWatch 控制台中
- 使用 CloudWatch CLI

- 使用 CloudWatch API

查看 CloudWatch 指标和警报 (Amazon EFS 控制台)

1. 登录 AWS 管理控制台 并打开 Amazon EFS 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 选择文件系统。
3. 选择要查看其 CloudWatch 指标的文件系统。
4. 选择监控以显示文件系统指标页面。

“文件系统指标”页面显示文件系统的一组默认 CloudWatch 指标。您配置的所有 CloudWatch 警报也会显示这些指标。对于使用“最大 I/O 性能”模式的文件系统，默认的指标集包括突发积分余额，而不是 IO 百分比限制。可以使用指标设置对话框覆盖默认设置，通过打开设置即可访问该对话框。

Note

吞吐量利用率 (%) 指标不是 CloudWatch 指标；它是使用 CloudWatch 公制数学计算得出的。

5. 可以使用文件系统指标页面上的控制调整指标和警报的显示方式，如下所示。
 - 在时间序列或单个值之间切换显示模式。
 - 显示或隐藏为文件系统配置的所有 CloudWatch 警报。
 - 选择“查看更多” CloudWatch 以查看中的指标 CloudWatch。
 - 选择“添加到控制面板”以打开您的 CloudWatch 控制面板并添加显示的指标。
 - 将显示的指标时间窗口从 1 小时调整为 1 周。

查看 CloudWatch 指标和警报 (CloudWatch 控制台)

1. 打开 CloudWatch 控制台，网址为 <https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择指标。
3. 选择 EFS 命名空间。
4. (可选) 要查看某个指标，请在搜索字段中输入其名称。
5. (可选) 要按维度筛选，请选择 FileSystemId。

要访问来自的指标 AWS CLI

- 使用带有 `--namespace "AWS/EFS"` 命名空间的 [list-metrics](#) 命令。有关更多信息，请参阅 [AWS CLI 命令参考](#)。

从 CloudWatch API 访问指标

- 调用 [GetMetricStatistics](#)。有关更多信息，请参阅 [Amazon CloudWatch API 参考](#)。

使用 Amazon EFS 的 CloudWatch 指标

Amazon EFS 报告的指标为您提供可通过不同方式分析的信息。下面的列表显示这些指标的一些常见用途。这些是入门建议，并不全面。

如何？	相关指标
如何确定我的吞吐量？	您可以监控 Sum 指标的每日 TotalIOBytes 统计数据以查看您的吞吐量。
如何确定传输的数据总量？	您可以监控 MeteredIOBytes 的每日 Sum 统计数据，以查看传输的数据总量。
如何跟踪连接到文件系统的 Amazon EC2 实例数量？	您可以监控 Sum 指标的 ClientConnections 统计数据。要计算超过一分钟的时段的 ClientConnections 平均值，请将总和除以该时段的分钟数。
如何查看我的突增积分余额？	您可以通过监控文件系统的 BurstCreditBalance 指标来查看您的余额。有关突增和突增积分的更多信息，请参阅 突增吞吐量 。

监控吞吐量性能

吞吐量监控 CloudWatch 指标 — TotalIOBytes ReadIOBytes、WriteIOBytes、和 MetadataIOBytes — 表示您在文件系统上实现的实际吞吐量。指标 MeteredIOBytes 表示您正在驱动的总计量吞吐量的计算结果。可以使用 Amazon EFS 控制台监控部分中的吞吐量利用率 (%) 图表来监控吞吐量利用率。如果您使用自定义 CloudWatch 仪表板或其他监控工具，则可以创建与进行比较 MeteredIOBytes 的 [CloudWatch 指标数学表达式](#) PermittedThroughput。

PermittedThroughput 衡量文件系统允许的吞吐量。此值基于以下方法之一：

- 对于采用弹性吞吐量的文件系统，此值反映了文件系统的最大写入吞吐量。
- 对于使用预置吞吐量的文件系统，如果 EFS 标准存储类中存储的数据量允许您的文件系统驱动比您预置的吞吐量更高的吞吐量，则此指标反映的是更高的吞吐量，而不是预置的吞吐量。
- 对于采用突增吞吐量的文件系统，此值是文件系统大小与 BurstCreditBalance 的函数。监控 BurstCreditBalance 以确保您的文件系统以其突发速率而不是基本速率运行。如果余额一直等于或接近零，请考虑切换到弹性吞吐量或预置吞吐量，以获得额外吞吐量。

当 MeteredIOBytes 和 PermittedThroughput 的值相等时，您的文件系统将消耗所有可用吞吐量。对于使用预置吞吐量的文件系统，可以预置额外的吞吐量。

将公制数学与 CloudWatch 指标配合使用

使用指标数学，您可以查询多个 Amazon CloudWatch 指标，并使用数学表达式根据这些指标创建新的时间序列。您可以在 CloudWatch 控制台中可视化生成的时间序列并将其添加到仪表板中。例如，可以使用 Amazon EFS 指标将 DataRead 操作样本数除以 60。结果是在给定 1 分钟间隔内在文件系统上平均每秒读取的次数。有关指标数学的更多信息，请参阅 Amazon CloudWatch 用户指南中的[将数学表达式与 CloudWatch 指标配合使用](#)。

可以在下文中找到一些有用的 Amazon EFS 指标数学表达式。

主题

- [公制数学：吞吐量 MiBps](#)
- [指标数学：百分比吞吐量](#)
- [指标数学：允许的吞吐量利用率百分比](#)
- [指标数学：吞吐量 IOPS](#)
- [指标数学：IOPS 百分比](#)
- [公制数学：以 KiB 为 I/O 单位的平均大小](#)
- [通过 Amazon EFS 的 CloudFormation 模板使用公制数学](#)

公制数学：吞吐量 MiBps

要计算一段时间内的平均吞吐量 (in MiBps)，请先选择一个总和统计数据 (DataReadIOBytes、DataWriteIOBytes、MetadataIOBytes、或TotalIOBytes)。然后，将该值转换为 MiB，并将该值除以该时间段的秒数。

假设您的示例逻辑是： $(\text{TotalIOBytes 总和} \div 1048576 \text{ (以转换为 MiB)}) \div \text{该时间段的秒数}$

那么您的 CloudWatch 指标信息如下所示。

ID	可用的指标	Statistic	周期
m1	- DataReadIOBytes - DataWriteIOBytes - MetadataIOBytes - TotalIOBytes	sum	1 minute

您的指标数学 ID 和表达式如下所示。

ID	Expression
e1	$(m1/1048576)/\text{PERIOD}(m1)$

指标数学：百分比吞吐量

此度量数学表达式计算用于不同 I/O 类型的总吞吐量的百分比，例如，由读取请求驱动的总吞吐量的百分比。要计算一段时间内某个 I/O 类型 (DataReadIOBytes、DataWriteIOBytes 或 MetadataIOBytes) 使用的总吞吐量百分比，请首先将相应的总和统计值乘以 100。然后，将结果除以同一时间段的 TotalIOBytes 总计统计数据。

假设您的示例逻辑是： $(\text{DataReadIOBytes 总和} \times 100 \text{ (以转换为百分比)}) \div \text{TotalIOBytes 总和}$

那么您的 CloudWatch 指标信息如下所示。

ID	可用的一个或多个指标	Statistic	周期
m1	TotalIOBytes	sum	1 minute

ID	可用的一个或多个指标	Statistic	周期
m2	• DataReadIOBytes	sum	1 minute

您的指标数学 ID 和表达式如下所示。

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

指标数学：允许的吞吐量利用率百分比

要计算一段时间内允许的吞吐量利用率 (MeteredIOBytes) 的百分比，请先将吞吐量乘以 100。MiBps 然后，将结果除以同一时段的 PermittedThroughput 平均统计数据（转换为 MiB）。

假设你的示例逻辑是这样的：（以 MiBps x 100 为单位的吞吐量的度量数学表达式（转换为百分比））▼（总和 ± 1,048,576（将字节转换为 MiB））PermittedThroughput

那么您的 CloudWatch 指标信息如下所示。

ID	可用的一个或多个指标	Statistic	周期
m1	MeteredIOBytes	sum	1 minute
m2	Permitted Throughput	average	1 minute

您的指标数学 ID 和表达式如下所示。

ID	Expression
e1	$(m1 / 1048576) / \text{PERIOD}(m1)$

ID	Expression
e2	m2/1048576
e3	((e1)*100)/(e2)

指标数学：吞吐量 IOPS

要计算某个时间段的平均每秒操作数 (IOPS)，请将样本数统计数据 (DataReadIOBytes、DataWriteIOBytes、MetadataIOBytes 或 TotalIOBytes) 除以该时间段的秒数。

假设您的示例逻辑是：DataWriteIOBytes 样本数 ÷ 该时间段的秒数

那么您的 CloudWatch 指标信息如下所示。

ID	可用的指标	Statistic	周期
m1	- DataReadIOBytes - DataWriteIOBytes - MetadataIOBytes - TotalIOBytes	样本数	1 minute

您的指标数学 ID 和表达式如下所示。

ID	Expression
e1	m1/PERIOD(m1)

指标数学：IOPS 百分比

要计算一段时间内不同 I/O 类型 (DataReadIOBytes、DataWriteIOBytes 或 MetadataIOBytes) 每秒 IOPS 的百分比，请先将

相应的样本计数统计数据乘以 100。然后，将该值除以同一时间段的 TotalIOBytes 样本数统计数据。

假设您的示例逻辑是： $(\text{MetadataIOBytes 样本数} \times 100 \text{ (以转换为百分比)}) \div \text{TotalIOBytes 样本数}$

那么您的 CloudWatch 指标信息如下所示。

ID	可用的指标	Statistic	周期
m1	• TotalIOBytes	样本数	1 minute
m2	• DataReadIOBytes • DataWriteIOBytes • MetadataIOBytes	样本数	1 minute

您的指标数学 ID 和表达式如下所示。

ID	Expression
e1	$(m2 \times 100) / m1$

公制数学：以 KiB 为 I/O 单位的平均大小

要计算一段时间内的平均 I/O 大小（以 KiB 为单位），请将 DataReadIOBytes、DataWriteIOBytes、或 MetadataIOBytes 指标的相应总和统计数据除以该指标的相同样本计数统计量。

假设您的示例逻辑是： $(\text{DataReadIOBytes 总和} \div 1024 \text{ (以转换为 KiB)}) \div \text{DataReadIOBytes 样本数}$

那么您的 CloudWatch 指标信息如下所示。

ID	可用的指标	Statistic	周期
m1	- DataReadIOBytes - DataWriteIOBytes - MetadataIOBytes	sum	1 minute
m2	- DataReadIOBytes - DataWriteIOBytes - MetadataIOBytes	样本数	1 minute

您的指标数学 ID 和表达式如下所示。

ID	Expression
e1	$(m1/1024)/m2$

通过 Amazon EFS 的 CloudFormation 模板使用公制数学

您也可以通过 CloudFormation 模板创建公制数学表达式。其中一个模板可供您下载和自定义，以便在 [Amazon EFS 教程](#) 中使用 GitHub。有关使用 CloudFormation 模板的更多信息，请参阅《AWS CloudFormation 用户指南》中的 [使用 CloudFormation 模板](#)。

监控挂载尝试的成功与失败

您可以使用 Amazon CloudWatch Logs 远程监控和报告 EFS 文件系统的挂载尝试成功或失败，而无需登录客户端。使用以下过程将您的 EC2 实例配置为使用 CloudWatch 日志来监控其文件系统挂载尝试的成功或失败。

在 CloudWatch 日志中启用装载尝试成功或失败通知

1. 在挂载文件系统的 EC2 实例上安装 amazon-efs-utils。有关更多信息，请参阅 [使用自动安装或更新 Amazon EFS 客户端 AWS Systems Manager](#) 或 [手动安装 Amazon EFS 客户端](#)。
2. 在将挂载文件系统的 EC2 实例上安装 boto3。有关更多信息，请参阅 [安装和升级 boto3](#)。
3. 在中启用“CloudWatch 日志”功能amazon-efs-utils。当您使用 AWS Systems Manager 安装和配置时amazon-efs-utils，系统会自动为您完成 CloudWatch 日志记录。手动安装 amazon-efs-utils 软件包时，必须通过取消对 cloudwatch-log 部分中 # enabled = true 行的注释来手动更新 /etc/amazon/efs/efs-utils.conf 配置文件。使用以下命令之一手动启用 CloudWatch 日志。

对于 Linux 实例：

```
sudo sed -i -e '/\[cloudwatch-log\]/{N;s/# enabled = true/enabled = true/}' /etc/amazon/efs/efs-utils.conf
```

对于 MacOS 实例：

```
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
```

对于 Mac2 实例：

```
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. 或者，您可以配置 CloudWatch 日志组名称并在efs-utils.conf文件中设置日志保留天数。如果要 CloudWatch 为每个已装载的文件系统设置单独的日志组，请在efs-utils.conf文件中的log_group_name字段末尾添加/{fs_id}，如下所示：

```
[cloudwatch-log]
log_group_name = /aws/efs/utils/{fs_id}
```

5. 将AmazonElasticFileSystemsUtils AWS 托管策略附加到您附加到 EC2 实例的 IAM 角色或实例上配置的 AWS 证书。可以使用 Systems Manager 执行此操作，有关更多信息，请参阅[步骤 1：使用所需权限配置 IAM 实例配置文件](#)。

以下是挂载尝试状态日志条目的示例：

```
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"
```

在 CloudWatch 日志中查看装载状态

1. 打开 CloudWatch 控制台，网址为<https://console.aws.amazon.com/cloudwatch/>。
2. 在左侧导航栏中，选择日志组。
3. 选择/aws/efs/utils日志组。您将看到每个 Amazon EC2 实例和 EFS 文件系统组合的日志流。
4. 选择日志流以查看特定日志事件，包括挂载尝试成功或失败状态。

创建 CloudWatch 警报以监控 Amazon EFS

您可以创建一个 CloudWatch 警报，在警报状态发生变化时发送 Amazon SNS 消息。告警会监控您指定的时间段内的某个指标。然后警报会根据指标值在多个时间段内对比给定阈值的情况执行一项或多项操作。操作是一个发送到 Amazon SNS 主题或自动扩缩策略的通知。

警报仅针对持续的状态变化调用操作。CloudWatch 警报不会仅仅因为它们处于特定状态而调用操作；该状态必须已更改并保持了指定的时间段。

Amazon EFS CloudWatch 警报的一个重要用途是对文件系统强制执行静态加密。可以在创建 Amazon EFS 文件系统时启用静态加密。要对 Amazon EFS 文件系统强制执行数据 encryption-at-rest策略，您可以使用 Amazon CloudWatch 和 AWS CloudTrail 来检测文件系统的创建并验证是否启用了静态加密。

Note

目前，您无法实施传输中加密。

以下过程简要说明了如何为 Amazon EFS 创建警报。

使用控制台

使用 CloudWatch 控制台设置警报

1. 登录 AWS 管理控制台 并打开 CloudWatch 控制台，网址为<https://console.aws.amazon.com/cloudwatch/>。
2. 选择创建警报。创建警报向导随即启动。
3. 选择 EFS 指标，并滚动 Amazon EFS 指标以找到要为其设置警报的指标。要在此对话框中仅显示 Amazon EFS 指标，请搜索文件系统的文件系统 ID。选择要创建警报的指标，然后选择下一步。
4. 填写指标的 Name、Description、Whenever 值。
5. 如果 CloudWatch 要在达到警报状态时向您发送电子邮件，请在“每当此警报：”字段中，选择“状态为警报”。在 Send notification to (发送通知到) 字段中，选择一个现有 SNS 主题。如果您选择创建主题，那么您就可以为新电子邮件订阅列表设置名称和电子邮件地址。此列表将保存下来并会在将来的警报字段中显示出来。

Note

如果您使用 Create topic (创建主题) 创建一个新 Amazon SNS 主题，那么电子邮件地址在接收通知之前必须通过验证。当警报进入警报状态时，才会发送电子邮件。如果在验证电子邮件地址之前警报状态发生了变化，那么它们不会接收到通知。

6. 此时，Alarm Preview 区域将为您提供一次机会来预览即将创建的警报。选择创建警报。

使用 AWS CLI

要使用设置警报 AWS CLI

- 调用 [put-metric-alarm](#)。有关更多信息，请参阅 [AWS CLI 命令参考](#)。

使用 CloudWatch API

使用 CloudWatch API 设置警报

- 调用 [PutMetricAlarm](#)。有关更多信息，请参阅 [Amazon CloudWatch API 参考](#)。

使用记录 Amazon EFS API 调用 AWS CloudTrail

Amazon EFS 与 AWS CloudTrail 一项服务集成，该服务提供用户、角色或 AWS 服务在 Amazon EFS 中执行的操作的记录。CloudTrail 将 Amazon EFS 的所有 API 调用捕获为事件，包括来自亚马逊 EFS 控制台的调用和对亚马逊 EFS API 操作的代码调用。

如果您创建了跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 Amazon EFS 的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的事件历史记录中查看最新的事件。通过收集的信息 CloudTrail，您可以确定向 Amazon EFS 发出的请求、发出请求的 IP 地址、谁提出了请求、何时发出请求以及其他详细信息。

有关更多信息，请参阅[什么是 AWS CloudTrail？](#) 在《AWS CloudTrail 用户指南》中。

Amazon EFS 中的信息 CloudTrail

CloudTrail 在您创建账户 AWS 账户 时已在您的账户上启用。当 Amazon EFS 中发生活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在中查看、搜索和下载最近发生的事件 AWS 账户。有关更多信息，请参阅[使用 CloudTrail 事件历史记录](#)。

要持续记录您的事件 AWS 账户，包括 Amazon EFS 的事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，当您在控制台中创建跟踪时，该跟踪将应用于所有 AWS 区域跟踪。跟踪记录 AWS 分区 AWS 区域 中所有事件并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅《AWS CloudTrail 用户指南》中的以下主题：

- [为您的 AWS 账户创建跟踪](#)
- [AWS 与日志的服务集成 CloudTrail](#)
- [配置 Amazon SNS 通知 CloudTrail](#)
- [接收来自多个区域的 CloudTrail 日志文件](#)和[接收来自多个账户的 CloudTrail 日志文件](#)

所有 Amazon EFS [Amazon EFS API](#) 都由其登录 CloudTrail。例如，调用 CreateMountTarget 和 CreateTags 操作会在 CloudTrail 日志文件中生成条目。CreateFileSystem

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户还是 AWS Identity and Access Management (IAM) 用户证书发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。

- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 [《CloudTrail 用户指南》中的AWS CloudTrail 用户身份元素](#)。

了解 Amazon EFS 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序出现。

以下示例显示了一个 CloudTrail 日志条目，该条目演示了从控制台为文件系统创建标签时的CreateTags操作。

```
{
  "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"
}

```

以下示例显示了一个 CloudTrail 日志条目，该条目演示了从控制台中删除文件系统的标签时的 DeleteTags 操作。

```

{
  "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"
}
```

EFS 服务相关角色的日志条目

Amazon EFS 服务相关角色对 AWS 资源进行 API 调用。您将看到 EFS 服务相关角色发出的呼叫的 CloudTrail 日志条目。username: AWSServiceRoleForAmazonElasticFileSystem 有关 EFS 和服务相关角色的更多信息，请参阅[对 Amazon EFS 使用服务相关角色](#)。

以下示例显示了一个 CloudTrail 日志条目，该条目演示了 Amazon EFS 创建 AWSServiceRoleForAmazonElasticFileSystem 服务相关角色时的 CreateServiceLinkedRole 操作。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/user1",
    "accountId": "111122223333",
    "accessKeyId": "A111122223333",
    "userName": "user1",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-10-23T22:45:41Z"
      }
    }
  },
  "invokedBy": "elasticfilesystem.amazonaws.com",
  "eventTime": "2019-10-23T22:45:41Z",
  "eventSource": "iam.amazonaws.com",
  "eventName": "CreateServiceLinkedRole",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "user_agent",
  "requestParameters": {
    "awsServiceName": "elasticfilesystem.amazonaws.com"
  },
  "responseElements": {
    "role": {
```

```

    "assumeRolePolicyDocument":
      "111122223333-10-111122223333Statement111122223333Action111122223333AssumeRole111122223333Effect%22%3A%20%22Allow%22%2C%20%22Principal%22%3A%20%7B%22Service%22%3A%20%5B%22elasticfilesystem.amazonaws.com%22%5D%7D%7D%5D%7D",
      "arn": "arn:aws:iam::111122223333:role/aws-service-role/elasticfilesystem.amazonaws.com/AWSServiceRoleForAmazonElasticFileSystem",
      "roleId": "111122223333",
      "createDate": "Oct 23, 2019 10:45:41 PM",
      "roleName": "AWSServiceRoleForAmazonElasticFileSystem",
      "path": "/aws-service-role/elasticfilesystem.amazonaws.com/"
    }
  },
  "requestID": "11111111-2222-3333-4444-abcdef123456",
  "eventID": "11111111-2222-3333-4444-abcdef123456",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

以下示例显示了一个 CloudTrail 日志条目，该条目演示了AWSServiceRoleForAmazonElasticFileSystem服务相关角色所执行的CreateNetworkInterface操作，如中所sessionContext述。

```

{
  "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"
}

```

EFS 身份验证的日志条目

针对 NFS 客户端发射 NewClientConnection 和 UpdateClientConnection CloudTrail 事件的 Amazon EFS 授权。当在初始连接之后立即授权连接以及在重新连接之后，会立即发出 NewClientConnection 事件。当重新授权连接并且允许的操作列表发生变化时，会发出 UpdateClientConnection。当新的允许操作列表不包含 ClientMount 时，也会发出该事件。有关 EFS 授权的更多信息，请参阅[使用 IAM 控制文件系统访问](#)。

以下示例显示了演示 NewClientConnection 事件的 CloudTrail 日志条目。

```

{
  "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"
}
}

```

文件系统的 Amazon EFS 日志 encrypted-at-rest 文件条目

Amazon EFS 允许您选择在文件系统中使用静态加密和/或传输中加密。有关更多信息，请参阅 [Amazon EFS 中的数据加密](#)。

Amazon EFS 在发出 AWS KMS API 请求以生成数据密钥和解密 [Amazon EFS 数据时会发送加密上下文](#)。文件系统 ID 是静态加密的所有文件系统的加密上下文。在 CloudTrail 日志条目的 `requestParameters` 字段中，加密上下文类似于以下内容。

```
"EncryptionContextEquals": {}  
"aws:elasticfilesystem:filesystem:id" : "fs-4EXAMPLE"
```

使用 AWS 账单和使用情况报告

使用 Amazon EFS 时，您只需为实际使用的资源付费，无最低费用或设置费用。与其他 AWS 服务一样，尽管定价按月度报价，但使用量会按小时折算，最终将根据您一个月内的平均使用量计费。

AWS 提供 Amazon EFS 的以下报告：

- 账单报告 - 多份涵盖您正在使用的 AWS 服务（包括 Amazon EFS）的所有活动详细信息的报告。AWS 始终向 EFS 文件系统所有者收取 Amazon EFS 费用。有关账单报告的更多信息，请参阅 [Amazon EFS 的 AWS Billing 报告](#)。
- 使用情况报告 - 概括了特定服务按小时、天或月聚合的活动情况。可以选择要包括的使用类型和操作。您还可以选择数据聚合的方式。有关更多信息，请参阅 [Amazon EFS 的 AWS 使用情况报告](#)。

以下主题提供有关 Amazon EFS 账单和使用情况报告的信息。

主题

- [使用成本分配 EFS 文件系统标签](#)
- [Amazon EFS 的 AWS Billing 报告](#)
- [Amazon EFS 的 AWS 使用情况报告](#)
- [了解 Amazon EFS 的账单和使用情况报告](#)

使用成本分配 EFS 文件系统标签

要跟踪单个项目或项目组的存储成本或其他标准，请使用成本分配标签标记您的 EFS 文件系统。成本分配标签是与文件系统相关联的键-值对。在激活成本分配标签后，AWS 将使用这些标签在成本分配报告上整理您的资源成本。

成本分配报告会按产品分类和关联的账户用户列出您账户的 AWS 使用量。该报告包含与详细账单报告相同的行项目（请参阅 [了解 Amazon EFS 的账单和使用情况报告](#)）和用于您的标签键的附加列。

AWS 提供了两种类型的成本分配标签，即 AWS 生成的标签 和用户定义的标签。AWS 将在 Amazon EFS CreateFileSystem 事件之后为您定义、创建和应用 AWS 生成的 `aws:createdBy` 标签。您定义、创建用户定义的标签并将其应用到您的文件系统。

必须在账单和成本管理控制台中分别激活这两种类型的标签，然后它们才能出现在账单报告中。有关 AWS 生成的标签的更多信息，请参阅《AWS Billing 用户指南》中的 [使用 AWS 成本分配标签组织和跟踪成本](#)。

- 有关为 EFS 资源创建用户定义的标签的更多信息，请参阅[标记 EFS 资源](#)。
- 有关在账单和成本管理控制台中激活用户定义的标签的更多信息，请参阅《AWS Billing 用户指南》中的[激活用户定义的成本分配标签](#)。
- 有关在账单和成本管理控制台中激活 AWS 生成的标签的更多信息，请参阅《AWS Billing 用户指南》中的[激活 AWS 生成的成本分配标签](#)。

用户定义的成本分配标签

用户定义的成本分配标签有以下组成部分：

- 标签键。标签键是标签的名称。例如，在标签 project/Trinity 中，project 是键。标记键是一个区分大小写的字符串，它可以包含 1 到 128 个 Unicode 字符。
- 标签值。标签值是必需的字符串。例如，在标签 project/Trinity 中，Trinity 是值。标记值是一个区分大小写的字符串，它可以包含 0 到 256 个 Unicode 字符。

有关用户定义的标签的更多信息，请参阅《AWS Billing 用户指南》中的[使用用户定义的成本分配标签](#)。

Amazon EFS 的 AWS Billing 报告

AWS 为您提供的月度账单按 AWS 服务和功能分隔您的使用率信息和成本。有多种 AWS 账单报告可用：月度报告、成本分配报告和详细账单报告。有关更多信息，请参阅《AWS Billing 用户指南》中的[了解您的账单](#)。

有关您的 Amazon EFS 存储使用情况的更多详细报告，请使用 AWS Data Exports 创建 AWS 成本和使用情况报告 (AWS CUR) 的导出。您可以通过 AWS CUR 2.0 将 AWS 账单报告发布到您拥有的 Amazon Simple Storage Service (Amazon S3) 存储桶。Amazon EFS 使用情况报告将按使用类型和 AWS 区域 列示操作。有关更多信息，请参阅《AWS Data Exports 用户指南》中的[成本和使用情况报告 \(CUR\) 2.0](#)。

EFS 文件系统的费用按以下使用类别来计费。

- 存储 (每 GB) - 每月存储在 EFS 文件系统中的数据量，价格因存储类别 (EFS 标准存储、EFS 不频繁访问 (IA) 存储和 EFS 存档存储) 而异。
- 吞吐量 (每 GB 或每秒兆字节 [MiBps]) - 每月传输的数据量 (读/写操作) 或超过预配置量的量，具体取决于文件系统的吞吐量。

- 数据访问和转移 (每 MiBps) - 每月读取和传输到 EFS 不频繁访问 (IA) 和 EFS 存档存储类别的数据量。
- 备份存储 (每 GB) - 备份每月消耗的存储空间量。如果延长备份保留期或增加用户启动的卷备份数量，将增加文件系统所消耗的备份存储量。有关更多信息，请参阅 [备份 EFS 文件系统](#)。
- 复制 (每 GB) - 源 EFS 文件系统和目标 EFS 文件系统的现行存储、读取、写入和分层活动费率，以及它们之间每月数据传输活动的任何适用费用。

有关 Amazon EFS 费用的详细信息，请参阅 [Amazon EFS 定价](#)。

有关了解在 Amazon EFS 的账单和使用情况报告中使用的代码和缩写的信息，请参阅 [了解 Amazon EFS 的账单和使用情况报告](#)。

Amazon EFS 的 AWS 使用情况报告

在下载使用情况报告时，您可以选择按小时、天或月聚合使用量数据。Amazon EFS 使用情况报告将按使用类型和 AWS 区域 列示操作。有关您的 Amazon EFS 存储使用量的更多详细报告，请下载动态生成的 AWS 使用情况报告。可以选择要包括的使用类型、操作和时间段。您还可以选择数据聚合的方式。有关下载使用情况报告的更多信息，请参阅《AWS 数据导出用户指南》中的 [下载 AWS 使用情况报告](#)。

Amazon EFS 使用情况报告包含以下信息：

- 服务 - Amazon EFS
- UsageType – 以下值之一：
 - 标识存储类型的代码
 - 标识请求类型的代码
 - 标识数据传输类型的代码
 - 标识吞吐量模式的代码
 - 标识备份使用情况的代码

有关 Amazon EFS 使用类型的详细说明，请参阅 [了解 Amazon EFS 的账单和使用情况报告](#)。

- Resource - 与列出的使用率相关联的资源名称。
- StartTime – 以协调世界时 (UTC) 表示的使用开始时间。
- EndTime – 以协调世界时 (UTC) 表示的使用结束时间。
- UsageValue – 以下容量值之一：数据的典型度量单位是千兆字节 (GB)。但是，根据服务和报告的不同，可能会显示万亿字节 (TB)。

- 指定时间段内的请求数
- 传输的数据量
- 给定小时内存储的数据量

有关了解在 Amazon EFS 的账单和使用情况报告中使用的代码和缩写的信息，请参阅[了解 Amazon EFS 的账单和使用情况报告](#)。

了解 Amazon EFS 的账单和使用情况报告

Amazon EFS 账单和使用情况报告使用代码和缩写。对于下表中的使用类型，请使用此列表中的缩写替换 *region*：

- APE1: 亚太地区（香港）
- APN1: 亚太地区（东京）
- APN2: 亚太地区（首尔）
- APN3: 亚太地区（大阪）
- APS1: 亚太地区（新加坡）
- APS2: 亚太地区（悉尼）
- APS3: 亚太地区（孟买）
- APS4: 亚太地区（雅加达）
- APS5: 亚太地区（海得拉巴）
- APS6: 亚太地区（墨尔本）
- CAN1: 加拿大（中部）
- CAN2: 加拿大西部（卡尔加里）
- CNN1: 中国（北京）
- CNW1: 中国（宁夏）
- AFS1: 非洲（开普敦）
- EUC2: 欧洲（苏黎世）
- EUN1: 欧洲（斯德哥尔摩）
- EUS2: 欧洲（西班牙）
- EUC1: 欧洲（法兰克福）

- EU : 欧洲 (爱尔兰)
- EUS1: 欧洲 (米兰)
- EUW2: 欧洲 (伦敦)
- EUW3: 欧洲 (巴黎)
- ILC1: 以色列 (特拉维夫)
- MEC1: 中东 (阿联酋)
- MES1: 中东 (巴林)
- SAE1: 南美洲 (圣保罗)
- UGW1: AWS GovCloud (美国西部)
- UGE1: AWS GovCloud (美国东部)
- USE1 (或无前缀) : 美国东部 (弗吉尼亚北部)
- USE2: 美国东部 (俄亥俄州)
- USW1: 美国西部 (加利福尼亚北部)
- USW2: 美国西部 (俄勒冈)

有关定价方式的信息 AWS 区域，请参阅 [Amazon EFS 定价](#)。

下表中的第一列列出了您的账单和使用情况报告中显示的使用类型。数据的典型度量单位是千兆字节 (GB)。但是，对于预配置的吞吐量，改用每秒 mbibytes (MiBps)。

Amazon EFS 的使用类型

使用情况类型	CloudWatch 指标	单位	粒度	说明
<i>region</i> -ArchiveDataAccess-Bytes	不适用	GB	不适用	在 EFS 存档存储中传输或访问的数据量。
<i>region</i> -ArchiveEarlyDelete-ByteHrs	不适用	GB-Mo	不适用	在 90 天最低承诺结束之前，从 EFS 存档存储中删除的文件的按比例存储使用量。

使用情况类型	CloudWatch 指标	单位	粒度	说明
<i>region</i> -ArchiveEarlyDelete-SmallFiles	不适用	GB-Mo	不适用	在 90 天最低承诺结束之前，从 EFS 存档存储中删除的小文件（小于 128 KB）的按比例存储使用量。
<i>region</i> -ArchiveTimedStorage-ByteHrs	StorageElasticity.ArchivedStorage	GB-Mo	每小时	数据存储在 EFS 存档存储中的 GB-月数。
<i>region</i> -ArchiveTimedStorage-SmallFiles	StorageElasticity.ArchivedStorageSizeOnDisk	GB-Mo	每小时	小对象（小于 128 KB）存储在 EFS 存档存储中的 GB-月数。
<i>region</i> -ETDataAccess-Bytes	TotalIOBytes	GB	每小时	使用弹性吞吐量在文件中写入或读取的数据量。
<i>region</i> -IADataAccess-Bytes	不适用	GB	每小时	在 EFS 不频繁访问（IA）存储中传输或访问的数据量。
<i>region</i> -IATimedStorage-ByteHrs	StorageElasticity.IA	GB-Mo	每小时	数据存储在 EFS IA 存储中的 GB-月数。
<i>region</i> -IATimedStorage-ET-ByteHrs	StorageElasticity.IA	GB-Mo	每小时	数据存储在 EFS IA 存储中的 GB-月数，适用于使用弹性吞吐量的文件。

使用情况类型	CloudWatch 指标	单位	粒度	说明
<i>region</i> -IATimedStorage-ET-SmallFiles	StorageElasticity.IASizeOverhead	GB-Mo	每小时	小文件（小于 128 KB）存储在 EFS IA 存储中的 GB-月数，适用于使用弹性吞吐量的文件。
<i>region</i> -IATimedStorage-SmallFiles	StorageElasticity.IASizeOverhead	GB-Mo	每小时	小文件（小于 128 KB）存储在 EFS IA 存储中的 GB-月数。
<i>region</i> -IATimedStorage-Z-ByteHrs	StorageElasticity.IA	GB-Mo	每小时	存储在单区文件系统的 EFS IA 存储中的数据的实际大小。
<i>region</i> -IATimedStorage-Z-SmallFiles	StorageElasticity.IASizeOverhead	GB-Mo	每小时	小对象（小于 128 KB）存储在单区文件系统的 EFS IA 存储中的 GB-月数。
<i>region</i> -IncludedTP-MiBpsHrs	PermittedThroughput	MiBps-Mo	每小时	文件系统可以驱动的最大吞吐量。
<i>region</i> -ProvisionedTP-MiBpsHrs	PermittedThroughput	MiBps-Mo	每小时	预置吞吐量或支持的最大吞吐量，适用于使用预置吞吐量的文件系统。
<i>region</i> -TimedStorage-ByteHrs	StorageElasticity.Standard	GB-Mo	每小时	存储在区域文件系统的 EFS 标准存储中的实际数据总量。

使用情况类型	CloudWatch 指标	单位	粒度	说明
<i>region-</i> TimedStorage-Z-ByteHrs	StorageElasticStorageStandard	GB-Mo	每小时	存储在单区文件系统的 EFS 标准存储中的实际数据总量。

注意

- 即使在 90 天之前删除数据，EFS 存档存储也会按最低 90 天的存储期限收费。
- 分层费用适用于从 EFS 标准存储分层到 EFS 不频繁访问 (IA) 存储，以及从 EFS IA 存储分层到 EFS 标准存储的数据。
- EFS IA 和 EFS 存档存储的最小计费文件大小为 128 KiB。小于 128 KiB 的文件可以分层到这些存储类别，但按适当的存储类别费率收取 128 KiB 的存储费用。
- 对于使用弹性吞吐量的文件系统，元数据操作在前 4 KiB 之后以 1 KiB 的增量计量，数据操作在前 32 KiB 之后以 1 KiB 的增量计量。
- 存储使用量以二进制千兆字节 (GB) 计算，其中 1 GB 等于 2^{30} 字节。有关如何计量存储和吞吐量使用情况以计算您的 Amazon EFS 账单的更多详细信息，请参阅[Amazon EFS 如何报告文件系统和对象大小](#)。
- GB-月的计算方法是：取 GB 小时总数，将一个月内的这些数量汇总，然后除以该月的小时数。

跟踪使用情况报告中的操作

操作描述指定的使用类型对您的 EFS 文件系统所采取的操作。操作由不言自明的代码表示，如 Read 或 Delete。要查看文件系统上的哪些操作生成了特定类型的用途，请使用这些代码。当您创建使用率报告时，可以选择包括 All Operations 或特定操作 (例如 Write) 以进行报告。

Amazon EFS 性能规范

以下章节概述了 Amazon EFS 性能，还介绍了文件系统配置如何影响关键性能维度。我们还提供了一些用于优化文件系统性能的重要提示和建议。

主题

- [性能摘要](#)
- [存储类](#)
- [性能模式](#)
- [吞吐量模式](#)
- [Amazon EFS 性能提示](#)
- [Amazon EFS 性能问题排查](#)
- [排查 AMI 和内核问题](#)

性能摘要

文件系统性能通常使用延迟、吞吐量和每秒 Input/Output 操作数 (IOPS) 等维度来衡量。Amazon EFS 在这些方面的性能取决于文件系统的配置。以下配置会影响 Amazon EFS 文件系统的性能：

- 文件系统类型 – 区域性或单区
- 性能模式-通用或最大 I/O

Important

与通用 I/O 性能模式相比，最大性能模式的每次操作延迟更高。为了提高性能，我们建议始终使用通用性能模式。有关更多信息，请参阅 [性能模式](#)。

- 吞吐量模式 – 弹性、预配置或突增

下表概述了使用通用性能模式的文件系统的性能规格，以及不同文件系统类型和吞吐量模式的可能组合。

使用通用性能模式的文件系统的性能规格

存储和吞吐量配置		延迟 ¹		最大 IOPS		最大吞吐量		
文件系统类型	吞吐量模式	读取操作	写入操作	读取操作	写入操作	Per-file-system 阅读 ²	Per-file-system 写下 ²	Per-client read/write
区域性	弹性	~1 毫秒 (ms)	大约 2.7 毫秒 (ms)	900,000—2,500,000 ³	500,000 ³	每秒 20—60 千兆字节 () GiBps	1—5 GiBps	每秒 1,500 兆字节 (4) MiBps
区域性	已预置	大约 1 毫秒	大约 2.7 毫秒	55,000	25,000	3—10 GiBps	1—3.33 GiBps	500 MiBps
区域性	突增	大约 1 毫秒	大约 2.7 毫秒	35,000	7,000	3—5 GiBps	1—3 GiBps	500 MiBps
单区	弹性、预配置、爆发	大约 1 毫秒	大约 1.6 毫秒	35,000	7,000	3 GiBps ⁵	1 GiBps ⁵	500 MiBps

1. 所示延迟值代表了最佳条件下的最佳性能。实际结果可能因网络、工作负载和系统因素而异。
2. 最大读取和写入吞吐量取决于 AWS 区域。如果吞吐量超过最大吞吐量，则需要增加吞吐量配额。AWS 区域 Amazon EFS 服务团队会根据具体情况考虑任何额外吞吐量请求。是否批准可能取决于工作负载类型。有关请求增加配额的更多信息，请参阅[Amazon EFS 限额](#)。
3. 默认情况下，使用弹性吞吐量的文件系统可针对不频繁访问的数据，实现高达 90,000 次读取 IOPS；而针对频繁访问的数据，可以实现 250,000 次读取 IOPS 和 50,000 次写入 IOPS。如果您的工作负载需要更高 IOPS，您可以请求将这些数字增加高达 10 倍。有关更多信息，请参阅[您可以提高的 Amazon EFS 配额](#)。其它建议适用于实现最大 IOPS。有关更多信息，请参阅[the section called “优化需要高吞吐量和 IOPS 的工作负载”](#)。
4. MiBps 对于使用弹性吞吐量并使用 2.0 或更高版本的 Amazon EFS 客户端 (amazon-efs-utils 版本) 或 Amazon EFS CSI 驱动程序 (aws-efs-csi-driver) 安装的文件系统，最大合并读写吞吐量

为 1,500。对于所有其他文件系统，吞吐量限制为 500 MiBps。有关 Amazon EFS 客户端的更多信息，请参阅[安装 Amazon EFS 客户端](#)。

5. 使用突发吞吐量的 One Zone 文件系统可以驱动与使用突发吞吐量的区域文件系统相同的每个文件系统的读取和写入吞吐量（读取的最大读取量为 5 GiBps，写入的最大读取量为 3）。GiBps

存储类

Amazon EFS 存储类专为实现最有效的存储而设计，具体取决于用例。

- EFS 标准存储类使用固态硬盘 (SSD) 存储为频繁访问的文件提供最低延迟级别。该存储类为读取提供低至 1 毫秒的首字节延迟，写入的首字节延迟低至 2.7 毫秒。
- EFS 不频繁访问 (IA) 和 EFS 存档存储类可存储访问频率较低的数据，这些数据不需要频繁访问的数据所需的延迟性能。这些存储类提供的第一个字节延迟为数十毫秒。

有关 EFS 存储类的更多信息，请参阅 [the section called “EFS 存储类”](#)。

性能模式

Amazon EFS 提供两种性能模式，通用模式和最大性能模式 I/O。

- 通用模式的每次操作延迟最低，是文件系统的默认性能模式。单区文件系统始终使用通用性能模式。为了提高性能，我们建议始终使用通用性能模式。
- Max I/O 模式是上一代性能类型，专为高度并行化的工作负载而设计，这些工作负载可以容忍比通用模式更高的延迟。One Zone 文件系统或使用弹性吞吐量的文件系统不支持最大 I/O 模式。

Important

由于 Max 的每次操作延迟较高 I/O，因此我们建议对所有文件系统使用通用性能模式。

为了帮助确保您的工作负载保持在使用通用性能模式的文件系统可用的 IOPS 限制范围内，您可以监控该 PercentIOLimit CloudWatch 指标。有关更多信息，请参阅 [CloudWatch 亚马逊 EFS 的指标](#)。

应用程序可以弹性扩展其 IOPS，以达到与性能模式相关的限制。无需单独为 IOPS 付费；它们已包含在文件系统的吞吐量核算中。每个网络文件系统 (NFS) 请求都以 4 千字节 (KB) 吞吐量或其实际请求和响应大小计算，以较大者为准。

吞吐量模式

文件系统的吞吐量模式决定了文件系统可用的吞吐量。Amazon EFS 提供三种吞吐量模式：弹性、预配置和突增。读取吞吐量已打折，使您能够获得比写入吞吐量更高的读取吞吐量。每种吞吐量模式下可用的最大吞吐量取决于 AWS 区域。有关不同区域中最大文件系统吞吐量的更多信息，请参阅[Amazon EFS 限额](#)。

文件系统可以实现 100% 的读写组合吞吐量。例如，如果文件系统使用其读取吞吐量限制的 33%，则该文件系统可以同时达到其写入吞吐量限制的 67%。可以在控制台的文件系统详细信息页面上的吞吐量利用率（%）图表中监控文件系统的吞吐量使用情况。有关更多信息，请参阅[监控吞吐量性能](#)。

为文件系统选择正确的吞吐量模式

为文件系统选择正确的吞吐量模式取决于工作负载的性能要求。

- 弹性吞吐量（推荐）– 当工作负载激增或不可预测且性能要求难以预测时，或者当应用程序以 5% 或更低的平均峰值比驱动吞吐量时，请使用默认的弹性吞吐量。有关更多信息，请参阅[弹性吞吐量](#)。
- 预置吞吐量 – 如果您知道工作负载的性能要求，或者当您的应用程序以 5% 或更高的平均峰值比驱动吞吐量时，请使用预置吞吐量。有关更多信息，请参阅[预配置吞吐量](#)。
- 突增吞吐量 – 如果您希望吞吐量随文件系统存储量扩展，请使用突增吞吐量。

如果在使用突增吞吐量后，发现您的应用程序受到吞吐量限制（例如，使用的吞吐量超过允许吞吐量的 80%，或者已经使用了所有突增点数），则应使用弹性吞吐量或预置吞吐量。有关更多信息，请参阅[突增吞吐量](#)。

有关 Amazon EFS 指标的更多信息，请参阅[CloudWatch 亚马逊 EFS 的指标](#)。

弹性吞吐量

对于使用弹性吞吐量的文件系统，Amazon EFS 会自动扩缩吞吐量性能，以满足工作负载活动需求。对于性能要求难以预测的激增或不可预测的工作负载，或者对于以平均峰值吞吐量的 5% 或更低（平均峰值比）驱动吞吐量的应用程序，弹性吞吐量是最佳吞吐量模式。

由于具有弹性吞吐量的文件系统的吞吐量性能会自动扩缩，因此无需指定或预置吞吐能力来满足应用程序需求。只需为读取或写入的元数据和数据量付费，并且在使用弹性吞吐量时不会累积或消耗突增点数。

 Note

虽然 Elastic 吞吐量旨在根据您的吞吐量进行弹性扩展，但我们建议您通过使用 CloudWatch (MeteredIOBytes) 监控指标和使用情况警报来实施适当的治理，以此作为最佳运营实践的一部分。这有助于保持资源的最优利用率，并确保始终保持在计划的操作参数范围内。有关更多信息，请参阅 [使用 Amazon 监控指标 CloudWatch](#)。

有关每个区域的弹性吞吐量限制的信息，请参阅[您可以提高的 Amazon EFS 配额](#)。

预配置吞吐量

使用预置吞吐量时，可以指定文件系统可以驱动的吞吐量级别，不受文件系统大小或突增点数余量影响。如果您知道工作负载的性能要求，或者当您的应用程序以 5% 或更高的平均峰值比驱动吞吐量时，请使用预置吞吐量。

对于使用预置吞吐量的文件系统，会针对为文件系统启用的吞吐量向您收费。一个月内计费的吞吐量基于预配置的吞吐量，该吞吐量超过文件系统包含的标准存储的基准吞吐量，不超过 AWS 区域中现行的突增基准吞吐量限制。

如果文件系统的基准吞吐量超过预配置的吞吐量，则它会自动使用文件系统允许的突增吞吐量（不超过其中的现行突发基准吞吐量限制）。AWS 区域

有关每个区域的预置吞吐量限制的信息，请参阅[您可以提高的 Amazon EFS 配额](#)。

突增吞吐量

对于需要吞吐量随文件系统存储量扩展的工作负载，建议使用突增吞吐量。使用突增吞吐量时，基本吞吐量与标准存储类中的文件系统大小成正比，每 GiB 存储空间的速率为 KiBps 每 GiB 存储 50。当文件系统消耗的吞吐量低于其基本吞吐量速率时，突增点数就会累积，当吞吐量超过基本速率时，会扣除突增点数。

当有突发积分可用时，文件系统可以在标准存储中将吞吐量提高到 MiBps 每个 TiB 100（每 KiBps GiB 100 个），但不 AWS 区域 超过限制，最小值为 100。MiBps 如果没有可用的突发积分，则文件系统最多可以驱动 MiBps 每 TiB 存储 50 个，最少为 1。MiBps

有关每个区域的突增吞吐量的信息，请参阅[General resource quotas that cannot be changed](#)。

了解 Amazon EFS 突增点数

使用突增吞吐量，每个文件系统随时间推移获得基准速率的突增点数，该基准速率由存储在 EFS 标准存储类中的文件系统大小决定。基准速率为 MiBps 每 TiB [TiB] 存储 50 个（相当于每 KiBps GiB 存储 50 个）。Amazon EFS 可将读取操作计量到写入操作速率的三分之一，从而允许文件系统将基准速率提高到 KiBps 每 GiB 读取吞吐量 150，或每 KiBps GiB 写入吞吐量 50。

文件系统可以其基准计量速率持续提高吞吐量。每当文件系统处于不活动状态或吞吐量低于其基准计量速率时，文件系统就会累积突增点数。累计的突增积分使文件系统可以推高吞吐量，使其高于其基准速率。

例如，标准存储类中具有 100 GiB 计量数据的文件系统的基准吞吐量为 5。MiBps 在 24 小时的非活动状态下，文件系统将获得价值 43.2 万 MiB (5 MiB × 86,400 秒 = 432,000 MiB) 的积分，这些积分可用于以 100 的速度突发持续 72 分钟 (432,000 MiB ÷ 100 MiBps = 72 分钟)。MiBps

如果大于 1 TiB 的文件系统在 50% 的时间内处于不活动状态，该文件系统在其余 50% 的时间内始终可以突增。

下表提供了突增行为的示例。

文件系统大小	突增吞吐量	基准吞吐量
标准存储中有 100 GiB 计量数据	- 突发至 300 (MiBps) 只读模式，每天最多 72 分钟，或者 - 突增至 100 MiBps 只写模式，每天最长可持续 72 分钟	- 连续驱动最多 15 个 MiBps 只读驱动器 - 连续驱动最多 5 个只 MiBps 写模式
标准存储中有 1 TiB 计量数据	- 突增至 300 MiBps 只读模式，每天 12 小时，或者 - 在每天 12 小时内突增至 100 MiBps 只写模式	- 驱动器 150 持续 MiBps 为只读状态 - 连续驱动 50 只 MiBps 写模式
标准存储中有 10 TiB 计量数据	- 以 GiBps 只读模式突变为 3，每天 12 小时，或者 - 每天 12 小时内突变为 1 GiBps 只写模式	- 驱动器 1.5 持续 GiBps 只读 - 连续驱动 500 只 MiBps 写模式

文件系统大小	突增吞吐量	基准吞吐量
通常，较大的文件系统	- 每天 TiB 存储空间突增至 300 个 MiBps 只读状态，持续 12 小时，或者 - 每天 12 小时内每 TiB 存储空间可突增至 100 MiBps 只写入	- 每 TiB 存储空间连续驱动 150 个 MiBps 只读存储空间 - 每 TiB 存储空间连续驱动 50 个 MiBps 只写操作

Note

Amazon EFS 为所有文件系统提供的计量吞吐量为 1 MiBps，即使基准速率较低。

确定基准速率和突增速率时所使用的文件系统大小是通过 [DescribeFileSystems](#) API 操作可用的 ValueInStandard 计量大小。

小于 1 TiB 的文件系统可以获得的积分可达到最高 2.1 TiB 积分余额，对于大于 1 TiB 的文件系统，可达到每 TiB 存储 2.1 TiB 的积分余额。此行为意味着文件系统可以累积足够的点数来持续突增长达 12 小时。

对切换吞吐量和更改预置量的限制

可以切换现有文件系统的吞吐量模式并更改吞吐量。但是，在将吞吐量模式切换到预置吞吐量或更改预置吞吐量后，以下操作将在 24 小时内受到限制：

- 从预置吞吐量模式切换到弹性或突增吞吐量模式。
- 减少预置吞吐量。

Amazon EFS 性能提示

在使用 Amazon EFS 时，请记住以下性能提示。

平均 I/O 大小

Amazon EFS 的分布式特性实现了高水平的可用性、持久性和可扩展性。这种分布式架构使得每次文件操作只产生很小的延迟开销。由于每次操作都有这种延迟，因此总体吞吐量通常会随着平均 I/O 大小的增加而增加，因为开销会分摊到较大的数据量上。

优化需要高吞吐量和 IOPS 的工作负载

对于需要高吞吐量和 IOPS 的工作负载，请使用配置了通用性能模式和弹性吞吐量的区域性文件系统。

Note

要对频繁访问的数据实现最大读取 IOPS，文件系统必须使用弹性吞吐量。

要实现最高级别的性能，必须通过按如下方式配置应用程序或工作负载来利用并行化。

1. 在所有客户端和目录之间均匀分配工作负载，目录数至少与使用的客户端数量相同。
2. 通过将各个线程与不同的数据集或文件对齐，最大限度地减少争用。
3. 将工作负载分配到 10 个或更多的 NFS 客户端，单个装载目标中每个客户端至少有 64 个线程。

同时连接

您可以同时在多达数千个 Amazon EC2 和其他 AWS 计算实例上挂载 Amazon EFS 文件系统。如果可以跨更多实例并行执行应用程序，则可以在跨计算实例的聚合中提高文件系统的吞吐量级别。

请求模型

如果启用对文件系统的异步写入，则待处理的写入操作会在 Amazon EC2 实例上缓冲，然后再异步写入 Amazon EFS。异步写入通常具有较低的延迟。在执行异步写入时，内核使用额外内存进行缓存。

启用了同步写入的文件系统或使用绕开缓存选项（例如 `O_DIRECT`）打开文件的文件系统将向 Amazon EFS 发出同步请求。每个操作都将在客户端和 Amazon EFS 之间往返一次。

Note

您选择的请求模型将在一致性（如果您使用多个 Amazon EC2 实例）和速率之间进行取舍。使用同步写入可以在处理下一个请求之前完成每个写入请求事务，从而提高数据一致性。使用异步写入可通过缓冲待处理的写入操作来提高吞吐量。

NFS 客户端挂载设置

确认您使用的是 [挂载 EFS 文件系统](#) 和 [Linux 的挂载注意事项](#) 中推荐的挂载选项。

在亚马逊 EC2 实例上安装文件系统时，Amazon EFS 支持网络文件系统版本 4.0 和 4.1 (NFSv4) 协议。NFSv4.1 与 (每秒少于 1,000 个文件) 相比，为并行小文件读取操作 NFSv4.0 (每秒大于 10,000 个文件) 提供了更好的性能。仅支持运行 macOS Big Sur 的亚马逊 EC2 macOS 实例。NFSv4.0

请勿使用以下挂载选项：

- `noac`、`actimeo=0`、`acregmax=0`、`acdirmax=0` – 这些选项会禁用属性缓存，这会对性能产生非常大的影响。
- `lookupcache=pos`、`lookupcache=none` – 这些选项会禁用文件名查找缓存，这会对性能产生非常大的影响。
- `fsc` – 此选项启用本地文件缓存，但不会更改 NFS 缓存的一致性，也不会减少延迟。

Note

在挂载文件系统时，请考虑将 NFS 客户端的读写缓冲区大小增加到 1 MB。

优化小文件性能

可以通过最大限度地减少文件重新打开次数、增加并行度，以及尽可能捆绑参考文件来提高小文件的性能。

- 尽量减少往返服务器的次数。

如果以后在工作流中需要文件，请不要不必要地关闭这些文件。保持文件描述符处于打开状态可以直接访问缓存中的本地副本。文件打开、关闭和元数据操作通常不能以异步方式或通过管道进行。

读取或写入小文件时，两次额外往返非常重要。

每次往返 (文件打开、文件关闭) 所花费的时间可能与读取或写入兆字节批量数据一样多。在计算作业开始时打开一次输入或输出文件，并在整个作业期间保持打开状态会更有效。

- 使用并行度来减少往返时间的影响。
- 将参考文件捆绑到 .zip 文件中。有些应用程序使用大量较小的主要是只读文件的参考文件。将这些文件捆绑到一个 .zip 文件中，只需一次打开-关闭往返操作即可读取多个文件。

.zip 格式允许随机访问单个文件。

优化目录性能

在同时修改的超大目录（超过 10 万个文件）上执行列出操作（ls）时，Linux NFS 客户端可能会挂起而不返回响应。此问题已在内核 5.11 中修复，该内核已移植到 Amazon Linux 2 内核 4.14、5.4 和 5.10。

我们建议您在可能的情况下，将文件系统上的目录数保持在 1 万以内。尽可能多地使用嵌套子目录。

列出目录时，如果不需要文件属性，应避免获取这些属性，因为它们并未存储在目录本身中。

优化 NFS read_ahead_kb 的大小

NFS read_ahead_kb 属性定义了 Linux 内核在顺序读取操作期间要提前读取或预取的千字节数。

对于 5.4.* 之前的 Linux 内核版本，read_ahead_kb 值是通过 NFS_MAX_READAHEAD 乘以 rsize（挂载选项中设置的客户端配置的读取缓冲区大小）的值来设置的。使用[推荐的挂载选项](#)时，此公式将 read_ahead_kb 设置为 15 MB。

Note

从 Linux 内核版本 5.4.* 开始，Linux NFS 客户端使用默认 read_ahead_kb 值 128 KB。我们建议将此值增加到 15 MB。

挂载文件系统后，amazon-efs-utils 版本 1.33.2 及更高版本中提供的 Amazon EFS 挂载帮助程序会自动将 read_ahead_kb 值修改为等于 15 * rsize 或 15 MB。

对于 Linux 内核 5.4 或更高版本，如果不使用挂载帮助程序来挂载文件系统，请考虑手动将 read_ahead_kb 设置为 15 MB 以提高性能。挂载文件系统后，可使用以下命令重置 read_ahead_kb 值。在使用此命令之前，替换以下值：

- 将 *read-ahead-value-kb* 替换为所需的大小（以千字节为单位）。
- 将 *efs-mount-point* 替换为文件系统的挂载点。

```
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"
```

以下示例将 `read_ahead_kb` 大小设置为 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"
```

Amazon EFS 性能问题排查

通常，如果您遇到难以解决的 Amazon EFS 问题，请确认您使用的是最新 Linux 内核。如果使用的是企业 Linux 发行版，我们建议您使用以下版本：

- 内核版本为 4.3 或更高版本的 Amazon Linux 2
- Amazon Linux 2015.09 或更高版本
- RHEL 7.3 或更高版本
- 所有 Ubuntu 16.04 版本
- 具有内核 3.13.0-83 或更高版本的 Ubuntu 14.04
- SLES 12 Sp2 或更高版本

如果使用其他发行版或自定义内核，我们建议您使用内核 4.3 或更高版本。

Note

由于[并行打开多个文件时，性能不佳](#)，RHEL 6.9 可能对于特定工作负载不够理想。

主题

- [无法创建 EFS 文件系统](#)
- [拒绝访问 NFS 文件系统上允许的文件](#)
- [访问 Amazon EFS 控制台时出错](#)
- [Amazon EC2 实例挂起](#)
- [写入大量数据的应用程序挂起](#)
- [并行打开多个文件时，性能不佳](#)
- [自定义 NFS 设置导致写入延迟](#)
- [使用 Oracle Recovery Manager 创建备份的速度很慢](#)

无法创建 EFS 文件系统

创建 EFS 文件系统的请求失败，并显示以下消息：

```
User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

要采取的操作

检查您的 AWS Identity and Access Management (IAM) 策略，确认您有权创建具有指定资源条件的 EFS 文件系统。有关更多信息，请参阅 [Amazon EFS 的身份和访问管理](#)。

拒绝访问 NFS 文件系统上允许的文件

当分配了超过 16 个访问组 IDs (GIDs) 的用户尝试在 NFS 文件系统上执行操作时，可能会拒绝他们访问文件系统上允许的文件。[之所以出现此问题，GIDs 是因为 NFS 协议支持 GIDs 每个用户最多 16 个，并且根据 RFC 5531 中的定义，NFS 客户端请求中的任何其他请求都会被截断。](#)

要采取的操作

重组您的 NFS 用户和组映射，以便为每个用户分配的访问组不超过 16 个 ()。GIDs

访问 Amazon EFS 控制台时出错

本节介绍用户在访问 Amazon EFS 管理控制台时可能遇到的错误。

对的 **ec2:DescribeVPCs** 凭证进行身份验证时出错

访问 Amazon EFS 控制台时会显示以下错误消息：

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

此错误表示您的登录凭证未成功通过 Amazon EC2 服务的身份验证。在您选择的 VPC 中创建 EFS 文件系统时，Amazon EFS 控制台会代表您调用 Amazon EC2 服务。

要采取的操作

确保正确设置了客户端访问 Amazon EFS 控制台的时间。

Amazon EC2 实例挂起

Amazon EC2 实例挂起的原因可能是，您在未首先卸载文件系统的情况下删除了文件系统挂载目标。

要采取的操作

在删除文件系统挂载目标之前，请卸载文件系统。有关卸载您的 Amazon EFS 文件系统的更多信息，请参阅[卸载文件系统](#)。

写入大量数据的应用程序挂起

将大量数据写入 Amazon EFS 的应用程序挂起，并导致实例重新启动。

要采取的操作

如果应用程序需要太长时间才能将其所有数据写入 Amazon EFS，则 Linux 可能会重新启动，因为进程似乎已没有响应。两个内核配置参数可定义此行为，即 `kernel.hung_task_panic` 和 `kernel.hung_task_timeout_secs`。

在以下示例中，在实例重启之前，`ps` 命令将挂起的进程状态报告为 D，表明该进程正在等待 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
```

要防止重新启动，请增加超时期限或禁用检测到挂起任务时的内核崩溃。以下命令将禁用大多数 Linux 系统上的挂起任务内核崩溃。

```
$ sudo sysctl -w kernel.hung_task_panic=0
```

并行打开多个文件时，性能不佳

并行打开多个文件的应用程序不会体验到预期的并行 I/O 性能提升。

要采取的操作

网络文件系统版本 4 (NFSv4) 客户端和使用 NFSv4 .1 的 RHEL 6 客户端上会出现此问题，因为这些 NFS 客户端会序列化 NFS 打开和关闭操作。请使用 NFS 协议版本 4.1 和建议的 [Linux 发行版](#)（不存在此问题）之一。

如果您不能使用 NFSv4 .1，请注意 Linux NFSv4 .0 客户端会按用户 ID 和组对打开和关闭请求进行序列化。IDs 即使多个进程或多个线程同时发出请求，也会发生此序列化。当所有操作都 IDs 匹配时，客户端一次只向 NFS 服务器发送一个打开或关闭操作。要解决这些问题，可以执行下列任一操作：

- 您可以在同一 Amazon EC2 实例上通过不同用户 ID 运行每个进程。

- 您可以让所有未处理 IDs 的请求中的用户保持不变，改为修改群组 IDs 集。
- 您可以从单独的 Amazon EC2 实例运行每个进程。

自定义 NFS 设置导致写入延迟

您可以自定义 NFS 客户端设置，Amazon EC2 实例需要最多三秒钟时间来查看通过其他 Amazon EC2 实例对文件系统执行的写入操作。

要采取的操作

如果遇到该问题，可以通过以下任一方法加以解决：

- 如果 Amazon EC2 实例上读取数据的 NFS 客户端已激活属性缓存，请卸载文件系统。然后，使用 `noac` 选项重新挂载文件系统以禁用属性缓存。默认情况下，NFSv4.1 中的属性缓存处于启用状态。

Note

禁用客户端缓存可能会降低您的应用程序性能。

- 您还可以通过使用与 NFS 过程兼容的编程语言来按需清除您的属性缓存。要执行该操作，您可以在发送 ACCESS 过程请求后立即发送读取请求。

例如，您可以使用 Python 编程语言构造以下调用。

```
# 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)
```

使用 Oracle Recovery Manager 创建备份的速度很慢

如果在启动备份作业之前 Oracle Recovery Manager 暂停 120 秒，使用 Oracle Recovery Manager 创建备份的速度可能很慢。

要采取的操作

如果遇到该问题，请禁用 Oracle 直接 NFS，如 Oracle 帮助中心的[启用和禁用 NFS 的直接 NFS 客户端控制](#)中所述。

 Note

Amazon EFS 不支持 Oracle 直接 NFS。

排查 AMI 和内核问题

下文介绍了如何排查在从 Amazon EC2 实例使用 Amazon EFS 时遇到的与特定亚马逊机器映像（AMI）或内核版本相关的问题。

主题

- [无法更改所有权](#)
- [由于客户端错误，文件系统重复执行操作](#)
- [客户端发生死锁](#)
- [列出大型目录中的文件需要很长时间](#)

无法更改所有权

您无法 file/directory 使用 Linux chown 命令更改的所有权。

出现该错误的内核版本

2.6.32

要采取的操作

您可以执行以下操作以解决该错误：

- 如果要运行 chown 以执行更改 EFS 根目录所有权所需的一次性设置步骤，您可以从运行较新内核的实例中运行 chown 命令。例如，使用最新版本的 Amazon Linux。
- 如果 chown 是您的生产工作流程的一部分，则您必须更新内核版本才能使用 chown。

由于客户端错误，文件系统重复执行操作

由于某个客户端错误，文件系统重复执行操作。

要采取的操作

将客户端软件更新为最新版本。

客户端发生死锁

客户端变为死锁状态。

出现该错误的内核版本

- 内核为 Linux 3.10.0-229.20.1.el7.x86_64 的 CentOS-7
- 内核为 Linux 4.2.0-18-generic 的 Ubuntu 15.10

要采取的操作

请执行以下操作之一：

- 升级为更新的内核版本。对于 CentOS-7，内核版本 Linux 3.10.0-327 或更高版本中包含相应的修复程序。
- 降级为较旧的内核版本。

列出大型目录中的文件需要很长时间

如果在您的 NFS 客户端遍历目录以完成列出操作时，目录正在发生更改，则可能会出现这种情况。每当 NFS 客户端在这种遍历期间注意到目录内容发生更改时，它都会从头开始重新遍历。因此，对于包含经常更改的文件的大型目录，ls 命令可能需要很长时间才能完成。

出现该错误的内核版本

低于 2.6.32-696.el6 的 CentOS 和 RHEL 内核版本

要采取的操作

要解决该问题，请升级到较新的内核版本。

在 Amazon EFS 中备份和复制数据

为了保护您的数据，Amazon EFS 会自动备份您的 EFS 文件系统。为了进一步提高韧性和数据保护，您可以在 AWS 区域中复制 EFS 文件系统。备份和复制 EFS 文件系统可确保在 EFS 文件系统数据出现问题时能够提供持续的操作或服务。例如，在数据损坏或数据丢失的情况下。

主题

- [备份 EFS 文件系统](#)
- [复制 EFS 文件系统](#)

备份 EFS 文件系统

Amazon EFS 与一项完全托管的 AWS Backup、基于策略的服务进行了原生集成，您可以使用它来创建和管理备份策略，以保护您在 Amazon EFS 中的数据。默认情况下，系统会自动备份您使用 Amazon EFS 控制台创建的文件系统。AWS Backup 使用 CLI 或 API 创建文件系统时，默认情况下仅对单区文件系统启用自动备份。

使用 AWS Backup 适用于 Amazon EFS，您可以执行以下操作：

- 通过配置备份计划，管理自动化备份安排和保留。您可以指定备份频率、备份时间、保留备份的时长以及备份的生命周期策略。
- 恢复 Amazon EFS 数据的备份。可以将文件系统数据恢复到新的或现有的文件系统。还可以选择是执行完整恢复还是项目级恢复。

有关使用的更多信息 AWS Backup，请参阅《AWS Backup 开发人员指南》AWS Backup 中的[入门指南](#)。

主题

- [如何 AWS Backup 与 Amazon EFS 配合使用](#)
- [所需的 IAM 权限](#)
- [备份性能](#)
- [管理 EFS 文件系统的自动备份](#)

如何 AWS Backup 与 Amazon EFS 配合使用

默认情况下，系统会自动备份您使用 Amazon EFS 控制台创建的文件系统。AWS Backup 使用 CLI 或 API 创建文件系统时，默认情况下仅对单区文件系统启用自动备份。在创建 EFS 文件系统后，您可以使用 AWS CLI 或 API 开启自动备份。默认 EFS 备份计划使用 AWS Backup 推荐的自动备份设置，即保留期为 35 天的每日备份。使用默认 EFS 备份计划创建的备份存储在默认 EFS 备份库中，该备份库也是由 Amazon EFS 代表您创建的。无法删除默认备份计划和备份保管库。

无论数据的存储类为何，EFS 文件系统中的所有数据都将备份。当要备份的 EFS 文件系统启用了生命周期管理并且具有不频繁访问 (IA) 或存档存储类的数据时，不会产生数据访问费用。还原恢复点时，会将所有文件还原到标准存储类别。

增量备份

AWS Backup 执行 EFS 文件系统的增量备份。在初始备份期间，将创建整个文件系统的副本。在该文件系统的后续备份期间，只复制已更改、已添加或已删除的文件和目录。每次增量备份时，都会 AWS Backup 保留必要的参考数据以进行完全恢复。由于无需复制数据，这种方法将最大限度缩短完成备份所需的时间和节省存储成本。

备份一致性

Amazon EFS 旨在提供高度可用性。在 AWS Backup 中进行备份时，您可以访问和修改 EFS 文件系统。但是，如果在执行备份时对文件系统进行了修改，则可能会出现不一致，例如重复、偏差或排除的数据。这些修改包括写入、重命名、移动或删除操作。为确保一致的备份，我们建议您在备份过程中暂停修改文件系统的应用程序或进程。或者，将备份安排在不修改文件系统期间。

备份完成窗口

您可以视需要为备份指定完成窗口。此窗口定义需要完成备份的时间段。如果指定完成窗口，请确保考虑预期的性能以及文件系统的大小和构成。这样做有助于确保您的备份可以在窗口期间完成。

在指定窗口期间未完成的备份将标记为不完整状态。在下次定时备份期间，将从中断的位置 AWS Backup 恢复。您可以在 AWS Backup 管理控制台上查看所有备份的状态。

备份完成时段的最佳实践

为 EFS 文件系统配置备份完成时段时，请考虑以下最佳实践，以确保备份操作的可靠性：

- 监控备份持续时间 - 如果您发现备份需要较长的时间（例如 20 天或更长时间），请考虑将完成时段缩短到 3 天。这种方法 AWS Backup 允许执行多个较短的部分备份，而不必尝试在单个延长的窗口内完成整个备份。

- 考虑文件系统的特征 - 对于具有数百万个文件或大量数据量的大型文件系统，较短的完成时段可能更为有利。建议的 3 天时段有助于确保备份操作成功完成，并在需要时高效地恢复。
- 计划增量进度-使用较短的完成窗口时，在下次定时备份期间 AWS Backup 会自动从上次停下来的位置恢复。对于大型文件系统，这种增量方法比尝试在单个较长时段中完成所有备份更可靠。
- 测试您的备份策略 - 在依赖生产环境中的备份配置之前，请根据您的特定文件系统大小和特征对其进行测试，以确定适合您的使用案例的最佳完成时段。

Note

较短的完成窗口并不表示处理大型文件系统的能力存在限制。AWS Backup 相反，较短时段提供了更易于预测和管理的备份过程，可以适应不同的文件系统大小和网络条件。

按需备份

使用 AWS Backup，您可以按需将单个资源保存到备份保管库中。与计划备份不同，您无需创建备份计划即可启动按需备份。您仍然可以为备份分配生命周期，这会自动将恢复点移动到冷存储层，并记录何时删除它。

此外，仅针对最新热备份中已不存在的数据，才 AWS Backup 会自动将数据转换为冷存储。例如，在创建备份时，您的文件系统有 100 个文件，而您在创建备份后的第二天删除了两个文件（100 个文件 - 2 个文件 = 第二天有 98 个文件）。在将数据转到冷存储时，只有两个已删除的文件移至冷存储，其余 98 个文件将作为热存储计费。

并发备份

AWS Backup 将备份限制为每个资源只能进行一次并发备份。因此，如果备份作业已在进行中，则计划备份或按需备份可能会失败。有关 AWS Backup 限制的更多信息，请参阅《AWS Backup Developer Guide》中的 [AWS Backup quotas](#)。

备份删除

默认 EFS 备份保管库访问策略设置为拒绝删除恢复点。要删除 EFS 文件系统的现有备份，必须更改保管库访问策略。如果尝试在不修改保管库访问策略的情况下删除 EFS 恢复点，会收到以下错误消息：

```
"Access Denied: Insufficient privileges to perform this action. Please consult with the account administrator for necessary permissions."
```

要编辑默认备份保管库访问策略，必须具有编辑策略的权限。有关更多信息，请参阅《IAM 用户指南》中的[允许所有 IAM 操作（管理员访问）](#)。

所需的 IAM 权限

AWS Backup 在您的账户中代表您创建服务相关角色。此角色具有执行 Amazon EFS 备份所需的权限。

可以使用 `elasticfilesystem:backup` 和 `elasticfilesystem:restore` 操作来允许或拒绝 IAM 实体（例如用户、组或角色）创建或还原 EFS 文件系统备份的能力。您可以在文件系统策略或基于身份的 IAM 策略中使用这些操作。有关更多信息，请参阅[Amazon EFS 的身份和访问管理](#)和[使用 IAM 控制文件系统访问](#)。

备份性能

通常，您可以预期以下备份和还原速率 AWS Backup。对于某些工作负载，例如包含大型文件或目录的工作负载，速率可能会更低。

- 备份速率为每秒 2,000 个文件或每秒 400 兆字节 (MBps)，以较慢者为准。
- 恢复速率为每秒 1,500 个文件或 200 个文件 MBps，以较慢者为准。

备份操作的最长持续时间 AWS Backup 为 30 天。

使用 AWS Backup 不会消耗累积的突发积分，也不计入通用性能模式文件操作限制。有关更多信息，请参阅[Amazon EFS 文件系统的配额](#)。

管理 EFS 文件系统的自动备份

当您使用 Amazon EFS 控制台创建文件系统时，自动备份默认处于开启状态。在创建文件系统后，您可以使用 AWS CLI 或 API 开启自动备份。

您可以使用 AWS Backup 控制台编辑默认的备份计划设置。有关更多信息，请参阅《AWS Backup 开发人员》中的[管理备份计划](#)。您可以使用[AWS Backup 控制台](#)查看所有自动备份，并编辑默认 EFS 备份计划设置。

启用自动备份后，Amazon EFS 会将值为 `enabled` 的 `aws:elasticfilesystem:default-backup` 系统标签键应用于 EFS 文件系统。

创建文件系统后，您可以使用控制台、或 EFS API 开启或关闭自动备份。AWS CLI

使用控制台

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 在文件系统页面中，选择要打开或关闭自动备份的文件系统，并显示文件系统详细信息页面。
3. 在常规设置面板中，选择编辑。
4.
 - 要打开自动备份，请选择启用自动备份。
 - 要关闭自动备份，请清除启用自动备份。
5. 选择保存更改。

使用 AWS CLI

- 使用 put-backup-policy CLI 命令（相应的 API 操作是 [PutBackupPolicy](#)）打开或关闭现有文件系统的自动备份。
- 可使用以下命令打开自动备份。

```
$ aws efs put-backup-policy --file-system-id fs-01234567 \  
--backup-policy Status="ENABLED"
```

Amazon EFS 使用新备份策略进行响应。

```
{  
  "BackupPolicy": {  
    "Status": "ENABLING"  
  }  
}
```

- 可使用以下命令关闭自动备份。

```
$ aws efs put-backup-policy --file-system-id fs-01234567 \  
--backup-policy Status="DISABLED"
```

Amazon EFS 使用新备份策略进行响应。

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

}

复制 EFS 文件系统

为了增强韧性和数据保护，您可以在 AWS 区域中复制 EFS 文件系统。在 EFS 文件系统中启用复制功能时，Amazon EFS 会自动以透明的方式将源文件系统上的数据和元数据复制到目标文件系统。在发生灾难或进行游戏日练习时，您可以失效转移至副本文件系统。要恢复操作，您可以通过失效自动恢复功能恢复到主文件系统。

为了管理创建目标文件系统并使其与源文件系统保持同步的过程，Amazon EFS 使用复制配置。

创建复制配置后，Amazon EFS 会自动使源文件系统和目标文件系统保持同步。对源文件系统所做的更改不会以 point-in-time 一致的方式传输到目标文件系统。相反，它们是根据复制的上次同步时间进行传输的。上次同步时间表示源和目标之间最后一次成功同步的完成时间。上次同步时对源文件系统所做的更改将复制到目标文件系统，但可能无法复制在上次同步时间之后对源文件系统所做的更改。有关更多信息，请参阅 [查看复制详细信息](#)。

所有提供 Amazon EFS 的 AWS 区域 地区均可进行复制。要复制默认情况下禁用的某个区域中的 EFS 文件系统，必须首先选择加入该区域。有关更多信息，请参阅《AWS General Reference Guide》中的 [Specify which AWS 区域 your account can use](#)。如果以后选择退出区域，Amazon EFS 会暂停该区域的所有复制活动。要恢复该地区的复制活动，请 AWS 区域 再次选择加入。

Note

复制不支持将标签用于基于属性的访问权限控制 (ABAC)。

主题

- [成本](#)
- [复制性能](#)
- [所需的 IAM 权限](#)
- [配置到新 EFS 文件系统的复制](#)
- [配置到现有 EFS 文件系统的复制](#)
- [跨 AWS 账户复制 EFS 文件系统](#)
- [查看复制详细信息](#)
- [删除复制配置](#)

- [使用副本](#)

成本

为便于复制，Amazon EFS 会在目标文件系统中创建隐藏的目录和元数据。这相当于大约 12 兆字节 (MiB) 的计量数据，您需要为其付费。有关为文件系统存储计量的更多信息，请参阅[Amazon EFS 如何报告文件系统和对象大小](#)。

复制性能

当您在失效自动恢复过程中创建新的复制或反转现有复制的方向时，Amazon EFS 会执行初始同步，其中包括一系列支持复制的一次性设置操作。只有在初始同步完成后，才能在目标文件系统中访问复制的数据。完成初始同步所需的时间量取决于源文件系统的大小和其中的文件数等因素。

初始复制完成后，Amazon EFS 将大多数文件系统的恢复点目标 (RPO) 保持在 15 分钟。但是，如果源文件系统的文件更改频繁且文件超过 1 亿个，或者文件大于 100GB，则复制所需的时间可能超过 15 分钟。有关监控上次复制成功完成的的时间的信息，请参见[查看复制详细信息](#)。

您可以使用控制台、AWS Command Line Interface (AWS CLI)、API 和 Amazon 监控上次成功同步的时间 CloudWatch。在中 CloudWatch，使用 E [TimeSinceLastSync](#)FS 指标。有关更多信息，请参阅[查看复制详细信息](#)。

所需的 IAM 权限

Amazon EFS 使用名为 AWSServiceRoleForAmazonElasticFileSystem 的 EFS 服务相关角色或您指定的 IAM 角色，来在源文件系统和目标文件系统之间同步复制操作。要提供 IAM 角色，创建复制配置的 IAM 用户或角色必须拥有 iam:PassRole 权限。有关更多信息，请参阅《[用户指南](#)》中的[授予AWS Identity and Access Management 用户向 AWS 服务传递角色的权限](#)。

- 有关 iam:CreateServiceLinkedRole 的更多信息，请参阅[对 Amazon EFS 使用服务相关角色](#)中的示例。
- 有关自定义 IAM 角色的更多信息，请参阅[创建具有自定义信任策略的 IAM 角色](#)。

Note

如果要执行跨账户复制，则在创建复制配置时必须提供 IAM 角色。严禁使用服务相关角色。有关更多信息，请参阅[跨 AWS 账户复制 EFS 文件系统](#)。

您在创建复制配置时提供的服务相关角色或 IAM 角色必须具有以下复制权限。

- elasticfilesystem:DescribeFileSystems
- elasticfilesystem:CreateFileSystem
- elasticfilesystem:CreateReplicationConfiguration
- elasticfilesystem>DeleteReplicationConfiguration
- elasticfilesystem:DescribeReplicationConfigurations

可以使用 AmazonElasticFileSystemFullAccess 托管式策略自动获取所有必需的 EFS 权限。有关更多信息，请参阅 [AWS 托管策略：AmazonElasticFileSystemFullAccess](#)。

配置到新 EFS 文件系统的复制

Amazon EFS 会自动创建新的文件系统，并将源文件系统上的数据和元数据复制到您选择的新的只读目标文件系统。AWS 区域 复制到新的文件系统时，您可以选择用于加密的文件系统类型和 AWS Key Management Service (AWS KMS) 密钥。而且，Amazon EFS 在创建目标文件系统时不会创建任何挂载目标。创建复制配置后，必须[创建一个或多个挂载目标](#)来[挂载目标文件系统](#)。

Note

一个文件系统只能属于一个复制配置。不能在其他复制配置中将目标文件系统用作源文件系统。

- 文件系统类型 – 文件系统类型决定了 Amazon EFS 文件系统在 AWS 区域中存储数据的可用性和持久性。
 - 选择区域性可创建一个文件系统，该文件系统可跨 AWS 区域中的所有可用区以冗余方式存储数据和元数据。
 - 选择单区可创建一个文件系统，该文件系统在单个可用区内以冗余方式存储数据和元数据。

有关文件系统类型的更多信息，请参阅[EFS 文件系统类型](#)。

- 加密 – 所有目标文件系统都是在启用静态加密的情况下创建的。您可以指定用于加密目标文件系统的密 AWS KMS 钥。如果不指定 KMS 密钥，则使用您的 Amazon EFS 的服务托管式 KMS 密钥。

 Important

创建目标文件系统后，无法更改 KMS 密钥。

目标文件系统使用基于源文件系统的默认设置创建。额外设置可在创建后更改。

- 自动备份 – 对于使用单区存储的目标文件系统，默认情况下启用自动备份。创建文件系统后，可以更改自动备份设置。有关更多信息，请参阅 [管理 EFS 文件系统的自动备份](#)。
- 性能模式 – 目标文件系统的性能模式与源文件系统的性能模式相匹配，除非目标文件系统使用单区存储。在这种情况下，将使用通用模式。无法更改性能模式。
- 吞吐量模式 – 目标文件系统的吞吐量模式与源文件系统的吞吐量模式相匹配。创建文件系统后，可以更改模式。

如果源文件系统的吞吐量模式为预置，则目标文件系统的预置吞吐量与源文件系统的预置吞吐量相匹配，除非源文件的预置量超过目标文件系统区域的限制。如果源文件系统的预配置量超过目标文件系统的区域限制，则目标文件系统的预配置吞吐量为区域限制。有关更多信息，请参阅 [您可以提高的 Amazon EFS 配额](#)。

- 生命周期管理 – 未在目标文件系统上启用生命周期管理。创建目标文件系统后，您可以启用它。有关更多信息，请参阅 [管理存储生命周期](#)。

步骤 1：创建复制配置

复制到新文件系统的第一步是创建复制配置。只有在初始同步完成后，才能访问复制到目标文件系统的数据。初始同步所用时长取决于源文件系统的大小和其中所含文件数等因素。有关复制性能的更多信息，请参阅[复制性能](#)。

使用控制台

1. 登录 AWS 管理控制台 并打开 Amazon EFS 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 打开要复制的文件系统：
 - a. 在左侧导航窗格中，选择文件系统。

- b. 在文件系统列表中，选择要复制的文件系统。选择的文件系统不能是现有复制配置中的源文件系统或目标文件系统。
3. 选择复制选项卡。
4. 在复制部分，选择创建复制。
5. 在复制设置部分，定义复制设置：
 - a. 对于复制配置，选择是要复制到新文件系统还是现有文件系统。
 - b. 在目标中 AWS 区域，选择要 AWS 区域 在其中复制文件系统的目标。
6. 在目标文件系统设置部分中，定义目标文件系统设置。
 - a. 对于文件系统类型，选择文件系统的存储选项：
 - 要创建文件系统，在中多个地理位置分隔的可用区中冗余存储数据 AWS 区域，请选择区域。
 - 要创建在中的单个可用区内以冗余方式存储数据的文件系统 AWS 区域，请选择一个区域，然后选择该可用区。

有关更多信息，请参阅 [EFS 文件系统类型](#)。

 Note

在提供 Amazon EFS 的 AWS 区域 中，并非所有可用区都提供单区文件系统。

- b. 对于加密，在目标文件系统上自动启用静态数据加密。默认情况下，Amazon EFS 使用您的 AWS Key Management Service (AWS KMS) 服务密钥 (aws/elasticfilesystem)。要使用其它 KMS 密钥，请选择 KMS 密钥或输入密钥的 Amazon 资源名称 (ARN)。

 Important

创建文件系统后，无法更改 KMS 密钥。

创建复制配置 (AWS CLI)

本节提供了在 AWS CLI 中使用 `create-replication-configuration` 命令创建复制配置的示例。等效的 API 命令是 [CreateReplicationConfiguration](#)。

Example：为区域性目标文件系统创建复制配置

以下示例为文件系统 *fs-0123456789abcdef1* 创建复制配置。该示例使用Region参数在中创建目标文件系统*eu-west-2* AWS 区域。KmsKeyId 参数指定加密目标文件系统时要使用的 KMS 密钥 ID：

```
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 答案如下：

```
{  
  "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：为单区目标文件系统创建复制配置

以下示例为文件系统 *fs-0123456789abcdef1* 创建复制配置。此示例使用 AvailabilityZoneName 参数在 *us-west-2a* 可用区中创建一个单区目标文件系统。由于未指定 KMS 密钥，因此使用账户的默认 AWS KMS 服务密钥 (aws/elasticfilesystem) 对目标文件系统进行加密。

```
aws efs create-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1 \  
--availability-zone-name us-west-2a
```

```
--destinations AvailabilityZoneName=us-west-2a
```

步骤 2：挂载目标文件系统

Amazon EFS 在创建目标文件系统时不会创建任何挂载目标。要挂载目标文件系统，必须创建一个或多个挂载目标。有关更多信息，请参阅 [挂载 EFS 文件系统](#)。

配置到现有 EFS 文件系统的复制

Amazon EFS 会将源文件系统上的数据和元数据复制到您选择 AWS 区域的目标文件系统。在复制过程中，Amazon EFS 会识别文件系统之间的数据差异，并将差异应用于目标文件系统。

要复制到现有文件系统，请执行以下步骤。

主题

- [步骤 1：禁用文件系统的复制覆盖保护功能](#)
- [步骤 2：创建复制配置](#)

Note

一个文件系统只能属于一个复制配置。不能在其他复制配置中将目标文件系统用作源文件系统。

步骤 1：禁用文件系统的复制覆盖保护功能

创建 Amazon EFS 文件系统时，将默认启用其复制覆盖保护功能。复制覆盖保护功能可防止在复制配置中将文件系统用作目标。在复制配置中使用文件系统作为目标之前，必须先禁用保护功能。如果您删除复制配置，则文件系统的复制覆盖保护功能将重新启用，文件系统将变为可写状态。

Amazon EFS 文件系统的复制覆盖保护功能的状态可以具有下表中描述的值之一。

文件系统状态	说明
已启用	文件系统不能用作复制配置中的目标文件系统。文件系统是可写的。默认情况下，复制覆盖保护功能处于 ENABLED 状态。
DISABLED	文件系统可以用作复制配置中的目标文件系统。

文件系统状态	说明
复制	文件系统正用作复制配置中的目标文件系统。文件系统是只读的，只有 Amazon EFS 在复制期间才能对其进行修改。

所需的权限

禁用复制覆盖保护功能需要 `elasticfilesystem:UpdateFileSystemProtection` 操作的权限。有关更多信息，请参阅 [AWS 托管策略：AmazonElasticFileSystemFullAccess](#)。

使用控制台

1. 登录 AWS 管理控制台 并打开 Amazon EFS 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 在左侧导航窗格中，选择文件系统。
3. 在文件系统列表中，选择要在复制配置中用作目标文件系统的 Amazon EFS 文件系统。
4. 在文件系统保护部分中，关闭复制覆盖保护。

禁用复制覆盖保护功能 (AWS CLI)

在以下示例中，`update-file-system-protection` CLI 命令禁用了对指定文件系统的复制覆盖保护。等效的 API 命令是 [UpdateFileSystemProtection](#)。

```
aws efs update-file-system-protection
--file-system-id fs-0a8b2be428114d97c
--replication-overwrite-protection DISABLED
```

AWS CLI 答案如下。

```
{
  "ReplicationOverwriteProtection": "DISABLED"
}
```

步骤 2：创建复制配置

在目标文件系统中禁用复制覆盖保护功能后，可以创建复制配置。复制到现有文件系统时，目标文件系统可以与源文件系统使用相同的账户或不同的账户。

在为 Amazon EFS 创建复制配置之前，请查看以下重要要求和注意事项：

- 如果源文件系统已加密，也必须对目标文件系统进行加密。此外，如果源文件未加密而目标文件系统已加密，则在执行失效转移后无法失效自动恢复到源目标。有关加密的更多信息，请参阅[Amazon EFS 中的数据加密](#)。
- 当您最初为现有文件系统配置复制时，Amazon EFS 会在目标文件系统中写入数据或删除现有数据，以与源文件系统中的数据相匹配。如果您不想更改目标文件系统中的数据，则应改为复制到新的文件系统。有关更多信息，请参阅[配置到新 EFS 文件系统的复制](#)。
- 只有在初始同步完成后，才能访问复制到目标文件系统的文件。同步所用时长取决于源文件系统的大小和其中所含文件数等因素。有关复制性能的更多信息，请参阅[复制性能](#)。

先决条件

拥有要使用的目标文件系统 ID（用于同账户复制）或目标文件系统 ARN（用于跨账户复制）的副本。

如果目标文件系统与源文件系统 AWS 账户不同，请创建一个 IAM 角色以允许 Amazon EFS 执行复制并为文件系统分配资源策略。有关更多信息，请参阅[跨 AWS 账户复制 EFS 文件系统](#)。

使用控制台

1. 登录 AWS 管理控制台 并打开 Amazon EFS 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 打开要复制的文件系统：
 - a. 在左侧导航窗格中，选择文件系统。
 - b. 在文件系统列表中，选择要复制的 Amazon EFS 文件系统。选择的文件系统不能是现有复制配置中的源文件系统或目标文件系统。
3. 选择复制选项卡。
4. 在复制部分，选择创建复制。
5. 对于复制配置，请选择现有文件系统。
6. 选择目标文件系统。
 - 要复制到与源文件系统 AWS 账户 相同的文件系统，请执行以下操作：
 1. 选择“选择此帐户中的文件系统”，然后在“目标”中 AWS 区域，选择 AWS 区域 要将文件系统复制到哪个。
 2. 选择浏览 EFS，然后选择文件系统。目标文件系统的路径显示在目标框中。

- 要复制到与源文件系统 AWS 账户 不同的文件系统，请执行以下操作：
 1. 选择在另一个账户中指定文件系统。
 2. 对于目标文件系统 ARN，输入目标文件系统的 Amazon 资源名称 (ARN)。

Note

如果文件系统上启用了复制覆盖保护功能，则会显示一条警告。选择禁用保护，在新选项卡中打开文件系统并关闭其复制覆盖保护功能。禁用保护功能后，请返回创建复制选项卡，并单击刷新按钮以清除消息。

7. 对于 IAM 角色，请输入支持 Amazon EFS 复制到目标文件系统的 IAM 角色的 ARN。对于同账户复制，此为可选项；但对于跨账户复制，此为必填项。有关更多信息，请参阅 [跨 AWS 账户复制 EFS 文件系统](#)。
8. 选择创建复制，在确认消息输入框中键入 confirm，然后选择创建复制。复制部分将显示复制详细信息。

创建复制配置 (AWS CLI)

本节提供了在 AWS CLI 中使用 `create-replication-configuration` 命令创建复制配置的示例。等效的 API 命令是 [CreateReplicationConfiguration](#)。

Example为另一区域中的现有目标文件系统创建复制配置

以下示例创建了一个复制配置，其中文件系统 ID `fs-0123456789abcdef1` 被复制到 `fs-0a8b2be428114d97c` 中的文件系统 ID `eu-west-2` AWS 区域。

```
aws efs create-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1 \  
--destinations "[{\"Region\":\"eu-west-2\",\"FileSystemId\":\"fs-0a8b2be428114d97c\"}]"
```

AWS CLI 答案如下：

```
{  
  "SourceFileSystemId": "fs-0123456789abcdef1",  
  "SourceFileSystemRegion": "us-east-1",  
  "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-  
system/fs-0123456789abcdef1",  
}
```

```

    "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-
east-1:111122223333:file-system/fs-0123456789abcdef1",
    "CreationTime": "2024-10-20T20:40:13+00:00",
    "Destinations": [
      {
        "Status": "ENABLING",
        "FileSystemId": "fs-0a8b2be428114d97c",
        "Region": "eu-west-2",
        "OwnerId": "123456789012",
      }
    ],
    "SourceFileSystemOwnerId": "123456789012"
  }
}

```

Example 创建跨账户复制配置

以下示例创建了一个复制配置，其中源文件系统和目标文件系统位于不同的位置 AWS 账户。账户 *fs-0123456789abcdef1* 中的源文件系统 ID 将复制到账户 *555666777888fs-0a8b2be428114d97c* 中的文件系统 ID *123456789012*。此示例指定了目标文件系统的 Amazon 资源名称 (ARN) 和源账户中支持 Amazon EFS 代表其执行复制的 IAM 角色的 ARN。由于未指定 KMS 密钥，因此使用账户的默认 AWS KMS 服务密钥 (aws/elasticfilesystem) 对目标文件系统进行加密。

```

aws efs
--region $REGION
--endpoint $ENDPOINT create-replication-configuration
--source-file-system-id fs-0123456789abcdef1
--destinations Region=eu-west-2,FileSystemId=arn:aws:elasticfilesystem:eu-
west-2:123456789012:file-system/
fs-0a8b2be428114d97c,RoleArn=arn:aws:iam::555666777888:role/cross-account-replication

```

AWS CLI 答案如下：

```

{
  "SourceFileSystemId": "fs-0123456789abcdef1",
  "SourceFileSystemRegion": "us-east-1",
  "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:555666777888:file-
system/fs-0123456789abcdef1",
  "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-
east-1:555666777888:file-system/fs-0123456789abcdef1",
  "CreationTime": "2024-10-20T20:40:13+00:00",
  "Destinations": [

```

```
{
  "Status": "ENABLING",
  "FileSystemId": "fs-0a8b2be428114d97c",
  "Region": "eu-west-2",
  "OwnerId": "123456789012",
  "RoleArn": "arn:aws:iam::555666777888:role/cross-account-replication"
},
"SourceFileSystemOwnerId": "555666777888"
}
```

跨 AWS 账户复制 EFS 文件系统

您可以跨 AWS 账户复制 EFS 文件系统。跨账户复制可增强灾难恢复 (DR) 策略的整体弹性和可靠性，并且可以帮助您满足公司的合规要求。

例如，合规策略可能要求您在不同的环境 [例如生产、暂存和灾难恢复 (DR)] 中使用不同的账户。或者，您可能会发现，跨不同的复制可以 AWS 账户 提供更强的隔离、对权限和访问策略的更精细的控制，以及更直接的资源审计。如果生产账户遭到入侵（例如出现安全漏洞、配置错误或内部威胁），则将灾难恢复服务器放在单独的账户中可以防止攻击者访问它们，缩小安全事件的爆炸半径，并大幅降低发生未经授权更改的风险。

跨复制 AWS 账户 需要额外的安全设置和策略设置。您必须创建使 Amazon EFS 有权在目标账户中执行复制操作的 IAM 角色，而不是使用服务相关角色来执行跨账户复制。您还需要在要跨账户共享的文件系统上创建策略。创建 IAM 角色和文件系统策略后，可以创建复制配置。

主题

- [创建具有自定义信任策略的 IAM 角色](#)
- [在源文件系统和目标文件系统上创建策略](#)
- [创建复制配置](#)

创建具有自定义信任策略的 IAM 角色

为了让 Amazon EFS 能够代表源账户执行跨账户复制，必须在源账户上创建 IAM 角色。该角色必须拥有 `elasticfilesystem.amazonaws.com` 信任策略，以支持 Amazon EFS 代入该角色并充当服务主体。该角色必须包含执行复制所需的全部 IAM 权限（请参阅[所需的 IAM 权限](#)），并授予向目标账户中文件系统复制的显式权限。

先决条件

您必须在复制配置中同时创建源文件系统和目标文件系统，然后才能为源账户创建 IAM 角色。在复制过程中，Amazon EFS 无法为您创建目标文件系统。此外，您还必须知道并提供每个文件系统的 Amazon 资源名称 (ARN)。

创建 IAM 角色来进行跨账户复制

以下是使用自定义信任策略创建用于执行 Amazon EFS 跨账户复制的 IAM 角色的一般步骤。有关创建 IAM 角色的 step-by-step 说明，请参阅 AWS Identity and Access Management 用户指南中的 [使用自定义信任策略创建角色](#)。

1. 在源账户的 AWS Identity and Access Management 控制台中，创建使用以下信任策略的 IAM 角色。相关说明，请参阅《AWS Identity and Access Management 用户指南》中的 [使用自定义信任策略创建角色](#)。

JSON

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

2. 创建该角色后，为该角色分配以下权限。将 `arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1` 替换为目标文件系统的 ARN，并将 `arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1` 替换为源文件系统的 ARN。有关为角色分配权限的说明，请参阅 [使用 JSON 编辑器创建策略](#)。

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "elasticfilesystem:DescribeFileSystems",
      "elasticfilesystem:CreateReplicationConfiguration",
      "elasticfilesystem:DescribeReplicationConfigurations",
      "elasticfilesystem>DeleteReplicationConfiguration",
      "elasticfilesystem:ReplicationWrite"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-  
east-1:111122223333:file-system/fs-0123456789abcdef1"
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticfilesystem:ReplicationRead",
      "elasticfilesystem:DescribeFileSystems"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-  
east-1:444455556666:file-system/fs-5678910112hijkqr1"
  }
]
```

3. 复制或输入 IAM 角色的 ARN。创建复制配置时，您需要提供此 ARN。

在源文件系统和目标文件系统上创建策略

要在 Amazon EFS 中跨账户共享文件系统，必须为目标文件系统和源文件系统分配策略。这些策略授予或限制跨账户访问其应用到的文件系统。只有有权编辑文件系统的账户所有者才能为其账户中的文件系统分配策略。

除了授予或限制跨账户访问权限外，这些策略还需要授予客户端使用文件系统所需的其他权限，例如 `elasticfilesystem:ClientMount`。否则，客户端可能无法访问文件系统。

Important

您无法限制通过 TLS 连接进行的资源访问。如果您在语句中包含 `"aws:SecureTransport": "false"` 条件，NFS 客户端连接将失败。

用于目标文件系统的策略

要授予源账户向目标文件系统执行复制操作的权限，以及从目标账户删除复制配置的权限，必须在目标文件系统中创建以下策略。将 `arn:aws:iam::444455556666:root` 替换为拥有源文件系统的账户的 ID。将 `arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1` 替换为目标文件系统的 ARN

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSourceAccountReplicationActions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
      },
      "Action": [
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:CreateReplicationConfiguration",
        "elasticfilesystem:DescribeReplicationConfigurations",
        "elasticfilesystem>DeleteReplicationConfiguration",
        "elasticfilesystem:ReplicationWrite"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1"
    },
    {
      "Sid": "AllowReadOnlyClientAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
      },
      "Action": [
        "elasticfilesystem:ClientMount"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1"
    }
  ]
}
```

用于源文件系统的策略

要授予目标账户从源账户中删除复制配置的权限，必须将以下策略分配给源文件系统。

将 `arn:aws:iam::111122223333:root` 替换为拥有目标文件系统的账户的 ID。

将 `arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1` 替换为源文件系统的 ARN。

JSON

```
{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
  "Statement": [
    {
      "Sid": "AllowDestinationAccountToDeleteReplication",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "elasticfilesystem:DeleteReplicationConfiguration",
      "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1"
    },
    {
      "Sid": "AllowClientAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
      },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientRootAccess"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1",
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    }
  ]
}
```

```
]
}
```

创建文件系统策略

使用上一节中创建的策略，对目标文件系统和源文件系统执行以下步骤。

1. AWS 管理控制台 使用拥有文件系统的账户登录，然后在 Amazon EFS 控制台上打开 [Amazon EFS 控制台](#)。
2. 打开文件系统：
 - a. 在左侧导航窗格中，选择文件系统。
 - b. 在文件系统列表中，选择所需文件系统。
3. 在文件系统策略 选项卡中，选择编辑。
4. 将策略粘贴到策略编辑器 {Json} 中，然后选择保存。

创建复制配置

创建 IAM 角色并将文件系统策略添加到源文件系统和目标文件系统后，请按照[配置到现有 EFS 文件系统的复制](#)中的说明创建复制配置。

查看复制详细信息

可以监控复制配置中上次成功完成同步的时间。在此时间之前对源文件系统上的数据所做的任何更改都已成功复制到目标文件系统。在此时间之后发生的任何更改都可能无法完全复制。要监控上次复制的成功完成时间，您可以使用 Amazon EFS 控制台 AWS CLI、API 或亚马逊 CloudWatch。

- 在 EFS 控制台中 – 文件系统详细信息 > 复制部分中的上次同步属性显示源和目标之间上次成功同步完成的时间。
- 在 AWS CLI 或 API 中 — Destination 对象中的 LastReplicatedTimestamp 属性显示上次成功完成同步的时间。要访问此属性，请使用 describe-replication-configurations CLI 命令。[DescribeReplicationConfigurations](#) 是等效的 API 操作。
- 在 CloudWatch — Amazon EFS 的 TimeSinceLastSync CloudWatch 指标显示自上次成功完成同步以来经过的时间。有关更多信息，请参阅 [CloudWatch 亚马逊 EFS 的指标](#)。

复制配置可以具有下表中所述的状态值。

复制状态	说明
ENABLED	复制配置处于正常状态，可供使用。
ENABLING	Amazon EFS 正在创建复制配置。
DELETING	Amazon EFS 正在删除复制配置，以响应用户发起的删除请求。
PAUSING	Amazon EFS 正在暂停复制。
PAUSED	由于配置问题，复制已暂停。提供了有关问题的其他信息。 导致复制暂停的一些问题包括： - 授权错误。权限问题阻止 Amazon EFS 执行复制。确保用于创建复制配置的 IAM 角色具有执行复制的权限。此外，还要确保文件系统策略正确无误。 - AWS 账户不可用。确保源账户和目标账户被选入相应的账户，AWS 区域并且没有被暂停。有关更多信息，请参阅《AWS 通用参考指南》中的指定 AWS 区域 您的账户可以使用。 - 源文件系统或目标文件系统的 KMS 密钥不可访问。确保分配给每个文件系统的 KMS 密钥均可访问。有关更多信息，请参阅使用 AWS KMS 亚马逊 EFS 的密钥。
ERROR	复制配置处于故障状态且无法恢复。必须删除现有的复制配置，然后创建一个新的复制配置。 提供了有关问题的其他信息。对于跨账户或跨区域复制，错误可能是因为复制配置已从另一个 AWS 账户 或中删除。AWS 区域

使用控制台

1. 打开 Amazon Elastic File System 控制台，网址为<https://console.aws.amazon.com/efs/>。
2. 在左侧导航窗格中，选择文件系统。
3. 从列表中选择文件系统。
4. 选择复制选项卡以显示复制部分。

在复制部分中，可以看到以下复制配置信息：

- 复制状态可能为正在启用、已启用、正在删除、正在暂停、已暂停或错误。Amazon EFS 会显示有关已暂停或错误状态的详细原因。
- 复制方向显示复制数据的方向。列出的第一个文件系统是源文件系统，正在将其数据复制到列出的第二个文件系统，即目标文件系统。
- 上次同步显示目标文件系统上次成功同步的时间。在此时间之前对源文件系统上的数据所做的任何更改都已成功复制到目标文件系统。在此时间之后发生的任何更改都可能无法完全复制。
- 复制文件系统按其文件系统 ID、其在复制配置中的角色（源或目标）、所在位置及其权限列出复制配置 AWS 区域 中的每个文件系统。源文件系统具有可写权限，目标文件系统具有只读权限。

使用 AWS CLI

要查看复制配置，请使用 `describe-replication-configurations` 命令。您可以查看特定文件系统的复制配置，也可以查看 AWS 账户 中特定文件系统的所有复制配置 AWS 区域。等效的 API 命令是 [DescribeReplicationConfigurations](#)。

如果复制配置的状态为 `PAUSED` 或 `ERROR`，则 `StatusMessage` 参数将返回有关问题原因以及如何修复问题的信息。

Example：查看特定文件系统的复制配置

以下示例描述了文件系统 `fs-0123456789abcdef1` 的复制配置。

```
aws efs describe-replication-configurations --file-system-id fs-0123456789abcdef1
```

AWS CLI 答案如下。

```
{
  "Replications": [
    {
      "SourceFileSystemArn": "arn:aws:elasticfilesystem:eu-west-1:111122223333:file-system/fs-abcdef0123456789a",
      "CreationTime": 1641491892.0,
      "SourceFileSystemRegion": "eu-west-1",
      "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:eu-west-1:111122223333:file-system/fs-abcdef0123456789a",
      "SourceFileSystemId": "fs-abcdef0123456789a",
      "Destinations": [
        {
```

```

        "Status": "ENABLED",
        "FileSystemId": "fs-0123456789abcdef1",
        "Region": "us-east-1"
      }
    ]
  }
}

```

Example：查看某个账户中的所有复制配置

以下示例描述了文件系统中账户的所有复制配置 AWS 区域 的复制配置。

```
aws efs describe-replication-configurations
```

AWS CLI 答案如下。

```

{
  "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
      }
    ]
  }
}
```

删除复制配置

如果需要故障转移到目标文件系统，请删除其所属的复制配置。删除复制配置后，目标文件系统变为可写状态，其复制覆盖保护功能将重新启用。有关更多信息，请参阅 [使用副本](#)。

删除复制配置并将目标文件系统更改为可写，可能需要几分钟才能完成。删除配置后，Amazon EFS 可能会使用以下命名约定将一些数据写入目标文件系统根目录中的 `lost+found` 目录中：

```
efs-replication-lost+found-source-file-system-id-TIMESTAMP
```

Note

无法删除属于复制配置的文件系统。在删除文件系统之前，必须先删除复制配置。

您可以使用 Amazon EFS 控制台、或 API 从源文件系统或目标文件系统中 AWS CLI 删除现有复制配置。

对于跨账户或跨区域复制，Amazon EFS 会从源账户和目标账户或者源区域和目标区域中删除复制配置。如果存在配置或权限问题，导致 Amazon EFS 无法同时从两端删除复制配置，则只能从本地端（执行删除操作的账户或区域）删除配置。删除本地配置会使其他账户或区域中的配置无法恢复。

使用控制台

1. 打开 Amazon Elastic File System 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 在左侧导航窗格中，选择文件系统。
3. 选择复制配置中要删除的源文件系统或目标文件系统。
4. 选择复制选项卡以显示复制部分。
5. 选择删除复制以删除复制配置。提示时，确认选择。

如果您要删除跨账户复制配置，并且存在无法同时从源端和目标端删除配置的问题，则可以选择仅删除此文件系统的配置。

Note

仅当 Amazon EFS 无法同时删除源账户和目标账户或者源区域和目标区域中的复制配置时，才删除文件系统的配置。删除本地配置会使其他账户或区域中的配置无法恢复。

使用 AWS CLI

要删除复制配置，请使用 `delete-replication-configuration` CLI。等效的 API 命令是 [DeleteReplicationConfiguration](#)。

以下示例删除了源文件系统 `fs-0123456789abcdef1` 的复制配置。

```
aws efs --region us-west-2 delete-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1
```

如果存在配置或权限问题，导致 Amazon EFS 无法同时从两端删除复制配置，则只能从本地端（执行删除操作的账户或区域）删除配置。删除本地配置会使其他账户或区域中的配置无法恢复。等效的 API 参数为 `DeletionMode`，值为 `LOCAL_CONFIGURATION_ONLY`。

以下示例仅 `fs-0123456789abcdef1` 从本地删除源文件系统的复制配置。

```
aws efs --region us-west-2 delete-replication-configuration \  
--source-file-system-id fs-0123456789abcdef1 \  
--deletion-mode LOCAL_CONFIGURATION_ONLY
```

使用副本

在发生灾难或进行游戏日练习时，您可以通过删除副本文件系统的复制配置来失效转移至副本文件系统。删除复制配置后，副本将变为可写状态，您可以开始在应用程序工作流程中使用它。灾难缓解或游戏日练习结束后，您可以继续使用副本作为主文件系统，也可以执行失效自动恢复以恢复对原始主文件系统的操作。

在失效自动恢复过程中，您可以选择放弃对副本文件系统所做的更改，也可以通过将其复制回主文件系统来保留这些更改。

- 要丢弃在失效转移期间对副本所做的更改，请在主文件系统上重新创建原始复制配置，其中副本文件系统是复制目标。在复制过程中，Amazon EFS 通过更新副本文件系统的数据以与主文件系统的数据相匹配，从而同步文件系统。
- 要复制在失效转移期间对副本所做的更改，请在副本文件系统上创建复制配置，其中主文件系统是复制目标。在复制过程中，Amazon EFS 会识别副本文件系统的差异并将其传输回主文件系统。复制完成后，您可以通过重新创建原始复制配置或创建新配置来继续复制主文件系统。

Amazon EFS 完成复制过程所需的时间量各不相同，具体取决于文件系统的大小和其中的文件数量等因素。有关更多信息，请参阅 [复制性能](#)。

在 Amazon EFS 中保护数据

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。[责任共担模式](#)将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。AWS 还为您提供可以安全使用的服务。作为[AWS 合规计划](#)的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 Amazon Elastic File System 的合规计划，请参阅[按合规计划 AWS 提供的范围内的 AWS 服务](#)服务。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

此文档将帮助您了解如何在使用 Amazon EFS 时应用责任共担模式。以下主题说明如何配置 Amazon EFS 以实现安全性和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 Amazon EFS 资源。

主题

- [Amazon EFS 中的数据保护](#)
- [Amazon EFS 的身份和访问管理](#)
- [使用 IAM 控制文件系统访问](#)
- [Amazon EFS 的合规性验证](#)
- [Amazon EFS 中的韧性](#)
- [控制 NFS 客户端对 EFS 文件系统的网络访问](#)
- [网络文件系统 \(NFS \) 级别用户、组和权限](#)
- [使用接入点工作](#)
- [屏蔽对 EFS 文件系统的公共访问权限](#)
- [Amazon EFS 网络隔离](#)

Amazon EFS 中的数据保护

[责任 AWS 共担模式](#)适用于 Amazon EFS 中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使

用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题解答 AWS](#)条款。有关欧洲数据保护的信息，请参阅[通用数据保护条例 \(GDPR\) 中心](#)。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的[使用跟 CloudTrail 踪](#)。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 (FIPS) 第 140-3 版》<https://aws.amazon.com/compliance/fips/>。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API 或软件开发工具包 AWS 服务 使用 Amazon EFS 或其他 AWS 软件开发工具包的情况。AWS CLI 在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

主题

- [Amazon EFS 中的数据加密](#)
- [互联网络隐私](#)

Amazon EFS 中的数据加密

Amazon EFS 提供全面的加密功能来保护静态数据和传输中数据。

- 静态加密 - 加密存储在文件系统中的数据。
- 传输中加密 - 加密在客户端和文件系统之间传输的数据。

如果您组织的公司或监管策略要求加密数据和元数据，建议您创建一个静态加密的文件系统，并通过加密传输中数据来挂载文件系统。

主题

- [加密静态数据](#)
- [加密传输中数据](#)
- [使用 AWS KMS 亚马逊 EFS 的密钥](#)
- [排除加密故障](#)

加密静态数据

静态加密可对 EFS 文件系统中存储的数据进行加密。这可以帮助您满足合规性要求并确保敏感数据免遭未经授权的访问。您的组织可能要求加密符合特定分类条件的所有数据，或者加密与特定应用程序、工作负载或环境关联的所有数据。

Note

AWS 密钥管理基础设施使用经联邦信息处理标准 (FIPS) 140-3 批准的加密算法。该基础设施符合美国国家标准与技术研究院 (NIST) 800-57 建议。

使用 Amazon EFS 控制台创建文件系统时，静态加密默认处于启用状态。使用 AWS CLI、API 或 SDK 创建文件系统时，必须明确启用加密。

创建 EFS 文件系统后，将无法更改其加密设置。这意味着您无法修改未加密的文件系统以使其加密。相反，您需要通过[复制文件系统](#)，将数据从未加密的文件系统复制到新的加密文件系统。有关更多信息，请参阅[如何为现有 EFS 文件系统启用静态加密？](#)

静态加密的工作方式

在加密文件系统中，默认情况下，数据和元数据在写入存储之前会进行加密，并在读取时自动解密。这些过程是 Amazon EFS 以透明方式处理的，因此，您无需修改您的应用程序。

Amazon EFS AWS KMS 用于密钥管理，如下所示：

- 文件数据加密 - 使用您指定的 KMS 密钥对文件内容进行加密。您可以指定：
 - AWS 拥有的密钥 适用于 Amazon EFS (aws/elasticfilesystem) 的“默认”选项，不收取额外费用。

- 您创建和管理的客户托管密钥 - 提供额外的控制和审计功能。
- 元数据加密 - 文件名、目录名和目录内容是使用 Amazon EFS 在内部管理的密钥加密的。

加密过程

在创建文件系统或将某个文件系统复制到同一账户下的另一个文件系统时，Amazon EFS 使用[转发访问会话 \(FAS\)](#)，通过调用者凭证进行 KMS 调用。在 CloudTrail 日志中，kms:CreateGrant 调用似乎是由创建文件系统或复制的相同用户身份进行的。您可以 CloudTrail 通过查找带有值的 invokedBy 字段来识别 Amazon EFS 服务调用 elasticfilesystem.amazonaws.com。KMS 密钥上的资源策略必须支持 FAS 执行 CreateGrant 操作以发起调用。

Important

您可以管理对授权的控制权，并且可以随时撤销授权。撤销授权后，Amazon EFS 将无法在后续操作中访问该 KMS 密钥。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的[停用和撤销授权](#)。

使用客户托管的 KMS 密钥时，资源策略还必须支持 Amazon EFS 服务主体，并包括 kms:ViaService 条件来限制访问特定服务端点。例如：

```
"kms:ViaService":  
  "elasticfilesystem.us-east-2.amazonaws.com"
```

Amazon EFS 使用行业标准 AES-256 加密算法对静态数据和元数据进行加密。

有关适用于 Amazon EFS 的 KMS 密钥策略的更多信息，请参阅[使用 AWS KMS 亚马逊 EFS 的密钥](#)。

对新文件系统强制执行静态加密

您可以在基于 AWS Identity and Access Management (IAM) 身份的策略中使用 elasticfilesystem:Encrypted IAM 条件键，来在用户创建 EFS 文件系统时强制执行静态创建。有关使用此条件键的更多信息，请参阅[示例：强制创建加密文件系统](#)。

您还可以在内部定义服务控制策略 (SCP)，AWS Organizations 对组织 AWS 账户 中的所有人强制执行 Amazon EFS 加密。有关服务控制策略的更多信息 AWS Organizations，请参阅《AWS Organizations 用户指南》中的[服务控制策略](#)。

加密传输中数据

Amazon EFS 支持使用传输层安全性协议 (TLS) 对传输中数据进行加密。在将传输中数据加密声明为 EFS 文件系统的挂载选项时，Amazon EFS 会在挂载文件系统时与您的 EFS 文件系统建立安全的 TLS 连接。所有 NFS 流量都通过此加密连接进行路由。

传输中加密的工作方式

我们建议使用 EFS 挂载帮助程序来挂载您的文件系统，因为与使用 NFS mount 挂载相比，它简化了挂载过程。EFS 挂载帮助程序通过使用 `efs-proxy` (适用于 `efs-utils` 版本 2.0.0 及更高版本) 或 `stunnel` (适用于 `efs-utils` 的早期版本) 来管理该进程，从而与 EFS 文件系统建立安全的 TLS 连接。

如果未使用挂载帮助程序，您仍然可以启用传输中的数据加密。请按以下步骤操作。

在不使用挂载帮助程序的情况下对传输中数据进行加密

1. 下载并安装 `stunnel`，并记下该应用程序侦听的端口。有关更多信息，请参阅 [升级 stunnel](#)。
2. 运行 `stunnel` 以使用 TLS 通过端口 2049 连接到您的 EFS 文件系统。
3. 使用 NFS 客户端挂载 `localhost:port`，其中 `port` 是在第一步中记下的端口。

由于传输中的数据加密是根据每个连接配置的，因此，每个配置的挂载在实例上运行专用的 `stunnel` 进程。默认情况下，挂载帮助程序使用的 `stunnel` 进程会侦听 20049 和 20449 之间的本地端口，并通过端口 2049 连接到 Amazon EFS。

Note

默认情况下，在将 EFS 挂载帮助程序与 TLS 一起使用时，它会强制使用在线证书状态协议 (OCSP) 和证书主机名检查。EFS 挂载帮助程序使用 `stunnel` 程序提供 TLS 功能。某些版本的 Linux 不包含默认支持这些 TLS 功能的 `stunnel` 版本。在使用这些 Linux 版本之一时，使用 TLS 挂载 EFS 文件系统将失败。

在安装 `amazon-efs-utils` 软件包后，要升级您的系统的 `stunnel` 版本，请参阅[升级 stunnel](#)。有关加密问题，请参阅[排除加密故障](#)。

在使用传输中的数据加密时，将更改您的 NFS 客户端设置。在检查您主动挂载的文件系统时，将会看到一个文件系统挂载到 `127.0.0.1` 或 `localhost`，如以下示例中所示。

```
$ mount | column -t
```



```
127.0.0.1:/ on /home/ec2-user/efs type nfs4
(rw,relatime,vers=4.1,rsz=1048576,wsz=1048576,namlen=255,hard,proto=tcp,port=20127,timeo=6
```

在使用 TLS 和 EFS 挂载帮助程序进行挂载时，将重新配置 NFS 客户端以挂载到本地端口。EFS 挂载帮助程序启动一个客户端 `stunnel` 进程以侦听该本地端口，并且 `stunnel` 使用 TLS 打开到 EFS 文件系统的加密连接。EFS 挂载帮助程序负责设置和维护该加密连接和关联的配置。

要确定哪个 Amazon EFS 文件系统 ID 对应于哪个本地挂载点，您可以使用以下命令。请记住 `efs-mount-point` 替换为挂载文件系统的本地路径。

```
grep -E "Successfully mounted.*efs-mount-point" /var/log/amazon/efs/mount.log | tail -1
```

在将 EFS 挂载帮助程序用于传输中数据加密时，它还会创建一个名为 `amazon-efs-mount-watchdog` 的进程。该进程可确保每个挂载的 `stunnel` 进程正在运行，并在卸载 EFS 文件系统后停止 `stunnel`。如果 `stunnel` 进程由于某种原因意外终止，`watchdog` 进程将重新启动该进程。

使用 AWS KMS 亚马逊 EFS 的密钥

Amazon EFS 与 AWS Key Management Service (AWS KMS) 集成，用于密钥管理。Amazon EFS 使用客户托管密钥通过以下方法加密您的文件系统：

- 静态加密元数据 – Amazon EFS 使用适用于 Amazon EF 的 AWS 托管式密钥 `aws/elasticfilesystem` 来加密和解密文件系统元数据（即，文件名、目录名称和目录内容）。
- 静态加密文件数据 – 您选择用于加密和解密文件数据（即，文件内容）的客户托管文件。您可以启用、禁用或撤销对此客户托管密钥的授权。此客户托管密钥可以是以下两种类型之一：
 - AWS 托管式密钥 适用于 Amazon EFS — 这是默认的客户托管密钥 `aws/elasticfilesystem`。您无需为创建和存储客户托管密钥支付费用，但需要支付使用费用。要了解更多信息，请参阅 [AWS Key Management Service 定价](#)。
 - 客户托管式密钥 – 这是使用最灵活的 KMS 密钥，因为您可以配置其密钥策略以及为多个用户或服务提供授权。有关创建客户托管密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 [创建密钥](#)。

如果将客户托管式密钥用于数据加密和解密，您可以启用密钥轮换。启用密钥轮换后，每年 AWS KMS 自动轮换密钥一次。此外，对于客户托管式密钥，您还可以随时选择何时禁用、重新启用、删除或撤销对您的客管理式密钥的访问权限。有关更多信息，请参阅 [使用 AWS KMS 亚马逊 EFS 的密钥](#)。

 Important

Amazon EFS 仅接受对称的客户托管式密钥。您不能在 Amazon EFS 中使用非对称的客户托管式密钥。

静态数据加密和解密是透明处理的。但是，特定于 Amazon EFS 的 AWS 账户 ID 会出现在与 AWS KMS 操作相关的 AWS CloudTrail 日志中。有关更多信息，请参阅 [文件系统的 Amazon EFS 日志 encrypted-at-rest 文件条目](#)。

Amazon EFS 的密钥政策 AWS KMS

密钥策略是控制对客户托管式密钥的访问的主要方式。有关密钥策略的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 [AWS KMS 中的密钥策略](#)。以下列表描述了 Amazon EFS 对静态加密文件系统所需或以其他方式支持的所有 AWS KMS 相关权限：

- kms:Encrypt – (可选) 将明文加密为密文。该权限包含在默认密钥策略中。
- kms:Decrypt – (必需) 解密密文。密文是以前加密的明文。该权限包含在默认密钥策略中。
- kms: ReEncrypt — (可选) 使用新的客户托管密钥加密服务器端的数据，而不会在客户端暴露数据的明文。将先解密数据，然后重新加密。该权限包含在默认密钥策略中。
- kms: GenerateDataKeyWithoutPlaintext — (必填) 返回使用客户托管密钥加密的数据加密密钥。此权限包含在 kms: GenerateDataKey * 下的默认密钥策略中。
- kms: CreateGrant s: — (必填) 向密钥添加授权，以指定谁可以在什么条件下使用该密钥。授权是密钥政策的替代权限机制。有关授权的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 [AWS KMS 中的授权](#)。该权限包含在默认密钥策略中。
- kms: DescribeKey — (必填) 提供有关指定客户托管密钥的详细信息。该权限包含在默认密钥策略中。
- kms: ListAliases s: — (可选) 列出账户中的所有密钥别名。使用控制台创建加密的文件系统时，该权限将填充选择 KMS 密钥列表。我们建议您使用该权限以提供最佳的用户体验。该权限包含在默认密钥策略中。

AWS 托管式密钥 适用于 Amazon EFS KMS 政策

适用于 Amazon EFS AWS 托管式密钥 的 KMS 策略 JS aws/elasticfilesystem ON 如下所示：

```
{
  "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": "*"
      }
    ]
  }
}

```

密钥状态及其影响

KMS 密钥的状态直接影响到对加密文件系统的访问：

已启用

正常操作 - 对文件系统具有完全读取和写入权限

已禁用

文件系统在短时间为无法访问。可以重新启用。

待删除

文件系统变得无法访问。在等待期间可以取消删除。

已删除

文件系统永久无法访问。此操作无法撤消。

Warning

如果您禁用或删除用于文件系统的 KMS 密钥，或者撤销 Amazon EFS 对该密钥的访问权限，您的文件系统将变得无法访问。如果您没有备份，这可能会导致数据丢失。在更改加密密钥之前，请务必确保已创建正确的备份过程。

排除加密故障

您可以在下文中找到有关解决 Amazon EFS 加密问题的信息。

- [具有传输中数据加密的挂载失败](#)
- [具有传输中数据加密的挂载中断](#)
- [Encrypted-at-rest 无法创建文件系统](#)
- [无法使用的加密文件系统](#)

具有传输中数据加密的挂载失败

默认情况下，当您使用带有传输层安全性协议 (TLS) 的 Amazon EFS 挂载帮助程序时，它会强制执行主机名检查。某些系统不支持此功能，例如使用 Red Hat Enterprise Linux 或 CentOS 时。在这些情况下，挂载使用 TLS 的 EFS 文件系统会失败。

要采取的操作

我们建议您升级客户端上的 stunnel 版本以支持主机名检查。有关更多信息，请参阅 [升级 stunnel](#)。

具有传输中数据加密的挂载中断

在极少数情况下，客户端事件可能会导致到您的 Amazon EFS 文件系统的加密连接挂起或中断。

要采取的操作

如果到使用传输中数据加密的 Amazon EFS 文件系统的连接中断，请执行以下步骤：

1. 确保正在客户端上运行 stunnel 服务。
2. 确认正在客户端上运行监控程序应用程序 amazon-efs-mount-watchdog。您可以使用以下命令确定是否正在运行该应用程序：

```
ps aux | grep [a]mazon-efs-mount-watchdog
```

3. 检查您的支持日志。有关更多信息，请参阅 [获取支持日志](#)。
4. (可选) 您可以启用 stunnel 日志以及检查这些日志中的信息。您可以在 `/etc/amazon/efs/efs-utils.conf` 中更改日志配置以启用 stunnel 日志。但是，这样做需要卸载文件系统，然后使用挂载帮助程序重新挂载以使更改生效。

Important

启用 stunnel 日志可能会用完您的文件系统上的宝贵空间量。

如果中断仍在继续，请联系 Su AWS pport。

Encrypted-at-rest 无法创建文件系统

您已尝试创建新的 encrypted-at-rest 文件系统。但是，您会收到一条错误消息，提示该消息 AWS KMS 不可用。

要采取的操作

在极少数情况下，可能会发生此错误，这种情况在您中暂时 AWS KMS 不可用 AWS 区域。如果发生这种情况，请等待，直到 AWS KMS 恢复到完全可用状态，然后重试创建文件系统。

无法使用的加密文件系统

加密的文件系统持续返回 NFS 服务器错误。当 EFS 由于以下原因之一无法从中 AWS KMS 检索您的主密钥时，可能会发生这些错误：

- 禁用了密钥。
- 删除了密钥。
- 撤销了 Amazon EFS 使用密钥的权限。
- AWS KMS 暂时不可用。

要采取的操作

首先，确认 AWS KMS 密钥已启用。为此，您可以在控制台中查看这些密钥。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的[查看密钥](#)。

如果未启用密钥，请将其启用。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的[启用和禁用密钥](#)。

如果密钥处于待删除状态，该状态将禁用密钥。您可以取消删除，然后重新启用密钥。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的[计划和取消密钥删除](#)。

如果密钥已启用，但仍然遇到问题，或者在重新启用密钥时遇到问题，请联系 Su AWS pport。

互网络隐私

本主题介绍 Amazon EFS 如何保护从服务到其他位置的连接。

服务与本地客户端和应用之间的流量

您的私有网络和以下两种连接方式可供选择 AWS：

- 一个 AWS Site-to-Site VPN 连接。有关更多信息，请参阅[什么是 AWS Site-to-Site VPN ?](#)
- 一个 Direct Connect 连接。有关更多信息，请参阅[什么是 Direct Connect ?](#)

通过网络访问 Amazon EFS 需要通过 AWS 已发布的 API。客户端必须支持传输层 1.2 或更高版本。我们建议使用 TLS 1.3 或更高版本。客户端还必须支持具有完全正向保密 (PFS) 的密码套件，例如短暂的 (DHE) 或椭圆曲线短暂的 Diffie-Hellman (ECDHE)。Diffie-Hellman 大多数现代系统 (如 Java 7

及更高版本) 都支持这些模式。此外，必须使用与 IAM 主体关联的访问密钥 ID 和秘密访问密钥来对请求进行签名，也可以使用 [AWS Security Token Service \(AWS STS \)](#) 生成临时安全凭证来对请求进行签名。

VPC 和 Amazon EFS API 之间的流量

要在虚拟私有云 (VPC) 与 Amazon EFS API 之间建立专用连接，可以创建接口 VPC 端点。您可以使用此连接从 VPC 调用 Amazon EFS API，而无需通过互联网发送流量。端点提供与 Amazon EFS API 的安全连接，无需互联网网关、NAT 实例或虚拟专用网络 (VPN) 连接。有关更多信息，请参阅 [使用 Amazon EFS 中的接口 VPC 端点](#)。

之间的交通 AWS 同一地区的资源

Amazon EFS 的 Amazon Virtual Private Cloud (Amazon VPC) 端点是 VPC 内的逻辑实体，仅支持连接到 Amazon EFS。Amazon VPC 会将请求路由到 Amazon EFS 并将响应路由回 VPC。有关更多信息，请参阅《Amazon VPC 用户指南》的 [VPC 端点](#)。

Amazon EFS 的身份和访问管理

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以通过身份验证 (登录) 和获得授权 (具有权限) 来使用 Amazon EFS 资源。您可以使用 IAM AWS 服务，无需支付额外费用。

主题

- [受众](#)
- [使用身份进行身份验证](#)
- [使用策略管理访问](#)
- [Amazon Elastic File System 如何与 IAM 配合使用](#)
- [适用于 Amazon Elastic File System 的基于身份的策略示例](#)
- [适用于 Amazon EFS 的基于资源的策略示例](#)
- [AWS 亚马逊 EFS 的托管策略](#)
- [将标签与 Amazon EFS 一起使用](#)
- [对 Amazon EFS 使用服务相关角色](#)
- [Amazon Elastic File System 身份和访问问题排查](#)

受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（请参阅[Amazon Elastic File System 身份和访问问题排查](#)）
- 服务管理员：确定用户访问权限并提交权限请求（请参阅[Amazon Elastic File System 如何与 IAM 配合使用](#)）
- IAM 管理员：编写用于管理访问权限的策略（请参阅[适用于 Amazon Elastic File System 的基于身份的策略示例](#)）

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的[如何登录您的 AWS 账户](#)。

对于编程访问，AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的[适用于 API 请求的 AWS 签名版本 4](#)。

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。AWS IAM Identity Center 有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[什么是 IAM Identity Center？](#)。

IAM 用户和群组

[IAM 用户](#)是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的[要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问](#)。

[IAM 组](#)指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的[IAM 用户使用案例](#)。

IAM 角色

[IAM 角色](#)是具有特定权限的身份，可提供临时凭证。您可以通过[从用户切换到 IAM 角色（控制台）](#)或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的[担任角色的方法](#)。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的[JSON 策略概述](#)。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管策略与内联策略之间进行选择](#)。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中[指定主体](#)。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的[服务控制策略](#)。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的[资源控制策略 \(RCPs\)](#)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的[会话策略](#)。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的[策略评估逻辑](#)。

Amazon Elastic File System 如何与 IAM 配合使用

在使用 IAM 管理对 Amazon EFS 的访问权限之前，应了解哪些 IAM 功能可用于 Amazon EFS。

可与 Amazon Elastic File System 配合使用的 IAM 功能

IAM 功能	Amazon EFS 支持
基于身份的策略	是

IAM 功能	Amazon EFS 支持
基于资源的策略	是
策略操作	是
策略资源	是
策略条件键 (特定于服务)	是
ACLs	否
ABAC (策略中的标签)	部分
临时凭证	是
主体权限	是
服务角色	是
服务关联角色	是

要全面了解 Amazon EFS 和其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中与 IAM [配合使用的AWS 服务](#)。

Amazon EFS 基于身份的策略

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素引用](#)。

Amazon EFS 基于身份的策略示例

要查看 Amazon EFS 基于身份的策略示例，请参阅[适用于 Amazon Elastic File System 的基于身份的策略示例](#)。

Amazon EFS 基于资源的策略

支持基于资源的策略：是

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

要了解如何使用资源策略控制文件系统数据访问，请参阅[使用 IAM 控制文件系统访问](#)。要了解如何将基于资源的策略附加到系统，请参阅[创建文件系统策略](#)。

Amazon EFS 中基于资源的策略示例

要查看 Amazon EFS 基于资源的策略示例，请参阅[适用于 Amazon EFS 的基于资源的策略示例](#)。

Amazon EFS 的策略操作

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

有关 Amazon EFS 操作列表，请参阅《服务授权参考》中的[Amazon Elastic File System 定义的操作](#)。

Amazon EFS 中的策略操作在操作前面使用以下前缀：

```
elasticfilesystem
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
```

```
"elasticfilesystem:action1",  
"elasticfilesystem:action2"  
]
```

要查看 Amazon EFS 基于身份的策略示例，请参阅[适用于 Amazon Elastic File System 的基于身份的策略示例](#)。

Amazon EFS 的策略资源

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*" 
```

要查看 Amazon EFS 资源类型及其列表 ARNs，请参阅《服务授权参考》中的 [Amazon Elastic File System 定义的资源](#)。要了解您可以使用哪些操作指定每个资源的 ARN，请参阅 [Amazon Elastic File System 定义的操作](#)。

要查看 Amazon EFS 基于身份的策略示例，请参阅[适用于 Amazon Elastic File System 的基于身份的策略示例](#)。

Amazon EFS 的策略条件键

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用[条件运算符](#)（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

要查看 Amazon EFS 条件键列表，请参阅《服务授权参考》中的 [Amazon Elastic File System 的条件键](#)。要了解您可以对哪些操作和资源使用条件键，请参阅 [Amazon Elastic File System 定义的操作](#)。

要查看 Amazon EFS 基于身份的策略示例，请参阅[适用于 Amazon Elastic File System 的基于身份的策略示例](#)。

ACLs 在 Amazon EFS 中

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人 (账户成员、用户或角色) 有权访问资源。ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

ABAC 与 Amazon EFS

支持 ABAC (策略中的标签)：部分支持

基于属性的访问权限控制 (ABAC) 是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的[条件元素](#)中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的[使用 ABAC 授权定义权限](#)。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的[使用基于属性的访问权限控制 \(ABAC \)](#)。

将临时凭证用于 Amazon EFS

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的临时安全凭证](#)和[使用 IAM 的 AWS 服务](#)

Amazon EFS 的跨服务主体权限

支持转发访问会话 (FAS)：是

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。

Amazon EFS 的服务角色

支持服务角色：是

服务角色是由一项服务担任、代表您执行操作的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。

Warning

更改服务角色的权限可能会破坏 Amazon EFS 的功能。仅当 Amazon EFS 提供相关指导时才编辑服务角色。

Amazon EFS 的服务相关角色

支持服务关联角色：是

服务相关角色是一种与服务相关联的 AWS 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理 Amazon EFS 服务相关角色的详细信息，请参阅[对 Amazon EFS 使用服务相关角色](#)。

适用于 Amazon Elastic File System 的基于身份的策略示例

默认情况下，用户和角色没有创建或修改 Amazon EFS 资源的权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的[创建 IAM 策略 \(控制台\)](#)。

有关 Amazon EFS 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》中的 [Amazon Elastic File System 的操作、资源和条件密钥](#)。ARNs

主题

- [策略最佳实践](#)

- [使用 Amazon EFS 控制台](#)
- [示例：允许用户查看他们自己的权限](#)
- [示例：强制创建加密文件系统](#)
- [示例：强制创建未加密文件系统](#)

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 Amazon EFS 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#) 或 [工作职能的 AWS 托管策略](#)。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的策略和权限](#)。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定 AWS 服务的（例如）使用的，则也可以使用条件来授予对服务操作的访问权限 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 [使用 IAM Access Analyzer 验证策略](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 [使用 MFA 保护 API 访问](#)。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实践](#)。

使用 Amazon EFS 控制台

要访问 Amazon Elastic File System 控制台，您必须具有一组最低权限。这些权限必须允许您列出和查看有关您的 Amazon EFS 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用 Amazon EFS 控制台，还需要将 Amazon EFS AmazonElasticFileSystemReadOnlyAccess AWS 托管策略附加到实体。有关更多信息，请参阅《IAM 用户指南》中的[为用户添加权限](#)。

可在以下位置查看 AmazonElasticFileSystemReadOnlyAccess 和其他 Amazon EFS 托管式服务策略：[AWS 亚马逊 EFS 的托管策略](#)。

示例：允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
    },
  ],
}
```

```

        "Resource": "*"
    }
]
}

```

示例：强制创建加密文件系统

以下示例说明了一个基于身份的策略，该策略允许主体仅创建加密文件系统。

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticfilesystem:CreateFileSystem",
      "Condition": {
        "Bool": {
          "elasticfilesystem:Encrypted": "true"
        }
      },
      "Resource": "*"
    }
  ]
}

```

如果将此策略分配给尝试创建未加密文件系统的用户，请求将失败。无论用户使用的是、[AWS CLI](#) 还是 [AWS 管理控制台](#)，他们都会看到类似于以下内容的消息：

```

User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.

```

示例：强制创建未加密文件系统

以下示例说明了一个基于身份的策略，该策略允许主体仅创建未加密的文件系统。

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticfilesystem:CreateFileSystem",
      "Condition": {
        "Bool": {

```

```
        "elasticfilesystem:Encrypted": "false"
      }
    },
    "Resource": "*"
  }
]
```

如果将此策略分配给尝试创建已加密文件系统的用户，请求将失败。无论用户使用的是、[AWS CLI](#) 还是 [SDK AWS 管理控制台](#)，他们都会看到类似于以下内容的消息：

```
User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

您还可以通过创建 AWS Organizations 服务控制策略 (SCP) 强制创建加密或未加密的 EFS 文件系统。有关服务控制策略的更多信息 AWS Organizations，请参阅《[AWS Organizations 用户指南](#)》中的[服务控制策略](#)。

适用于 Amazon EFS 的基于资源的策略示例

在本节中，您可以找到为各种 Amazon EFS 操作授予或拒绝权限的示例文件系统策略。EFS 文件系统策略有 2 万个字符的限制。有关基于资源的策略的元素的信息，请参阅[Amazon EFS 基于资源的策略](#)。

Important

如果您在文件系统策略中向单个 IAM 用户或角色授予权限，则不要在策略在文件系统上有效时删除或重新创建该用户或角色。如果这样做，该用户或角色将实际在文件系统中锁定，并且将无法访问该用户或角色。有关更多信息，请参阅《[IAM 用户指南](#)》中的[指定主体](#)。

有关如何创建系统策略的信息，请参阅[创建文件系统策略](#)。

主题

- [示例：向特定 AWS 角色授予读写权限](#)
- [示例：授予只读访问权限](#)
- [示例：授予对 EFS 接入点的访问权限](#)

示例：向特定 AWS 角色授予读写权限

在此示例中，EFS 文件系统策略具有以下特征：

- 效果是 Allow。
- AWS 账户中的主体设置为 Testing_Role。
- 操作设置为 ClientMount（读取）和 ClientWrite。
- 授予权限的条件设置为 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"
        }
      }
    }
  ]
}
```

示例：授予只读访问权限

以下文件系统策略仅向 EfsReadOnly IAM 角色授予 ClientMount 或只读权限。

```
{
  "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"
    }
]
}

```

要了解如何设置其他文件系统策略，包括拒绝对所有 IAM 主体（特定管理工作站除外）的根访问权限，请参阅[使用 IAM 授权为 NFS 客户端启用根挤压](#)。

示例：授予对 EFS 接入点的访问权限

您可以使用 EFS 访问策略向 NFS 客户端提供 EFS 文件系统上基于文件的共享数据集的应用程序特定视图。您可以使用文件系统策略向访问点授予对文件系统的权限。

此文件策略示例使用条件元素向由其 ARN 标识的特定访问点授予对文件系统的完全访问权限。

有关使用 EFS 接入点的更多信息，请参阅[使用接入点工作](#)。

```

{
    "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 亚马逊 EFS 的托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住，AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于使用案例的[客户管理型策略](#)来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的[AWS 托管策略](#)。

AWS 托管策略：AWSServiceRoleForAmazonElasticFileSystem

Amazon EFS 使用名为 AWSServiceRoleForAmazonElasticFileSystem 的服务相关角色允许 Amazon EFS 代表您管理 AWS 资源。此角色信任 elasticfilesystem.amazonaws.com 服务来代入相关角色。有关更多信息，请参阅[对 Amazon EFS 使用服务相关角色](#)。

AWS 托管策略：AmazonElasticFileSystemFullAccess

您可以将 AmazonElasticFileSystemFullAccess 策略附加到 IAM 身份。

此策略授予管理权限，允许对 Amazon EFS 进行完全访问并通过访问相关 AWS 服务 AWS 管理控制台。

权限详细信息

该策略包含以下权限。

- elasticfilesystem – 允许主体在 Amazon EFS 控制台中执行所有操作。它还允许委托人使用 AWS Backup 创建 (elasticfilesystem:Backup) 和恢复 (elasticfilesystem:Restore) 备份。
- cloudwatch — 允许委托人在 Amazon EFS 控制台中描述亚马逊 CloudWatch 文件系统指标和某个指标的警报。
- ec2 — 允许委托人在 Amazon EFS 控制台中创建、删除和描述网络接口，描述和修改网络接口属性，描述可用区、安全组、子网、虚拟私有云 (VPCs) 以及与 EFS 文件系统关联的 VPC 属性。

- kms— 允许委托人列出 AWS Key Management Service (AWS KMS) 密钥的别名并在 Amazon EFS 控制台中描述 KMS 密钥。
- iam— 授予创建服务关联角色的权限，该角色允许 Amazon EFS 代表用户管理 AWS 资源。
- iam:PassRole - 授予将 IAM 角色传递至 Amazon EFS 的权限。

要查看此策略的权限，请参阅《AWS 托管策略参考指南》中的 [AmazonElasticFileSystemFullAccess](#)。

AWS 托管策略：AmazonElasticFileSystemReadOnlyAccess

您可以将 AmazonElasticFileSystemReadOnlyAccess 策略附加到 IAM 身份。

此策略授予通过对 Amazon EFS 的只读访问权限 AWS 管理控制台。

权限详细信息

该策略包含以下权限。

- elasticfilesystem – 允许主体在 Amazon EFS 控制台中描述 Amazon EFS 文件系统的属性，包括账户首选项、备份和文件系统策略、生命周期配置、挂载目标及其安全组、标签和接入点。
- cloudwatch— 允许委托人在 Amazon EFS 控制台中检索 CloudWatch 指标并描述指标警报。
- ec2— 允许委托人在 Amazon EFS 控制台中查看可用区、网络接口及其属性、VPCs 安全组、子网及其属性。
- kms— 允许委托人在 Amazon EFS 控制台中列出 AWS KMS 密钥的别名。

要查看此策略的权限，请参阅《AWS 托管策略参考指南》中的 [AmazonElasticFileSystemReadOnlyAccess](#)。

AWS 托管策略：AmazonElasticFileSystemClientFullAccess

可以将 AmazonElasticFileSystemClientFullAccess 策略附加到 IAM 实体。

此策略授予客户端对 EFS 文件系统的读写访问权限。此策略支持 NFS 客户端挂载、读取和写入 EFS 文件系统。

要查看此策略的权限，请参阅《AWS 托管策略参考指南》中的 [AmazonElasticFileSystemClientFullAccess](#)。

AWS 托管策略：AmazonElasticFileSystemClientReadWriteAccess

可以将 AmazonElasticFileSystemClientReadWriteAccess 策略附加到 IAM 实体。

此策略授予客户端对 EFS 文件系统的读写访问权限。此策略支持 NFS 客户端挂载、读取和写入 EFS 文件系统。

要查看此策略的权限，请参阅《AWS 托管式策略参考指南》中的 [AmazonElasticFileSystemClientReadWriteAccess](#)。

Amazon EFS 更新 AWS 了托管策略

查看自该服务开始跟踪这些更改以来对 Amazon EFS AWS 托管策略的更新的详细信息。要获得有关此页面更改的自动提示，请订阅 Amazon EFS [文档历史记录](#) 页面上的 RSS 源。

更改	描述	日期
更新现有策略	策略： AmazonElasticFileSystemFullAccess Amazon EFS 添加了以下策略： - ReplicationRead 和 Replicati onWrite ，授予读取和写入文件系统数据以进行复制的权限。 - iam:PassRole ，授予 Amazon EFS 创建复制配置的权限。	2024 年 11 月 7 日
更新现有策略	策略： AmazonElasticFileSystemServiceRolePolicy Amazon EFS 添加了 ReplicationRead 和 ReplicationWrite 策略，授予读取和写入文件系统数据以进行复制的权限。	2024 年 11 月 7 日
更新现有策略	策略： AmazonElasticFileSystemReadOnlyAccess Amazon EFS 添加了 ReplicationRead 操作，授予读取文件系统数据以进行复制的权限。	2024 年 11 月 7 日
更新现有策略	策略： AmazonElasticFileSystemReadOnlyAccess	2024 年 8 月 7 日

更改	描述	日期
	Amazon EFS 新增了权限，允许源账户和目标账户访问文件系统以进行跨账户复制。	
更新现有策略	策略： AmazonElasticFileSystemFullAccess Amazon EFS 添加了一项新权限，以允许主体禁用和启用文件系统的保护。需要这些权限才能允许 Amazon EFS 复制到现有文件系统。	2023 年 11 月 27 日
更新现有策略	策略： AmazonElasticFileSystemServiceRolePolicy Amazon EFS 添加了新的权限，允许主体创建、描述和删除 Amazon EFS 副本，以及创建 Amazon EFS 文件系统。需要这些权限才能允许 Amazon EFS 代表用户管理文件系统复制配置。	2022 年 1 月 25 日
更新现有策略	策略： AmazonElasticFileSystemReadOnlyAccess Amazon EFS 添加了一项新权限，允许主体描述 Amazon EFS 复制。需要这些权限才能允许用户查看文件系统复制配置。	2022 年 1 月 25 日
更新现有策略	策略： AmazonElasticFileSystemFullAccess Amazon EFS 添加了新的权限，允许主体创建、描述和删除 Amazon EFS 复制。需要这些权限才能允许用户管理文件系统复制配置。	2022 年 1 月 25 日
已开启跟踪策略	策略： AmazonElasticFileSystemClientReadWriteAccess 向 NFS 客户端授予对 Amazon EFS 文件系统的读写权限。	2022 年 1 月 3 日
已开启跟踪策略	策略： AmazonElasticFileSystemServiceRolePolicy Amazon EFS 的服务相关角色权限。	2021 年 10 月 8 日

更改	描述	日期
更新现有策略	策略： AmazonElasticFileSystemFullAccess Amazon EFS 添加了新权限，允许主体修改和描述 Amazon EFS 账户首选项。需要这些权限才能允许用户在 Amazon EFS 控制台中查看和设置账户首选项设置。	2021 年 5 月 7 日
对现有策略的更新	策略： AmazonElasticFileSystemReadOnlyAccess Amazon EFS 添加了新权限，允许主体描述 Amazon EFS 账户首选项。需要这些权限才能允许用户在 Amazon EFS 控制台中查看账户首选项设置。	2021 年 5 月 7 日
Amazon EFS 已开始跟踪更改	Amazon EFS 开始跟踪其 AWS 托管策略的变更。	2021 年 5 月 7 日

将标签与 Amazon EFS 一起使用

可以使用标签来控制对 Amazon EFS 资源的访问，并实现基于属性的访问权限控制 (ABAC)。有关更多信息，请参阅：

- [标记 EFS 资源](#)
- [根据资源上的标签控制访问权限](#)
- [ABAC 有什么用 AWS ?](#) 在 IAM 用户指南中

Note

Amazon EFS 复制不支持将标签用于基于属性的访问权限控制 (ABAC)。

要在创建期间对 Amazon EFS 资源应用标签，用户必须具有某些 AWS Identity and Access Management (IAM) 权限。

在创建过程中授予标记资源的权限

以下标签创建 Amazon EFS API 操作允许您在创建资源时指定标签。

- CreateAccessPoint
- CreateFileSystem

为使用户能够在创建时为资源添加标签，他们必须具有使用创建该资源的操作（如 `elasticfilesystem:CreateAccessPoint` 或 `elasticfilesystem:CreateFileSystem`）的权限。如果在资源创建操作中指定了标签，则会对该操作 AWS 执行额外授权，以验证用户是否有权创建标签。`elasticfilesystem:TagResource` 因此，用户还必须具有使用 `elasticfilesystem:TagResource` 操作的显式权限。

在 `elasticfilesystem:TagResource` 操作的 IAM 策略定义中，使用带有 `Condition` 条件键的 `elasticfilesystem:CreateAction` 元素，为创建资源的操作授予添加标签的权限。

Example策略：仅允许在创建时向文件系统添加标签

下面的示例策略允许用户创建文件系统，并在创建期间向文件系统应用标签。用户无权标记任何现有资源（他们无法直接调用 `elasticfilesystem:TagResource` 操作）。

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

使用标签控制对 Amazon EFS 资源的访问权限

要控制对 Amazon EFS 资源和操作的访问权限，您可以根据标签使用 IAM 策略。您可以使用两种方法提供此类控制：

- 您可以根据这些资源上的标签控制对 Amazon EFS 资源的访问权限。
- 您可以控制在 IAM 请求条件中传递哪些标签。

有关如何使用标签控制 AWS 资源访问的信息，请参阅 IAM 用户指南中的[使用标签控制访问权限](#)。

根据资源上的标签控制访问权限

要控制用户或角色可以对 Amazon EFS 资源执行的操作，您可以使用资源上的标签。例如，您可能希望根据文件系统资源上的标签的键/值对允许或拒绝对该资源执行特定的 API 操作。

Example策略：仅在使用特定标签时创建文件系统

只有当用户使用特定标签键值对标记文件系统时，以下示例策略才允许用户创建文件系统，在本示例中为 key=Department，value=Finance。

```
{
  "Effect": "Allow",
  "Action": [
    "elasticfilesystem:CreateFileSystem",
    "elasticfilesystem:TagResource"
  ],
  "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*",
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/Department": "Finance"
    }
  }
}
```

Example策略：删除带有特定标签的文件系统

以下示例策略只允许用户删除带有 Department=Finance 标签的文件系统。

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "elasticfilesystem:DeleteFileSystem"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Department": "Finance"
      }
    }
  }
]
```

对 Amazon EFS 使用服务相关角色

Amazon EFS 使用 AWS Identity and Access Management (IAM) [服务相关角色](#)。Amazon EFS 服务相关角色是一种独特类型的 IAM 角色，它与 Amazon EFS 直接相关。预定义的 Amazon EFS 服务相关角色包括该服务 AWS 服务 代表您呼叫他人所需的权限。

服务相关角色可让您更轻松地设置 Amazon EFS，因为您不必手动添加必要的权限。Amazon EFS 定义了其服务相关角色的权限，并且仅 Amazon EFS 可以代入其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其他 IAM 实体的权限策略。

只有在先删除您的 Amazon EFS 文件系统后，才能删除 Amazon EFS 服务相关角色。这将保护您的 Amazon EFS 资源，因为您不会无意中删除对资源的访问权限。

服务相关角色允许通过 AWS CloudTrail 查看所有 API 调用。这样便于满足监控和审核要求，因为您可以跟踪 Amazon EFS 代表您执行的所有操作。有关更多信息，请参阅 [EFS 服务相关角色的日志条目](#)。

有关更多信息，请参阅《IAM 用户指南》中的 [服务关联角色权限](#)。

Amazon EFS 的服务相关角色权限

Amazon EFS 使用名 `AWSServiceRoleForAmazonElasticFileSystem` 为的服务相关角色允许 Amazon EFS 代表您的 EFS 文件系统调用和管理 AWS 资源。

AWSServiceRoleForAmazonElasticFileSystem 服务相关角色信任代elasticfilesystem.amazonaws.com入该角色。

角色权限策略允许 Amazon EFS 完成策略定义 JSON 中包含的操作：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "backup-storage:MountCapsule",
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:ModifyNetworkInterfaceAttribute",
        "tag:GetResources"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:*:*:key/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "backup:CreateBackupVault",
        "backup:PutBackupVaultAccessPolicy"
      ],
      "Resource": [
        "arn:aws:backup:*:*:backup-vault:aws/efs/automatic-backup-vault"
      ]
    },
    {
      "Effect": "Allow",
```

```

        "Action": [
            "backup:CreateBackupPlan",
            "backup:CreateBackupSelection"
        ],
        "Resource": [
            "arn:aws:backup:*:*:backup-plan:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iam:CreateServiceLinkedRole"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:AWSServiceName": [
                    "backup.amazonaws.com"
                ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "iam:PassRole"
        ],
        "Resource": [
            "arn:aws:iam:*:*:role/aws-service-role/backup.amazonaws.com/AWSServiceRoleForBackup"
        ],
        "Condition": {
            "StringLike": {
                "iam:PassedToService": "backup.amazonaws.com"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticfilesystem:DescribeFileSystems",
            "elasticfilesystem:CreateReplicationConfiguration",
            "elasticfilesystem:DescribeReplicationConfigurations",
            "elasticfilesystem>DeleteReplicationConfiguration",

```

```
        "elasticfilesystem:ReplicationRead",
        "elasticfilesystem:ReplicationWrite"
    ],
    "Resource": "*"
}
]
```

Note

在创建静态加密的新 EFS 文件系统 AWS KMS 时，您必须手动配置 IAM 权限。要了解更多信息，请参阅[加密静态数据](#)。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。有关更多信息，请参阅《IAM 用户指南》中的[服务相关角色权限](#)。

为 Amazon EFS 创建服务相关角色

在大多数情况下，无需手动创建服务相关角色。当您在 AWS 管理控制台、或 AWS API 中为 EFS 文件系统创建挂载目标或复制配置时 AWS CLI，Amazon EFS 会为您创建服务相关角色。

此外，如果您手动删除此角色 service-linked-role，然后需要重新创建，则可以使用相同的过程在您的账户中重新创建该角色。当您为 EFS 文件系统创建挂载目标或复制配置时，Amazon EFS 将为您创建服务相关角色。

但是，如果 Amazon EFS 没有创建，service-linked-role 或者如果您在 Amazon EFS 支持服务相关角色之前就开始使用，则可以手动创建服务相关角色。有关说明，请参阅《IAM 用户指南》中的[创建服务相关角色](#)。

为 Amazon EFS 编辑服务相关角色

Amazon EFS 不允许您编辑 AWSServiceRoleForAmazonElasticFileSystem 服务相关角色。创建服务关联角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的[更新服务相关角色](#)。

删除 Amazon EFS 的服务相关角色

如果您不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。但是，您必须先清除服务相关角色的资源，然后才能手动删除它。有关更多信息，请参阅 [清理资源并保护您的 AWS 账户](#)。

Note

如果当您尝试删除资源时 Amazon EFS 服务正在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟后重试。

使用 IAM 控制台 AWS CLI、或 AWS API 删除 `AWSServiceRoleForAmazonElasticFileSystem` 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的 [删除服务关联角色](#)。

Amazon EFS 服务相关角色支持的区域

Amazon EFS 支持在所有提供服务 AWS 区域的地方使用与服务相关的角色。有关更多信息，请参阅《AWS 一般参考 用户指南》中的 [AWS 服务端点](#)。

Amazon Elastic File System 身份和访问问题排查

您可以使用以下信息，帮助诊断和修复在使用 Amazon EFS 和 IAM 时可能遇到的常见问题。

主题

- [我无权在 Amazon EFS 中执行操作](#)
- [我无权执行 `iam : PassRole`](#)
- [我想允许我以外的人访问我 AWS 账户 的 Amazon EFS 资源](#)

我无权在 Amazon EFS 中执行操作

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 `mateojackson` IAM 用户尝试使用控制台查看有关虚构 `my-example-widget` 资源的详细信息，但不拥有虚构 `elasticfilesystem:GetWidget` 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticfilesystem:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 elasticfilesystem:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam : PassRole

如果您收到一个错误，指明您无权执行 iam:PassRole 操作，则必须更新策略以允许您将角色传递给 Amazon EFS。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 Amazon EFS 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人访问我 AWS 账户 的 Amazon EFS 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解 Amazon EFS 是否支持这些功能，请参阅[Amazon Elastic File System 如何与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅[IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限](#)。
- 要了解如何向第三方提供对您的资源的访问[权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限](#)。AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的[为经过外部身份验证的用户（身份联合验证）提供访问权限](#)。

- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 [IAM 中的跨账户资源访问](#)。

使用 IAM 控制文件系统访问

您可以使用 IAM 身份策略和资源策略，以针对云环境可扩展和优化的方式控制客户端对 Amazon EFS 资源的访问。通过使用 IAM，您可以允许客户端对文件系统执行特定操作，包括只读、写入和根访问。对 IAM 身份策略或文件系统资源策略中的操作的“允许”权限允许访问该操作。无需在身份策略和资源策略中同时授予此权限。

NFS 客户端可以在连接到 EFS 文件系统时使用 IAM 角色标识自己。当客户端连接到文件系统时，Amazon EFS 会评估文件系统的 IAM 资源策略（称为文件系统策略）以及任何基于身份的 IAM 策略，以确定要授予的相应文件系统访问权限。

当您对 NFS 客户端使用 IAM 授权时，客户端连接和 IAM 授权决策将记录到 AWS CloudTrail。有关如何使用记录 Amazon EFS API 调用的更多信息 CloudTrail，请参阅[使用记录 Amazon EFS API 调用 AWS CloudTrail](#)。

Important

必须使用 EFS 挂载帮助程序挂载 EFS 文件系统，才能使用 IAM 授权来控制客户端访问。有关更多信息，请参阅[使用 IAM 授权挂载](#)。

默认 EFS 文件系统策略

默认 EFS 文件系统策略不使用 IAM 进行身份验证，它向可以使用挂载目标连接到文件系统的任何匿名客户端授予完全访问权限。每当用户配置的文件系统策略不生效时（包括在创建文件系统时），默认策略将生效。每当默认文件系统策略生效时，[DescribeFileSystemPolicy](#) API 操作都会返回 PolicyNotFound 响应。

客户端的 EFS 操作

您可以为使用文件系统策略访问文件系统的客户端指定以下操作。

Action	说明
elasticfilesystem:ClientMount	提供对文件系统的只读访问权限。

Action	说明
elasticfilesystem:ClientWrite	提供对文件系统的写入权限。
elasticfilesystem:ClientRootAccess	提供在访问文件系统时使用根用户的权限。

客户端的 EFS 条件键

要表示条件，您可以使用预定义的条件键。Amazon EFS 为 NFS 客户端提供了以下预定义的条件键。使用 IAM 控制来保护对 EFS 文件系统的访问时，不会强制执行任何其他条件键。

EFS 条件键	说明	运算符
aws:SecureTransport	使用此键可要求客户端在连接到 EFS 文件系统时使用 TLS。	布尔值
aws:SourceIp	使用此键可将请求者的 IP 地址与您在策略中指定的 IP 地址进行比较。aws:SourceIp 条件键只能用于公有 IP 地址范围。	字符串
elasticfilesystem:AccessPointArn	客户端正在连接到的 EFS 接入点的 ARN。	字符串
elasticfilesystem:AccessedViaMountTarget	使用此键可防止未使用文件系统挂载目标的客户机访问 EFS 文件系统。	布尔值

文件系统策略示例

要查看 Amazon EFS 文件系统策略的示例，请参阅[适用于 Amazon EFS 的基于资源的策略示例](#)。

Amazon EFS 的合规性验证

要了解是否属于特定合规计划的范围，请参阅AWS 服务“[按合规计划划分的范围](#)”，然后选择您感兴趣的合规计划。AWS 服务 有关一般信息，请参阅[AWS 合规计划AWS](#)。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的“[下载报告](#)”中的“[AWS Artifact](#)”。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。有关您在使用时的合规责任的更多信息 AWS 服务，请参阅[AWS 安全文档](#)。

Amazon EFS 中的韧性

AWS 全球基础设施围绕 AWS 区域和可用区 (AZ) 构建。AWS 区域 提供多个在物理上独立且隔离的可用区，这些可用区与延迟率低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

Amazon EFS 文件系统可灵活应对 AWS 区域中的一个或多个可用区故障。挂载目标本身设计为具有高可用性。在设计实现高可用性和失效转移到其它可用区的功能时，请务必注意，尽管在每个 AZ 中，您的挂载目标的 IP 地址和 DNS 都是静态的，但它们是由多个资源支持的冗余组件。有关更多信息，请参阅 [如何将 Amazon EFS 与 Amazon EC2 结合使用](#)。

有关 AWS 区域和可用区的更多信息，请参阅 [AWS 全球基础设施](#)。

控制 NFS 客户端对 EFS 文件系统的网络访问

您可以使用网络层安全性和 EFS 文件系统策略控制 NFS 客户端对 Amazon EFS 文件系统的访问。您可以使用 Amazon EC2 提供的网络层安全机制，例如 VPC 安全组规则和网络 ACLs。您还可以使用 AWS IAM 通过 EFS 文件系统策略和基于身份的策略来控制 NFS 访问权限。

主题

- [使用 VPC 安全组](#)
- [使用 Amazon EFS 中的接口 VPC 端点](#)

使用 VPC 安全组

使用 Amazon EFS 时，需要为 EC2 实例指定 VPC 安全组，并为与文件系统关联的 EFS 挂载目标指定安全组。安全组将充当防火墙，您添加的规则将定义流量。在[入门练习](#)中，您在启动 EFS 实例时创建了一个安全组。然后，您将另一个安全组与 EFS 挂载目标相关联（即，您的默认 VPC 的默认安全组）。这种方法适用于入门练习。但对于生产系统，应设置具有用于 Amazon EFS 的最低权限的安全组。

您可以为您的 EFS 文件系统授予入站和出站访问权限。为此，您添加一些规则，以支持 EFS 实例使用网络文件系统（NFS）端口通过挂载目标连接到 EFS 文件系统。

- 每个挂载文件系统的 EC2 实例都必须有一个安全组，其规则应支持在 NFS 端口 2049 上对挂载目标进行出站访问。
- EFS 挂载目标需要有一个安全组，其规则应支持从待挂载文件系统的每个 EC2 实例通过 NFS 端口 2049 进行入站访问。

下表显示了所需的特定安全组规则：

安全组	规则类型	协议	端口：	源/目标
EC2 实例	出站	TCP	2049	挂载目标安全组
挂载目标	入站	TCP	2049	EC2 实例安全组

使用 Amazon EFS 的源端口

为了支持各种不同的 NFS 客户端，Amazon EFS 允许来自任何源端口的连接。如果您要求仅授权的用户可以访问 Amazon EFS，我们建议您使用以下客户端防火墙规则。使用 SSH 连接到文件系统并运行以下命令：

```
iptables -I OUTPUT 1 -m owner --uid-owner 1-4294967294 -m tcp -p tcp --dport 2049 -j DROP
```

该命令在 OUTPUT 链（-I OUTPUT 1）开头插入新的规则。该规则禁止任何未授权的非内核进程（-m owner --uid-owner 1-4294967294）打开到 NFS 端口 2049（-m tcp -p tcp -dport 2049）的连接。

网络访问的安全注意事项

NFS 4.1 (NFSv4.1) 版本的客户机只有在能够与文件系统挂载目标之一的 NFS 端口 (TCP 端口 2049) 建立网络连接时才能装载文件系统。同样，NFSv4.1 客户端只有在能够建立网络连接的情况下才能在访问文件系统时声明用户和组 ID。

能否使此网络连接由以下各因素共同决定：

- 由挂载目标的 VPC 提供的网络隔离 – 文件系统挂载目标不能具有关联的公有 IP 地址。可挂载文件系统的唯一目标包括：
 - 本地 VPC 中的 Amazon EC2 实例
 - 已连接的 EC2 实例 VPCs
 - 使用 AWS Direct Connect 和 AWS Virtual Private Network (VPN) 连接到 Amazon VPC 的本地服务器
- 客户端和挂载目标的 VPC 子网的网络访问控制列表 (ACLs)，用于从挂载目标子网外部进行访问 — 要挂载文件系统，客户端必须能够与挂载目标的 NFS 端口 2049 建立 TCP 连接并接收返回流量。
- 客户端和挂载目标的 VPC 安全组的规则 (用于所有访问) – 要使 EC2 实例能够挂载文件系统，以下安全组规则必须生效：
 - 文件系统必须具有一个挂载目标，其网络接口具有的安全组的规则支持在 NFS 端口 2049 上具有来自实例的入站连接。您可以按 IP 地址 (CIDR 范围) 或安全组启用入站连接。挂载目标网络接口上的入站 NFS 端口的安全组规则来源是文件系统访问控制的关键要素。文件系统挂载目标的网络接口不使用 NFS 端口 2049 以外的入站规则以及任何出站规则。
 - 挂载实例必须具有一个网络接口，其安全组规则支持建立到文件系统的某个挂载目标上的 NFS 端口 2049 的出站连接。您可以按 IP 地址 (CIDR 范围) 或安全组启用出站连接。

有关更多信息，请参阅 [管理挂载目标](#)。

创建安全组

为 EC2 实例和 EFS 挂载目标创建安全组

以下是为 Amazon EFS 创建安全组时要执行的一般步骤。有关创建安全组的说明，请参阅《Amazon VPC 用户指南》中的 [创建安全组](#)。

1. 对于 EC2 实例，请使用以下规则创建安全组：

- 入站规则，支持在端口 22 上使用 Secure Shell (SSH) 从您的 IP 地址或网络进行入站访问。或者，限制源地址以提高安全性。

- 出站规则，支持通过 NFS 端口 2049 对挂载目标安全组进行出站访问。将挂载目标安全组识别为目标。
2. 对于 EFS 挂载目标，请使用以下规则创建安全组：
- 入站规则，支持通过 NFS 端口 2049 从 EC2 安全组进行访问。将 EC2 安全组识别为源。

 Note

您不需要添加出站规则，因为默认出站规则支持所有出站流量。

使用 Amazon EFS 中的接口 VPC 端点

要在虚拟私有云 (VPC) 与 Amazon EFS API 之间建立专用连接，可以创建接口 VPC 端点。端点提供与 Amazon EFS API 的安全连接，无需互联网网关、NAT 实例或虚拟专用网络 (VPN) 连接。有关更多信息，请参阅 Amazon VPC 用户指南中的使用接口 VPC [终端节点访问 AWS 服务](#)。

接口 VPC 终端节点由 AWS PrivateLink 该功能提供支持，该功能允许使用私有 IP 地址在 AWS 服务之间进行私有通信。要使用 AWS PrivateLink，请使用亚马逊 VPC 控制台、API 或 CLI 在您的 VPC 中为 Amazon EFS 创建接口 VPC 终端节点。这样做会使用为 Amazon EFS API 请求提供服务的私有 IP 地址在您的子网中创建一个弹性网络接口。您还可以 VPCs 使用 Site-to-Site VPN、Direct Connect 或 VPC 对等从本地环境或其他环境访问 VPC 终端节点。有关更多信息，请参阅《Amazon VPC 用户指南》中的[使用 AWS PrivateLink 将 VPC 连接到服务](#)。

为 Amazon EFS 创建接口端点

要为 Amazon EFS 创建接口 VPC 端点，请执行以下操作之一：

- **com.amazonaws.*region*.elasticfilesystem** – 为 Amazon EFS API 操作创建端点。
- **com.amazonaws.*region*.elasticfilesystem-fips** – 为 Amazon EFS API 创建符合[美国联邦信息处理标准 \(FIPS \) 140-2](#) 的端点。

有关 Amazon EFS 端点的完整列表，请参阅《Amazon Web Services 一般参考》中的 [Amazon Elastic File System 的端点和配额](#)。

有关如何创建接口终端节点的更多信息，请参阅 Amazon VPC 用户指南中的使用接口 VPC [终端节点访问 AWS 服务](#)。

为 Amazon EFS 创建 VPC 端点策略

要控制对 Amazon EFS API 的访问权限，您可以将 AWS Identity and Access Management (IAM) 策略附加到您的 VPC 终端节点。此策略指定以下内容：

- 可执行操作的主体。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅《Amazon VPC 用户指南》中的[使用端点策略控制对 VPC 端点的访问](#)。

以下示例显示了一个 VPC 终端节点策略，该策略拒绝所有人通过终端节点创建 EFS 文件系统的权限。示例策略还授予所有人执行所有其他操作的权限。

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "elasticfilesystem:CreateFileSystem",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

网络文件系统 (NFS) 级别用户、组和权限

创建文件系统后，默认情况下，只有根用户 (UID 0) 具有读取、写入和执行权限。为了让其他用户也能修改文件系统，根用户必须明确授予他们访问权限。您可以使用访问点自动创建非根用户可从中写入的目录。有关更多信息，请参阅[使用接入点工作](#)。

EFS 文件系统对象具有关联的 Unix 风格模式。此模式值定义了对该对象执行操作的权限。熟悉 Unix 风格系统的用户可以轻松了解 Amazon EFS 在这些权限方面的行为。

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

Note

NFS 协议支持每个用户最多 16 个组 IDs (GIDs)，任何其他组 GIDs 都将从 NFS 客户端请求中截断。有关更多信息，请参阅 [拒绝访问 NFS 文件系统上允许的文件](#)。

下面，您可以找到权限示例以及有关 Amazon EFS 的 NFS 权限注意事项的讨论。

主题

- [文件和目录权限](#)
- [EFS 文件系统使用案例和权限示例](#)
- [文件系统中文件和目录的用户和组 ID 权限](#)
- [无根挤压](#)
- [权限缓存](#)
- [更改文件系统对象所有权](#)
- [EFS 接入点](#)

文件和目录权限

除非被 EFS 接入点覆盖，否则 EFS 文件系统中的文件和目录支持基于挂载 NFSv4 .1 客户端声明的用户和组 ID 的标准 Unix 样式的读取、写入和执行权限。有关更多信息，请参阅 [网络文件系统 \(NFS\) 级别用户、组和权限](#)。

Note

默认情况下，这一层访问控制取决于对 NFSv4 .1 客户端的用户和组 ID 的信任。您可以使用 AWS Identity and Access Management (IAM) 基于资源的策略和身份策略对 NFS 客户端进行授权，并提供只读、写入和根访问权限。您可以使用 EFS 访问点覆盖 NFS 客户端提供的操作系统用户和组标识信息。有关更多信息，请参阅[使用 IAM 控制文件系统访问](#)和[创建接入点](#)。

文件和目录的读取、写入和执行权限的一个示例是，Alice 可能有权在文件系统上的个人目录 /alice 中读取和写入所需的任何文件。不过，在本示例中，不允许 Alice 在同一文件系统上的 Mark 个人目录 /mark 中读取或写入任何文件。允许 Alice 和 Mark 读取共享目录 /share 中的文件，但不能在其中写入文件。

EFS 文件系统使用案例和权限示例

创建 EFS 文件系统并在 VPC 中创建该文件系统的挂载目标后，您可以将远程文件系统本地挂载到您的 Amazon EC2 实例上。mount 命令可以挂载文件系统上的任何目录。不过，在您首次创建文件系统时，只有 / 处的一个根目录。根用户和根组拥有挂载的目录。

以下 mount 命令将由文件系统 DNS 名称标识的 Amazon EFS 文件系统的根目录挂载到 /efs-mount-point 本地目录中。

```
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
```

初始权限模式可授予以下权限：

- 对所有者根目录的 read-write-execute 权限
- 对组根目录的 read-execute 权限
- 对其他目录的 read-execute 权限

只有根用户可以修改此目录。根用户还可以向其他用户授予对此目录的写入权限。例如：

- 创建可写的每用户子目录。有关 step-by-step 说明，请参阅[教程：创建可写的每用户子目录](#)。
- 支持用户写入 EFS 文件系统根目录。具有根用户权限的用户可以向其他用户授予访问该文件系统的权限。
 - 要将 EFS 文件系统所有权更改为非根用户和组，请使用以下命令：

```
$ sudo chown user:group /EFSroot
```

- 要更改文件系统的权限使其更加宽松，请使用以下命令：

```
$ sudo chmod 777 /EFSroot
```

此命令向安装了文件系统的所有 EC2 实例上的所有用户授予 read-write-execute 权限。

文件系统中文件和目录的用户和组 ID 权限

EFS 文件系统中的文件和目录支持基于用户 ID 和组的标准 UNIX 式读取、写入和执行权限。IDs 当 NFS 客户端在不使用访问点的情况下挂载 EFS 文件系统时，客户端提供的用户 ID 和组 ID 将受信任。您可以使用 EFS 接入点来覆盖 NFS 客户端 IDs 使用的用户 ID 和组。当用户尝试访问文件和目录时，Amazon EFS 会检查其用户 IDs 和群组，IDs 以验证每个用户是否有权访问这些对象。Amazon EFS 还使用它们 IDs 来指明用户创建的新文件和目录的所有者和群组所有者。Amazon EFS 不会检查用户或组的名称，它仅使用数字标识符。

Note

在 EC2 实例上创建用户时，可为用户分配任何数字用户 ID (UID) 和组 ID (GID)。在 Linux 系统上，数字用户 IDs 是在 `/etc/passwd` 文件中设置的。数字组 IDs 在 `/etc/group` 文件中。这些文件定义名称与 ID 之间的映射。在 EC2 实例之外，Amazon EFS 不对这些实例执行任何身份验证 IDs，包括根 ID 为 0。

如果用户从两个不同的 EC2 实例访问 EFS 文件系统，根据用户的 UID 在这些实例上是相同还是不同，您会看到如下所示的不同行为：

- 如果两个 EC2 实例上的用户 ID 相同，Amazon EFS 会将其视为指明同一用户，而不考虑他们使用的 EC2 实例。从两个 EC2 实例访问文件系统的用户体验相同。
- 如果两个 EC2 实例上的用户 ID 不相同，则 Amazon EFS 会将其视为不同的用户。从两个不同的 EC2 实例访问 EFS 文件系统的用户体验不相同。
- 如果不同 EC2 实例上的两个不同用户共享一个 ID，则 Amazon EFS 会将其视为同一个用户。

您可以考虑以统一方式管理 EC2 实例间的用户标识映射。用户可以使用 `id` 命令检查其数字 ID。

```
$ id

uid=502(joe) gid=502(joe) groups=502(joe)
```

关闭 ID 映射器

操作系统中的 NFS 实用程序包括一个名为 ID 映射器的守护程序，用于管理用户名和之间的映射。IDs 在 Amazon Linux 中，该守护程序称为 `rpc.idmapd`，在 Ubuntu 中称为 `idmapd`。它能将用户和组 ID 转换为名称，以及反向转换。但是，Amazon EFS 仅处理数字 ID。我们建议您在 EC2 实例上关闭

此进程。在 Amazon Linux 上，ID 映射器通常处于禁用状态，且不启用它。要关闭 ID 映射器，请使用如下所示的命令。

```
$ service rpcidmapd status
$ sudo service rpcidmapd stop
```

无根挤压

默认情况下，EFS 文件系统上禁用根挤压。Amazon EFS 的行为与 `no_root_squash` 的 Linux NFS 服务器类似。如果用户或组 ID 为 0，Amazon EFS 会将该用户视为 root 用户，并绕过权限检查（允许访问和修改所有文件系统对象）。当 AWS Identity and Access Management (AWS IAM) 身份或资源策略不允许访问 `ClientRootAccess` 操作时，可以在客户端连接上启用根压缩。当根挤压处于启用状态时，根用户将被转换为在 NFS 服务器上具有有限权限的用户。

有关更多信息，请参阅 [使用 IAM 控制文件系统访问](#)。

使用 IAM 授权为 NFS 客户端启用根挤压

您可以将 Amazon EFS 配置为阻止除单个管理工作站之外的所有 AWS 委托人以 root 权限访问您的 EFS 文件系统。可以通过为网络文件系统（NFS）客户端配置 AWS Identity and Access Management（IAM）授权来执行此操作。

为此，需要配置两个 IAM 权限策略，如下所示：

- 创建 EFS 文件系统策略，该策略明确允许对文件系统进行了读取和写入访问，并隐式拒绝根访问。
- 使用 EC2 实例配置文件将 IAM 身份分配给需要对文件系统进行了根访问的 Amazon EC2 管理工作站。有关 Amazon EC2 实例配置文件的更多信息，请参阅《AWS Identity and Access Management 用户指南》中的 [使用实例配置文件](#)。
- 将 `AmazonElasticFileSystemClientFullAccess` AWS 托管策略分配给管理工作站的 IAM 角色。有关 Amazon EFS AWS 托管策略的更多信息，请参阅 [Amazon EFS 的身份和访问管理](#)。

要使用 IAM 授权为 NFS 客户端启用根挤压，请使用以下过程。

防止对文件系统进行了根访问

1. 打开 Amazon Elastic File System 控制台，网址为 <https://console.aws.amazon.com/efs/>。
2. 选择文件系统。
3. 从“文件系统”页面中选择要启用根挤压的文件系统。

4. 在文件系统详细信息页面上，选择文件系统策略，然后选择编辑。此时将显示 File system policy (文件系统策略) 页面。
5. 在策略选项下，选择默认阻止根访问*。策略 JSON 对象将显示在策略编辑器中。
6. 选择 Save (保存) 以保存文件系统策略。

非匿名客户端可以通过基于身份的策略获得对文件系统的根访问权限。将 AmazonElasticFileSystemClientFullAccess 托管的策略附加到工作站的角色时，IAM 会根据工作站的身份策略授予对工作站的根访问权限。

从管理工作站启用根访问权限

1. 使用 <https://console.aws.amazon.com/iam/> 打开 IAM 控制台。
2. 为 Amazon EC2 创建一个名为 EFS-client-root-access 的角色。IAM 会创建一个与您创建的 EC2 角色同名的实例配置文件。
3. 将 AWS 托管策略 AmazonElasticFileSystemClientFullAccess 分配给您创建的 EC2 角色。本策略的内容如下所示。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:DescribeMountTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

4. 将实例配置文件附加到您用作管理工作站的 EC2 实例，如下所述。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的[将 IAM 角色附加到实例](#)。
 - a. 打开位于 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 控制台。

- b. 在导航窗格中，选择 Instances (实例)。
- c. 选择实例。对于 Actions (操作)，选择 Instance Settings (实例设置)，然后选择 Attach/Replace IAM role (附加/替换 IAM 角色)。
- d. 选择您在第一步中创建的 IAM 角色 EFS-client-root-access，然后选择 Apply (应用)。
5. 在管理工作站上安装 EFS 挂载帮助程序。有关 EFS 挂载助手挂载帮助程序和 amazon-efs-utils 软件包的更多信息，请参阅[安装 Amazon EFS 客户端](#)。
6. 通过使用带 iam 挂载选项的以下命令，在管理工作站上挂载 EFS 文件系统。

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

您可以将 Amazon EC2 实例配置为通过 IAM 授权自动挂载文件系统。有关使用 IAM 授权挂载 EFS 文件系统的更多信息，请参阅[使用 IAM 授权挂载](#)。

权限缓存

Amazon EFS 会将文件权限缓存一小段时间。因此，可能会有一个短暂的窗口，最近被吊销访问权限的用户仍然可以访问该对象。

更改文件系统对象所有权

Amazon EFS 强制实施 POSIX chown_restricted 属性。这意味着只有根用户可以更改文件系统对象的所有者。root 用户或所有者用户可以更改文件系统对象的所有者组。但是，除非用户是 root 用户，否则该组只能更改为所有者用户所属的组。

EFS 接入点

访问点 将操作系统用户、组和文件系统路径应用于使用访问点发出的任何文件系统请求。访问点的操作系统用户和组覆盖 NFS 客户端提供的任何身份信息。文件系统路径作为访问点的根目录向客户端公开。此方法可确保每个应用程序在访问共享的基于文件的数据集时始终使用正确的操作系统身份和正确的目录。使用访问点的应用程序只能访问其自己的目录及之下目录中的数据。有关接入点的更多信息，请参阅[使用接入点工作](#)。

使用接入点工作

EFS 接入点是 EFS 文件系统中特定于应用程序的入口点，便于轻松地管理应用程序对共享数据集的访问。接入点可以为通过接入点发出的所有文件系统请求强制执行用户身份（包括用户的 POSIX 组）。接入点还可以为文件系统强制执行不同的根目录，以便客户端只能访问指定目录或其子目录中的数据。

您可以使用 AWS Identity and Access Management (IAM) 策略强制特定应用程序使用特定的接入点。可以通过将 IAM 策略与接入点相结合轻松地应用程序提供对特定数据集的安全访问。

您可以使用 AWS 管理控制台、AWS Command Line Interface (AWS CLI) 和 Amazon EFS API 为现有 EFS 文件系统创建接入点。有关创建接入点的 step-by-step 步骤，请参阅[创建接入点](#)。

接入点可与挂载目标配合使用

使用接入点之前，必须在 VPC 中至少创建一个挂载目标。挂载目标提供与 EFS 文件系统的网络连接，而接入点则提供访问控制和特定于应用程序的入口点。

接入点继承挂载目标的可用区放置。

- 安全组是在挂载目标级别应用的，而不是在接入点级别应用的。
- 在有挂载目标的所有可用区中都可以使用接入点。
- IAM 条件密钥 `elasticfilesystem:AccessedViaMountTarget` 可确保仅通过挂载目标访问文件系统，这适用于直接挂载和接入点挂载。

使用访问点挂载文件系统时，您可以使用 EFS 挂载帮助程序。在挂载命令中，包括文件系统 ID、访问点 ID 和 `tls` 挂载选项，如以下示例所示。

```
$ mount -t efs -o tls,iam,accesspoint=fsap-abcdef0123456789a fs-  
abc0123def456789a: /localmountpoint
```

有关使用访问点挂载文件系统的更多信息，请参阅[使用 EFS 接入点进行挂载](#)。

主题

- [使用接入点强制执行用户身份](#)
- [使用接入点强制执行根目录](#)
- [在 IAM 策略中使用接入点](#)

使用接入点强制执行用户身份

您可以使用访问点为通过访问点发出的所有文件系统请求强制执行用户和组信息。要启用此功能，您需要指定在创建访问点时要强制执行的操作系统身份。

其中，您将提供以下内容：

- 用户 ID – 用户的数字 POSIX 用户 ID。
- 组 ID – 用户的数字 POSIX 组 ID。
- 次要组 IDs -次要组的可选列表 IDs。

启用用户强制后，Amazon EFS 会将 NFS 客户端的用户和组 IDs 替换为在接入点上为所有文件系统操作配置的身份。用户强制执行还将执行以下操作：

- 新文件和目录的所有者和组设置为访问点的用户 ID 和组 ID。
- EFS 在评估文件系统权限时会考虑接入点 IDs 的用户 ID、组 ID 和辅助组。EFS 会忽略 NFS 客户端。IDs

Important

强制执行用户身份受 ClientRootAccess IAM 权限的约束。

例如，在某些情况下，您可能会将访问点用户 ID 和/或组 ID 配置为根 ID（即，将 UID 和/或 GID 设置为 0）。在此类情况下，您必须向 NFS 客户端授予 ClientRootAccess IAM 权限。

使用接入点强制执行根目录

您可以使用访问点覆盖文件系统的根目录。在强制执行根目录时，使用接入点的 NFS 客户端使用在接入点上配置的根目录，而不是文件系统的根目录。

您可以通过在创建访问点时设置访问点 Path 属性来启用此功能。Path 属性是通过此访问点发出的所有文件系统请求的文件系统根目录的完整路径。完整路径的长度不能超过 100 个字符。它最多可包含四个子目录。

在访问点上指定根目录时，该目录将成为挂载该访问点的 NFS 客户端的文件系统的根目录。例如，假设访问点的根目录为 /data。在此情况下，使用访问点挂载 fs-12345678:/ 与不使用访问点挂载 fs-12345678:/data 具有相同的效果。

在您的访问点中指定根目录时，请确保配置目录权限，使得接入点用户能够成功装载文件系统。具体而言，请确保为接入点用户或组或者为所有人设置了执行位。例如，目录权限值 755 允许目录用户所有者列出文件、创建文件和装载目录，并允许所有其他用户列出文件和装载目录。

为接入点创建根目录

如果文件系统上没有接入点的根目录路径，则 Amazon EFS 会自动创建具有指定所有权和权限的根目录。如果在创建时未指定目录所有权和权限，Amazon EFS 将不会创建根目录。利用此方法，可以为特定用户或应用程序预置文件系统访问权限，而无需从 Linux 主机挂载文件系统。要创建根目录，在创建接入点时，必须使用以下属性配置根目录所有权和权限：

- OwnerUid – 要用作根目录拥有者的数字 POSIX 用户 ID。
- OwnerGid – 要用作根目录拥有者组的数字 POSIX 组 ID。
- 权限 – 目录的 Unix 模式。一个常见配置是 755。确保为接入点用户设置了执行位，使得他们能够执行装载操作。此配置向目录拥有者授予在目录中输入、列出和写入新文件的权限。它向所有其他用户授予输入和列出文件的权限。有关使用 Unix 文件和目录模式的更多信息，请参阅[网络文件系统 \(NFS\) 级别用户、组和权限](#)。

只有在为接入点根目录指定了 OwnerUid、ownerGid 和权限时，Amazon EFS 才会创建该目录。如果您不提供此信息，Amazon EFS 不会创建根目录。如果根目录不存在，则使用接入点进行挂载的尝试将失败。

在装载带有接入点的文件系统时，如果该目录尚不存在，则会创建该接入点的根目录，前提是根目录 OwnerUid 和权限是在创建接入点时指定的。如果接入点的根目录在挂载时间之前已存在，则接入点不会覆盖现有权限。如果删除根目录，则 EFS 将在下次使用访问点挂载文件系统时重新创建该目录。

Note

如果没有指定接入点根目录的所有权和权限，Amazon EFS 将不会创建该根目录。所有挂载接入点的尝试都将失败。

接入点根目录的安全模型

当根目录覆盖生效时，Amazon EFS 的行为类似于启用了 no_subtree_check 选项的 Linux NFS 服务器。

在 NFS 协议中，服务器生成文件句柄，客户端将这些句柄用作访问文件时的唯一引用。EFS 可以安全地生成不可预测且特定于 EFS 文件系统的文件句柄。当根目录覆盖就位时，Amazon EFS 将不会在指定根目录之外泄露文件的文件句柄。但是，在某些情况下，用户可能会使用某种 out-of-band 机制来获取访问点之外的文件的文件句柄。例如，如果他们有权访问第二个访问点，则他们可能会这样做。如果他们这样做，则可对文件执行读取和写入操作。

对于访问用户的访问点根目录内外的文件，始终强制执行文件所有权和访问权限。

在 IAM 策略中使用接入点

您可以使用 IAM 策略强制规定：由其 IAM 角色标识的特定 NFS 客户端只能访问特定访问点。为此，您可以使用 `elasticfilesystem:AccessPointArn` IAM 条件键。`AccessPointArn` 是挂载文件系统的访问点的 Amazon 资源名称 (ARN)。

以下是一个文件系统策略示例，该策略允许 IAM 角色 `app1` 使用访问点 `fsap-01234567` 来访问文件系统。该策略还允许 `app2` 通过访问点使用文件系统 `fsap-89abcdef`。

JSON

```
{
  "Version": "2012-10-17",
  "Id": "MyFileSystemPolicy",
  "Statement": [
    {
      "Sid": "App1Access",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/app1" },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
      "Condition": {
        "StringEquals": {
          "elasticfilesystem:AccessPointArn" :
            "arn:aws:elasticfilesystem:us-east-1:222233334444:access-point/fsap-01234567"
        }
      }
    },
    {
      "Sid": "App2Access",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/app2" },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite"
      ],
    }
  ]
}
```

```
        "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-  
system/*",  
        "Condition": {  
            "StringEquals": {  
                "elasticfilesystem:AccessPointArn" :  
                "arn:aws:elasticfilesystem:us-east-1:222233334444:access-point/fsap-89abcdef"  
            }  
        }  
    }  
}
```

屏蔽对 EFS 文件系统的公共访问权限

Amazon EFS 屏蔽公共访问权限功能提供设置来帮助您管理对 EFS 文件系统的公共访问。默认情况下，新的 EFS 文件系统不允许公共访问。但是，您可以修改文件系统策略以允许公有访问。

Important

启用“屏蔽公共访问权限”可防止通过直接附加到文件系统的资源策略来授予公共访问权限，有助于保护您的资源。除了启用“屏蔽公共访问权限”之外，还要仔细检查以下策略，来确认它们不会授予公共访问权限：

- 附加到关联 AWS 委托人（例如，IAM 角色）的基于身份的策略
- 附加到关联资源的基于 AWS 资源的策略（例如，AWS Key Management Service (KMS) 密钥）

主题

- [使用以下方式阻止公众访问 AWS Transfer Family](#)
- [“公有”的含义](#)

使用以下方式阻止公众访问 AWS Transfer Family

当您使用 Amazon EFS 与配合使用时 AWS Transfer Family，如果文件系统允许公开访问，则从属于与文件系统不同的账户的 Transfer Family 服务器收到的文件系统访问请求将被阻止。Amazon EFS 会评估文件系统的 IAM 策略，如果策略是公有的，则会阻止相关请求。要允许 AWS Transfer Family 访问您的文件系统，请更新您的文件系统策略，使其不被视为公开。

 Note

对于在 2021 年 1 月 6 日之前创建的 EFS 文件系统且策略允许公开访问的，默认情况下会禁用 Transfer Family。要允许使用 Transfer Family 访问您的文件系统，请联系 AWS 支持部门。

“公有”的含义

在评估文件系统是否允许公有访问时，Amazon EFS 假设文件系统策略是公有的。然后对文件系统策略进行评估，以确定它是否符合非公有条件。当文件系统策略仅向以下一个或多个对象的固定值（不包含通配符的值）授予访问权限时，才会将该策略视为非公有：

- AWS 委托人、用户、角色或服务主体（例如，`aws:PrincipalOrgID`）
- `aws:SourceArn`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `elasticfilesystem:AccessedViaMountTarget`
- `aws:userid`, outside the pattern `"AROLEID:*"`

根据这些规则，以下示例策略被视为公有。

JSON

```
{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
  "Statement": [
    {
      "Sid": "efs-statement-14a7191c-9401-40e7-a388-6af6cfb7dd9c",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
```

```

        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientRootAccess"
    ],
    "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*"
}
]
}

```

您可以使用设置为 True 的 EFS 条件键 `elasticfilesystem:AccessedViaMountTarget` 将此文件系统策略设为非公有。您可以使用 `elasticfilesystem:AccessedViaMountTarget` 允许对通过文件系统挂载目标访问 EFS 文件系统的客户端执行指定的 EFS 操作。以下非公有策略使用设置为 True 的 `elasticfilesystem:AccessedViaMountTarget` 条件键。

JSON

```

{
  "Version": "2012-10-17",
  "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
  "Statement": [
    {
      "Sid": "efs-statement-14a7191c-9401-40e7-a388-6af6cfb7dd9c",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "elasticfilesystem:ClientMount",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientRootAccess"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
      "Condition": {
        "Bool": {
          "elasticfilesystem:AccessedViaMountTarget": "true"
        }
      }
    }
  ]
}

```

```
}
```

有关 Amazon EFS 条件键的更多信息，请参阅[客户端的 EFS 条件键](#)。有关创建文件系统策略的更多信息，请参阅[创建文件系统策略](#)。

Amazon EFS 网络隔离

作为一项托管式服务，Amazon Elastic File System 受 AWS 全球网络安全保护。有关 AWS 安全服务以及 AWS 如何保护基础设施的信息，请参阅 [AWS 云安全性](#)。要按照基础设施安全最佳实践设计您的 AWS 环境，请参阅《安全性支柱 AWS Well-Architected Framework》中的[基础设施保护](#)。

可以使用 AWS 发布的 API 调用通过网络访问 Amazon EFS。客户端必须支持以下内容：

- 传输层安全性协议 (TLS)。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE (临时 Diffie-Hellman) 或 ECDHE (临时椭圆曲线 Diffie-Hellman)。大多数现代系统 (如 Java 7 及更高版本) 都支持这些模式。

可从任何网络位置调用这些 API，但 Amazon EFS 的确支持基于资源的访问策略，其中可以包含基于源 IP 地址的限制。还可以使用 Amazon EFS 策略来控制来自特定 Amazon Virtual Private Cloud (Amazon VPC) 端点或特定 VPC 的访问。事实上，这仅将对给定 Amazon EFS 资源的网络访问与 AWS 网络中的特定 VPC 隔离开。

Amazon EFS 限额

您可以在下文中找到使用 Amazon EFS 时的配额。

主题

- [您可以提高的 Amazon EFS 配额](#)
- [您无法更改的 Amazon EFS 资源配额](#)
- [NFS 客户端的配额](#)
- [Amazon EFS 文件系统的配额](#)
- [不支持的 NFSv4 .0 和 4.1 功能](#)
- [其他注意事项](#)
- [排查与限额有关的文件操作错误](#)

您可以提高的 Amazon EFS 配额

Service Quotas 是一项 AWS 服务，可帮助您从一个位置管理配额或限制。在 [Service Quotas 控制台](#) 中，您可以查看 Amazon EFS 限制值，并请求增加中的 EFS 文件系统数量的配额 AWS 区域 以及频繁访问数据的读取 IOPS。

您还可以通过联系 AWS Support 请求提高以下 Amazon EFS 配额。要了解更多信息，请参阅[请求提高配额](#)。Amazon EFS 服务团队单独审查每个请求。

- 每个客户账户的文件系统数量。
- 每个文件系统的接入点数。
- 采用弹性吞吐量模式的每个文件系统的最大读取 IOPS。当频繁访问的文件系统的读取 IOPS 增加时，不频繁访问的文件系统的读取 IOPS 和写入 IOPS 也会增加。
- 每个区域性文件系统的弹性吞吐量，适用于 AWS 区域中所有已连接客户端。
- 每个区域性文件系统的预置吞吐量，适用于 AWS 区域中所有已连接客户端。

下表列出了您可以请求更改的常规默认资源配额。

资源	默认配额
一个客户账户中每个客户账户的文件系统数量 AWS 区域	1000
每个文件系统的接入点数	10000
采用弹性吞吐量模式的每个文件系统的最大 IOPS	不频繁访问的数据读取：90,000
对于频繁访问的数据，您可以请求将读取 IOPS 的默认配额提高多达 10 倍。在为频繁访问的数据提高读取 IOPS 后，也将提高不频繁访问的数据的读取 IOPS 和写入 IOPS。	频繁访问的数据读取：250,000
	写入：50,000

下表列出了每个文件系统中所有连接的客户端的弹性吞吐量配额 AWS 区域。

区域性文件系统 — 每个文件系统中所有连接的客户端的默认弹性吞吐量总计 AWS 区域

AWS 区域	最大读取吞吐量	最大写入吞吐量 (计量吞吐量)
美国东部 (俄亥俄州) 区域	每秒 60 千兆字节 () GiBps	5 GiBps
美国东部 (弗吉尼亚州北部) 区域		
US West (Oregon) Region		
亚太地区 (孟买) 区域		
亚太地区 (首尔) 区域		
亚太地区 (新加坡) 区域		
亚太地区 (悉尼) 区域		
Asia Pacific (Tokyo) Region		

AWS 区域	最大读取吞吐量	最大写入吞吐量 (计量吞吐量)
欧洲 (法兰克福) 区域		
欧洲地区 (爱尔兰) 区域		
欧洲地区 (伦敦) 区域		
所有其他 AWS 区域	20 GiBps	1 GiBps

下表列出了每个文件系统中所有连接的客户端的预配置吞吐量配额。AWS 区域

区域文件系统-每个文件系统中所有连接的客户端的默认预配置吞吐量总计 AWS 区域

AWS 区域	最大读取吞吐量	最大写入吞吐量 (计量吞吐量)
美国东部 (俄亥俄州) 区域	10 GiBps	3.33 GiBps
美国东部 (弗吉尼亚州北部) 区域		
美国西部 (俄勒冈州) 区域		
欧洲地区 (爱尔兰) 区域		
所有其他 AWS 区域	3 GiBps	1 GiBps

请求提高配额

要通过请求增加这些配额 AWS 支持，请执行以下步骤。Amazon EFS 团队审查每个配额提高请求。

要申请增加配额，请通过 AWS 支持

1. 打开 [AWS 支持 中心](#) 页面并登录 (如有必要)。然后选择 Create Case (创建案例)。
2. 在 Create case (创建案例) 下，选择 Service Limit Increase (提高服务限制)。
3. 对于 Limit Type (限制类型)，选择要提高了的限制类型。填写表单中的必填字段，然后选择您的首选联系方式。

您无法更改的 Amazon EFS 资源配额

有多个 Amazon EFS 资源的配额无法更改，包括：

- 常规资源的配额，例如每个文件系统的连接数量。
- 每个单区文件系统的弹性和预置吞吐量限额，适用于 AWS 区域中所有已连接的客户端。
- 每个区域性文件系统或单区文件系统的突增吞吐量限额，适用于 AWS 区域中所有已连接的客户端。

下表列出了无法更改的常规资源配额。

无法更改的通用资源配额

资源	限额
每个文件系统的连接数	25000
可用区中的每个文件系统的挂载目标数	1
每个虚拟私有云 (VPC) 的挂载目标数量	1,400
每个挂载目标的安全组数	5
每个文件系统的标签数	50
每个 VPCs 文件系统的数量	1

Note

客户端还可以连接到与文件系统不同的账户或 VPC 中的挂载目标。有关更多信息，请参阅 [从其他 AWS 账户 或 VPC 挂载 EFS 文件系统](#)。

下表列出了每个 AWS 区域文件系统中所有连接的客户端的默认弹性和预配置吞吐量总限制。

一个区域文件系统 — 每个文件系统中所有连接的客户端的默认弹性和预配置吞吐量总吞吐量 AWS 区域

AWS 区域	最大读取吞吐量	最大写入吞吐量 (计量吞吐量)
全部 AWS 区域	3 GiBps	1 GiBps

下表列出了每个文件系统中所有连接的客户端对每个文件系统的总突发吞吐量限制。 AWS 区域
区域文件系统和单区域文件系统 — 每个文件系统中所有连接的客户端每个文件系统的总突发吞吐量 AWS 区域

AWS 区域	最大读取吞吐量	最大写入吞吐量
美国东部 (俄亥俄州) 区域	5 GiBps	3 GiBps
美国东部 (弗吉尼亚州北部) 区域		
美国西部 (俄勒冈州) 区域		
亚太地区 (悉尼) 区域		
欧洲地区 (爱尔兰) 区域		
所有其他 AWS 区域	3 GiBps	1 GiBps

NFS 客户端的配额

假设是 Linux NFSv4 .1 客户端，则适用于 NFS 客户端的以下配额：

- 对于使用弹性吞吐量并使用版本 2.0 或更高版本的 Amazon EFS 客户端 (版本MiBps) 或 Amazon EFS CSI 驱动程序 (amazon-efs-utils) 装载的文件系统，最大合并读取和写入吞吐量为每秒 1,500 兆字节 (aws-efs-csi-driver)。所有其他文件系统的最大吞吐量为 500 MiBps。有关性能的更多信息，请参阅 [性能摘要](#)。NFS 客户端吞吐量以发送和接收的总字节数形式计算，最小 NFS 请求大小为 4KB (在对读取请求应用 1/3 计量速率后)。
- 每个客户端最多可以有 65536 个活动用户同时打开文件。

- 实例上可同时打开最多 65536 个文件。列出目录内容不会视为打开文件。
- 客户端上的每个唯一挂载每次连接最多可以获得总计 65536 个锁。
- 连接到 Amazon EFS 时，位于本地或位于其他 AWS 区域中的 NFS 客户端，相比从相同 AWS 区域连接到 EFS 时，会出现较低的吞吐量。出现此效果的原因是网络延迟增加。实现每客户端最大的吞吐量需要 1 毫秒或更低的网络延迟。将大型 DataSync 数据集从本地 NFS 服务器迁移到 EFS 时，请使用数据迁移服务。
- NFS 协议支持每个用户最多 16 个组 IDs (GIDs)，任何其他组 GIDs 都将从 NFS 客户端请求中截断。有关更多信息，请参阅 [拒绝访问 NFS 文件系统上允许的文件](#)。
- 不支持将 Amazon EFS 与 Microsoft Windows 结合使用。

Amazon EFS 文件系统的配额

以下配额特定于 Amazon EFS 文件系统：

资源	限额
文件名长度（以字节为单位）	255
符号链接（symlink）长度（以字节为单位）	4,080
文件的硬链接数	177
单个文件大小	52,673,613,135,872 字节（47.9 TiB）
目录深度级别数	1000
跨所有实例和用户的单个文件上的锁数	512
每个文件系统策略的字符限制	20000
*通用模式下每秒文件操作数	250,000

*有关通用模式的每秒文件操作数的更多信息，请参阅[性能摘要](#)。

不支持的 NFSv4 .0 和 4.1 功能

尽管 Amazon EFS 不支持或 NFSv2 NFSv3，但它确实同时支持 NFSv4 .1 和 NFSv4 .0，但以下功能除外：

- pNFS
- 扩展属性
- 任何类型的客户端委派或回调
 - OPEN 操作始终返回 OPEN_DELEGATE_NONE 作为委派类型。
 - OPEN 为 NFSERR_NOTSUPP 和 CLAIM_DELEGATE_CUR 声明类型返回 CLAIM_DELEGATE_PREV。
- 强制锁定

Amazon EFS 中的所有锁定都是建议性锁定，这意味着读取和写入操作在执行之前不会检查是否存在冲突锁定。

- 拒绝共享

NFS 支持共享拒绝的概念。共享拒绝主要由用户的 Windows 客户端用来拒绝其他人访问已打开的特定文件。Amazon EFS 不支持该操作，并对指定除 OPEN4_SHARE_DENY_NONE 之外的共享拒绝值的任何 OPEN 命令返回 NFS 错误 NFS4ERR_NOTSUPP。Linux NFS 客户端不使用 OPEN4_SHARE_DENY_NONE 之外的其他内容。

- 访问控制列表 (ACLs)
- Amazon EFS 不更新文件读取的 `time_access` 属性。Amazon EFS 更新以下事件中的 `time_access`：
 - 创建文件时 (将创建 inode)
 - 在 NFS 客户端显式调用 `setattr` 时
 - 由于文件大小更改或文件元数据更改等原因导致向 inode 写入内容
 - 更新任何 inode 属性
- 命名空间
- 持久性回复缓存
- 基于 Kerberos 的安全性
- NFSv4.1 数据保留
- 目录上的 SetUID

- 使用“创建”操作时不支持的文件类型：块设备 (NF4BLK)、字符设备 (CHR)、属性目录 (ATTR NF4 DI NF4 R) 和命名属性 (NF4NAMEDATTR)。
- 不支持的属性：FATTR4_ARCHIVE、_FILES_AV FATTR4 AIL、_FILES_FREE、_FILES_TOTAL、_FATTR4 FS_LOCATIONS、FATTR4 _MIMETYPE、_QUOTA_AVAI FATTR4 L_HARD、_QUOTA_USED、_TIME_BACKUP 和 _ACL。FATTR4 FATTR4 FATTR4 FATTR4 FATTR4 FATTR4

如果尝试设置这些属性，将导致向客户端发回 NFS4ERR_ATTRNOTSUPP 错误。

其他注意事项

此外，请注意以下情况：

- 有关可创建 Amazon EFS 文件系统的 AWS 区域 列表，请参阅 [《AWS 一般参考 用户指南》](#)。
- Amazon EFS 不支持 nconnect 挂载选项。
- 您可以使用 Direct Connect 和 VPN 从本地数据中心服务器挂载 Amazon EFS 文件系统。有关更多信息，请参阅 [教程：使用本地客户端挂载](#)。

排查与限额有关的文件操作错误

当您访问 EFS 文件系统时，对文件系统中文件的某些限制将适用。超出这些限制会导致文件操作错误。有关 Amazon EFS 中基于文件的限制的更多信息，请参阅[Amazon EFS 限额](#)。

您可以在下文中查找一些常见文件操作错误及与每个错误相关的限制。

主题

- [检查打开的文件和文件锁](#)
- [命令失败，并显示“超出磁盘配额”错误](#)
- [命令失败，并显示“I/O 错误”](#)
- [命令失败，并显示“文件名太长”错误](#)
- [命令失败，并显示“未找到文件”错误](#)
- [命令失败，并显示“链接太多”错误](#)
- [命令失败，并显示“文件太大”错误](#)

检查打开的文件和文件锁

要排查文件操作错误，您可以通过检查 EFS 挂载点上打开的文件和文件锁来检查您的客户端系统是否已达到限制：

- 检查打开的文件 - 使用操作系统的工具列出 EFS 挂载路径下打开的文件。这有助于确定是否已接近打开文件的上限。例如，在 Linux 上，您可以使用：

```
lsof <efs-mount-path>
```

- 检查文件锁 - 使用系统的锁监控工具，列出 EFS 挂载路径下存在锁的文件。这有助于确定是否已达到锁限制。例如，在 Linux 上，您可以使用：

```
lslocks | grep <efs-mount-path>
```

这些命令会显示当前用量相对于 EFS 限制的占比情况，帮助您确定文件操作错误是否与达到系统或服务限制有关。

命令失败，并显示“超出磁盘配额”错误

Amazon EFS 当前不支持用户磁盘配额。如果超出了以下任何限制，则可能会出现该错误：

- 最多 65536 个活动用户可以同时打开文件。多次登录的一个用户账户计为一个活动用户。
- 一个实例同一时刻最多可打开 65536 个文件。列出目录内容不会视为打开文件。
- 客户端上的每个唯一挂载每次连接最多可以获得总计 65536 个锁。

要采取的操作

如果遇到该问题，可通过确定超出了上述哪个限制，然后进行更改以满足该限制，加以解决。有关更多信息，请参阅 [NFS 客户端的配额](#)。要查看您的当前用量，请参阅[检查打开的文件和文件锁](#)。

命令失败，并显示“I/O 错误”

遇到下列问题之一时会发生此错误：

- 每个实例同一时刻有超过 65536 个活动用户账户打开文件。

要采取的操作

如果遇到该问题，您可以满足在实例上支持的打开文件数限制以解决该问题。为此，请减少在实例上同时打开 Amazon EFS 文件系统中的文件的活动用户数。要查看您的当前用量，请参阅[检查打开的文件和文件锁](#)。

- 加密您的文件系统的 AWS KMS 密钥已删除。

要采取的操作

如果遇到此问题，则您不能再解密用该密钥加密的数据，这意味着该数据将无法恢复。

命令失败，并显示“文件名太长”错误

当文件名或其符号链接 (symlink) 太长时，会出现该错误。文件名具有以下限制：

- 名称的长度最多为 255 个字节。
- 符号链接的大小最多为 4080 个字节。

要采取的操作

如果遇到该问题，可通过减小您的文件名或符号链接的长度以满足支持的限制，加以解决。

命令失败，并显示“未找到文件”错误

之所以出现此错误，是因为某些较早的 32 位版本的 Oracle 电子商务套件使用 32 位文件 I/O 接口，而 EFS 使用 64 位索引节点编号。可能失败的系统调用包括 ``stat ()`` 和 ``readdir ()``。

要采取的操作

如果遇到此错误，可以使用 `nfs.enable_ino64=0` kernel 启动选项解决问题。此选项将 64 位 EFS inode 数压缩为 32 位。对于不同的 Linux 发行版，内核启动选项的处理方式不同。在 Amazon Linux 上，向 `/etc/default/grub` 中的 `GRUB_CMDLINE_LINUX_DEFAULT` 变量添加 `nfs.enable_ino64=0` kernel 即可启用该选项。有关如何启用内核启动选项的信息，请参阅特定于您的发行版的具体文档。

命令失败，并显示“链接太多”错误

当文件的硬链接太多时，会出现该错误。一个文件中最多可有 177 个硬链接。

要采取的操作

如果遇到该问题，可通过减少文件硬链接的数量以满足支持的限制，加以解决。

命令失败，并显示“文件太大”错误

当文件太大时，会出现该错误。单个文件的大小最多为 52,673,613,135,872 个字节 (47.9 TiB)。

要采取的操作

如果遇到该问题，可通过减小文件的大小以满足支持的限制，加以解决。

Amazon EFS API

Amazon EFS API 是基于 [HTTP \(RFC 2616 \)](#) 的网络协议。对于每个 API 调用，您针对要在其中管理文件系统的 AWS 区域，向特定于区域的 Amazon EFS API 端点发出 HTTP 请求。API 会对 HTTP 请求/响应正文使用 JSON (RFC 4627) 文档。

Amazon EFS API 是一种 RPC 模型。在该模型中具有一套固定的操作，客户端已知每个操作的语法，而无需事先进行任何交互。在以下部分中，您可以找到使用抽象 RPC 表示法描述每个 API 操作的信息。不会在线显示每个操作的名称。对于每个操作，该主题指定了指向 HTTP 请求要素的映射。

给定请求映射到的特定 Amazon EFS 操作由请求的方法 (GET、PUT、POST 或 DELETE) 和其请求 URI 所匹配的各种模式共同确定。如果操作为 PUT 或 POST，则 Amazon EFS 将从请求 URI 路径分段、查询参数和请求正文内的 JSON 对象中提取调用参数。

Note

虽然不会在线显示操作名称 (如 `CreateFileSystem`)，但这些名称在 AWS Identity and Access Management (IAM) 策略中是有意义的。有关更多信息，请参阅 [Amazon EFS 的身份和访问管理](#)。

操作名称还可用于为命令行工具中的命令和 AWS SDK API 的元素命名。例如，名为 AWS CLI 的 `create-file-system` 命令映射到 `CreateFileSystem` 操作。

操作名称还会显示在 Amazon EFS API 调用的 AWS CloudTrail 日志中。

API 终端节点

端点是作为 AWS Web 服务入口点的 URL。Amazon EFS 支持以下端点类型：

- IPv4 端点
- 双栈 (IPv4 和 IPv6) 端点
- [FIPS 端点](#)

当您发出请求时，您可以指定要使用的端点。如果不指定端点，则默认使用 IPv4 端点。要使用不同的端点类型，您必须在请求中指定。有关如何执行此操作的示例，请参阅[指定端点](#)。有关可用端点的表，请参阅 [Amazon EFS 端点和配额](#)。

IPv4 端点

IPv4 端点仅支持 IPv4 流量。IPv4 端点适用于所有区域。IPv4 端点的格式取决于区域。

- 对于 AWS 商业区域中的 IPv4 端点，格式为 `elasticfilesystem.region.amazonaws.com`。

例如，如果您指定 `elasticfilesystem.us-west-2.amazonaws.com` 为端点，我们会将您的请求定向到美国西部（俄勒冈州）区域（us-west-2）端点。

- 对于中国区域的 IPv4 端点，格式为 `elasticfilesystem.region.amazonaws.com.cn`。

例如，在中国（北京）区域，Amazon EFS API 的端点为 `elasticfilesystem.cn-north-1.amazonaws.com.cn`。

双栈（IPv4 和 IPv6）端点

同时支持 IPv4 和 IPv6 流量的双堆栈端点。当您向双堆栈端点发出请求时，端点 URL 解析为 IPv6 或 IPv4 地址，具体取决于您的网络和客户端使用的协议。双栈端点的格式取决于区域。

- 对于 AWS 商业区域中的双栈端点，格式为 `efs.aws-region.api.aws`。

例如，对于美国西部（俄勒冈州）区域（us-west-2），双栈端点是 `efs.eu-west-2.api.aws`。

- 对于中国区域的双栈端点，格式为 `efs.region.api.amazonwebservices.com.cn`。

例如，中国（北京）区域的双堆栈端点是 `efs.cn-north-1.api.amazonwebservices.com.cn`。

指定端点

以下示例说明了如何使用 Amazon EFS 的 AWS CLI 为美国西部（北加利福尼亚）区域（us-west-1）指定端点。

IPv4

```
aws efs get-rest-apis --region us-west-1 --endpoint-url https://elasticfilesystem.us-west-1.amazonaws.com
```

双堆栈

```
aws efs get-rest-apis --region us-west-1 --endpoint-url https://efs.us-west-1.api.aws
```

API 版本

用于调用的 API 版本是由请求 URI 的第一个路径分段确定的，并且其格式为 ISO 8601 日期。有关示例，请查看 [CreateFileSystem](#)。

本文档中所描述的版本为 API 版本 2015-02-01。

相关主题

以下各节描述了 API 操作，以及如何创建签名以便进行请求身份验证和如何使用 IAM 策略为这些 API 操作授权。

- [Amazon EFS 的身份和访问管理](#)
- [操作](#)
- [数据类型](#)

使用 Amazon EFS 的查询 API 请求速率

针对每个区域限制每个 AWS 账户的 Amazon EFS API 请求可帮助提高服务性能。无论是来自于应用程序、AWS CLI，还是 Amazon EFS 控制台，所有 Amazon EFS API 调用都不能超过允许的最大 API 请求速率。对于不同的 AWS 区域，最大 API 请求速率可能会有所不同。发出 API 请求是由于底层 AWS 账户。

如果 API 请求超过其类别的 API 请求速率，请求将返回 `ThrottlingException` 错误代码。为防止出现该错误，请确保您的应用程序不会在高速率下重试 API 请求。您可以执行该操作，但前提是在轮询时格外小心并使用指数回退重试。

轮询

您的应用程序可能需要反复调用 API 操作以检查状态更新。在开始轮询之前，请为请求留出完成所需的估算时间。在开始轮询时，请在连续的请求之间添加相应的睡眠间隔。为了获得最佳的效果，请使用递增的睡眠间隔。

重试或批处理

您的应用程序可能需要在失败后重试 API 请求，或处理多个资源（例如，所有 Amazon EFS 文件系统）。要降低 API 请求的速率，请在连续的请求之间添加相应的睡眠间隔。为了获得最佳的效果，请使用递增或可变的睡眠间隔。

计算睡眠间隔

在需要轮询或重试 API 请求时，我们建议您使用指数回退算法计算 API 调用之间的睡眠间隔。指数回退的原理是对于连续错误响应，重试等待间隔越来越长。有关更多信息以及该算法的实施示例，请参阅《Amazon Web Services 一般参考》中的[重试行为](#)。

操作

支持以下操作：

- [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

创建 EFS 接入点 访问点是 EFS 文件系统特定于应用程序的视图，将操作系统用户和组以及文件系统路径应用到通过该访问点发出的任何文件系统请求。操作系统用户和组覆盖 NFS 客户端提供的任何身份信息。文件系统路径作为访问点的根目录公开。使用接入点的应用程序只能访问其自己的目录中以及任何子目录中的数据。一个文件系统最多可配置 10000 个接入点，除非您申请提高该上限。要了解更多信息，请参阅[使用 EFS 访问点挂载文件系统](#)。

Note

如果快速连续发送多个在同一文件系统上创建接入点的请求，并且文件系统接近接入点数量限制，则可能会遇到对这些请求的节流响应。这是为了确保文件系统不超过规定的接入点限制。

此操作需要 `elasticfilesystem:CreateAccessPoint` 操作的权限。

可以在创建时标记接入点。如果在创建操作中指定了标签，则 IAM 会对 `elasticfilesystem:TagResource` 操作执行额外的授权，以验证用户是否具备创建标签的权限。因此，您必须授予使用 `elasticfilesystem:TagResource` 操作的显式权限。有关更多信息，请参阅[在创建过程中授予标记资源的权限](#)。

请求语法

```
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"  
    }  
  ]  
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

[ClientToken](#)

一个由最多 64 个 ASCII 字符组成的字符串，Amazon EFS 使用此字符串来确保幂等性创建。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：.+

是否必需：是

[FileSystemId](#)

接入点提供访问权限的 EFS 文件系统的 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

是否必需：是

[PosixUser](#)

应用于使用接入点发出的所有文件系统请求的操作系统用户和组。

类型：[PosixUser](#) 对象

必需：否

[RootDirectory](#)

指定 EFS 文件系统上的目录，接入点将其作为文件系统的根目录，向使用接入点的 NFS 客户端公开。使用接入点的客户端只能对根目录及之下的目录进行访问。如果指定的 `RootDirectory > Path` 不存在，当客户端连接到接入点时，Amazon EFS 会创建此路径并应用 `CreationInfo` 设置。指定 `RootDirectory` 时，必须提供 `Path` 和 `CreationInfo`。

只有在您提供了 `CreationInfo: OwnUid`、`ownGID` 和目录权限后，Amazon EFS 才会创建根目录。如果您不提供此信息，Amazon EFS 不会创建根目录。如果根目录不存在，则使用接入点进行挂载的尝试将失败。

类型：[RootDirectory](#) 对象

必需：否

[Tags](#)

创建与接入点关联的标签。每个标签都是一个键值对，每个键都必须是唯一的。有关更多信息，请参阅《AWS 通用参考指南》中的为[AWS 资源添加标签](#)。

类型：[Tag](#) 对象数组

必需：否

响应语法

```
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"
    }
  ]
}

```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[AccessPointArn](#)

与接入点关联的唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度限制：最大长度为 128。

模式：`^arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}$`

[AccessPointId](#)

接入点 ID，由 Amazon EFS 分配。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

[ClientToken](#)

请求中指定的不透明字符串，以确保幂等创建。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：`.+`

[FileSystemId](#)

访问点应用到的 EFS 文件系统的 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[LifeCycleState](#)

标识接入点的生命周期阶段。

类型：字符串

有效值：`creating` | `available` | `updating` | `deleting` | `deleted` | `error`

[Name](#)

接入点的名称。这是 Name 标签的值。

类型：字符串

[OwnerId](#)

标识 AWS 账户 拥有接入点资源的。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\d{12})|(\d{4}-\d{4}-\d{4}))$`

PosixUser

完整的 POSIX 身份，包括接入点 IDs 上的用户 ID、组 ID 和辅助组，NFS 客户端使用该接入点进行所有文件操作。

类型：PosixUser 对象

RootDirectory

EFS 文件系统上的目录，接入点将其作为根目录，向使用接入点的 NFS 客户端公开。

类型：RootDirectory 对象

Tags

与接入点关联的标签，以标签对象数组的形式呈现。

类型：Tag 对象数组

错误

AccessPointAlreadyExists

如果尝试创建的接入点已存在，并且在请求中提供了创建令牌，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

AccessPointLimitExceeded

如果 AWS 账户 已经创建了每个文件系统允许的最大数量的接入点，则返回。有关更多信息，请参阅<https://docs.aws.amazon.com/efs/latest/ug/limits.html#limits-efs-resources-per-account-per-region>。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：403

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifecycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ThrottlingException

当调用 `CreateAccessPoint` API 操作的速度过快，且文件系统上的接入点数量接近[限制值 120](#)时返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：429

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

CreateFileSystem

创建新的空文件系统。该操作在 Amazon EFS 用于确保幂等创建的请求中需要一个创建令牌（使用相同的创建令牌调用该操作没有效果）。如果当前不存在由 AWS 账户具有指定创建令牌的调用者拥有的文件系统，则此操作将执行以下操作：

- 创建新的空文件系统。该文件系统将具有 Amazon EFS 分配的 ID，并且初始生命周期状态为 `creating`。
- 返回所创建文件系统的描述。

否则，此操作将返回 `FileSystemAlreadyExists` 错误，其中包含现有文件系统的 ID。

Note

对于基本使用案例，您可以对创建令牌使用随机生成的 UUID。

幂等操作允许您重试 `CreateFileSystem` 调用，而不会有创建额外文件系统的风险。当初始调用以某种方式失败时，可能会发生这种情况，使其无法确定是否已实际创建文件系统。例如，可能发生了传输级别超时或重置了连接。只要使用相同的创建令牌，如果初始调用已成功创建文件系统，则客户端可以从 `FileSystemAlreadyExists` 错误中了解文件系统的存在。

有关更多信息，请参阅《Amazon EFS 用户指南》中的[创建文件系统](#)。

Note

当文件系统的生命周期状态仍为 `creating` 时，`CreateFileSystem` 调用将返回。可以通过调用 [DescribeFileSystems](#) 操作来检查文件系统创建状态，此操作将返回文件系统状态以及其他方面的信息。

此操作接受为文件系统选择的可选 `PerformanceMode` 参数。我们建议对所有文件系统使用 `generalPurpose` `PerformanceMode`。`maxIO` 模式是上一代性能类型，专为高度并行化的工作负载而设计，可以比 `generalPurpose` 模式容忍更高的延迟。单区文件系统或使用弹性吞吐量的文件系统不支持 `MaxIO` 模式。

创建文件系统后，将无法更改 `PerformanceMode`。有关更多信息，请参阅[Amazon EFS 性能模式](#)。

可以使用 `ThroughputMode` 参数设置文件系统的吞吐量模式。

在完全创建文件系统之后，Amazon EFS 将其生命周期状态设置为 `available`，此时您可以在 VPC 中为文件系统创建一个或多个挂载目标。有关更多信息，请参阅 [CreateMountTarget](#)。您可以使用挂载目标将 Amazon EFS 文件系统挂载到您的 VPC 中的 EC2 实例上。有关更多信息，请参阅 [Amazon EFS：工作原理](#)。

此操作需要 `elasticfilesystem:CreateFileSystem` 操作的权限。

可以在创建时对文件系统进行标记。如果在创建操作中指定了标签，则 IAM 会对 `elasticfilesystem:TagResource` 操作执行额外的授权，以验证用户是否具备创建标签的权限。因此，您必须授予使用 `elasticfilesystem:TagResource` 操作的显式权限。有关更多信息，请参阅 [在创建过程中授予标记资源的权限](#)。

请求语法

```
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"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

AvailabilityZoneName

对于单区文件系统，指定要在其中创建文件系统的 AWS 可用区。使用格式 `us-east-1a` 指定可用区。有关单区文件系统的更多信息，请参阅《Amazon EFS 用户指南》中的 [EFS 文件系统类型](#)。

Note

一个区域文件系统并非在所有可用 Amazon EFS AWS 区域 的可用区域中都可用。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：. +

必需：否

Backup

指定是否在正在创建的文件系统上启用自动备份。将此值设置为 `true` 以启用自动备份。如果您创建单区文件系统，则默认情况下会启用自动备份。有关更多信息，请参阅《Amazon EFS 用户指南》中的 [自动备份](#)。

默认值为 `false`。但是，如果指定 `AvailabilityZoneName`，则默认为 `true`。

Note

AWS Backup 并非在所有提供 Amazon EFS AWS 区域 的地方都可用。

类型：布尔值

必需：否

CreationToken

长度最多为 64 个 ASCII 字符的字符串。Amazon EFS 使用它来确保幂等性创建。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：.+

是否必需：是

Encrypted

一个布尔值，如果为 true，则创建一个加密文件系统。创建加密文件系统时，您可以选择指定现有 AWS Key Management Service 密钥（KMS 密钥）。如果您不指定 KMS 密钥，则使用 Amazon EFS 的默认 KMS 密钥（即 /aws/elasticfilesystem）来保护加密文件系统。

类型：布尔值

必需：否

KmsKeyId

要用于保护加密文件系统的 KMS 密钥的 ID。仅当希望使用非默认 KMS 密钥时，此参数才是必需的。如果未指定此参数，则使用 Amazon EFS 的默认 KMS 密钥。可使用以下格式指定 KMS 密钥 ID。

- 键 ID - 键的唯一标识符，例如 1234abcd-12ab-34cd-56ef-1234567890ab。
- ARN - 键的 Amazon 资源名称（ARN），例如 arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab。
- 键别名 - 之前为键创建的显示名称，例如 alias/projectKey1。
- 键别名 ARN - 键别名的 ARN，例如 arn:aws:kms:us-west-2:444455556666:alias/projectKey1。

如果使用 KmsKeyId，则必须将 encrypted [CreateFileSystem](#) 参数设置为 true。

Important

EFS 仅接受对称 KMS 密钥。不能在 Amazon EFS 文件系统上使用非对称 KMS 密钥。

类型：字符串

长度约束：最大长度为 2048。

模式：`^([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/_-]+))))$
```

必需：否

PerformanceMode

文件系统的性能模式。我们针对所有文件系统推荐使用 generalPurpose 性能模式。使用 maxIO 性能模式的文件系统可以扩展到更高级别的聚合吞吐量和每秒操作数，但代价是大多数文件操作的延迟较高。创建文件系统后，将无法更改性能模式。单区文件系统不支持 maxIO 模式。

Important

由于最大 I/O 的每次操作延迟较高，因此我们建议对所有文件系统使用通用性能模式。

默认值为 generalPurpose。

类型：字符串

有效值：generalPurpose | maxIO

必需：否

ProvisionedThroughputInMibps

您要为正在创建的文件系统预配置的吞吐量，以兆字节每秒 (MiBps) 为单位。如果将 ThroughputMode 设置为 provisioned，则是必需的。有效值为 1-3414 MiBps，上限视区域而定。要提高此限制，请联系支持。有关更多信息，请参阅《Amazon EFS 用户指南》中的[您可以提高的 Amazon EFS 配额](#)。

类型：双精度

有效范围：最小值为 1.0。

必需：否

Tags

用于创建与文件系统关联的一个或多个标签。每个标签均为一个用户定义的键值对。通过包含 "Key": "Name", "Value": "{value}" 键值对来在创建时为文件系统命名。每个键必须是唯一的。有关更多信息，请参阅《AWS 通用参考指南》中的为[AWS 资源添加标签](#)。

类型：[Tag](#) 对象数组

必需：否

[ThroughputMode](#)

指定文件系统的吞吐量模式。该模式可以是 bursting、provisioned 或 elastic。如果将 ThroughputMode 设置为 provisioned，则还必须设置 ProvisionedThroughputInMibps 的值。创建文件系统之后，您可以降低文件系统的预调配吞吐量，或者在吞吐量模式之间切换，但存在一定的时间限制。有关更多信息，请参阅《Amazon EFS 用户指南》中的[通过预置模式指定吞吐量](#)。

默认值为 bursting。

类型：字符串

有效值：bursting | provisioned | elastic

必需：否

响应语法

```
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"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 201 响应。

服务以 JSON 格式返回的以下数据。

AvailabilityZoneId

文件系统所在可用区的唯一且一致的标识符，仅对单区域文件系统有效。例如，use1-az1是 us-east AWS 区域-1 的可用区 ID，它在每个可用区中的位置都相同。AWS 账户

类型：字符串

AvailabilityZoneName

描述文件系统所在的 AWS 可用区，并且仅对单区域文件系统有效。有关更多信息，请参阅《Amazon EFS 用户指南》中的[使用 EFS 存储类](#)。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：.+

CreationTime

文件系统的创建时间，以秒为单位（自 1970-01-01T00:00:00Z 起）。

类型：时间戳

[CreationToken](#)

请求中指定的不透明字符串。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：.+

[Encrypted](#)

一个布尔值，如果设为 true，则指示文件系统已加密。

类型：布尔值

[FileSystemArn](#)

EFS 文件系统的 Amazon 资源名称 (ARN)，采用 `arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id` 格式。使用示例数据的示例：`arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567`

类型：字符串

[FileSystemId](#)

文件系统 ID，由 Amazon EFS 分配。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[FileSystemProtection](#)

描述文件系统上的保护功能。

类型：[FileSystemProtectionDescription](#) 对象

[KmsKeyId](#)

AWS KMS key 用于保护加密文件系统的 ID。

类型：字符串

长度约束：最大长度为 2048。

模式：`^([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

文件系统的生命周期阶段。

类型：字符串

有效值：`creating | available | updating | deleting | deleted | error`

Name

可以向文件系统添加标签，包括 Name 标签。有关更多信息，请参阅 [CreateFileSystem](#)。如果文件系统有 Name 标签，Amazon EFS 会返回此字段中的值。

类型：字符串

长度约束：最大长度为 256。

模式：`^([\p{L}\p{Z}\p{N}_.:\/=+\-@]*)$`

NumberOfMountTargets

文件系统当前的挂载目标数。有关更多信息，请参阅 [CreateMountTarget](#)。

类型：整数

有效范围：最小值为 0。

OwnerId

AWS 账户 创建文件系统的。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

[PerformanceMode](#)

文件系统的性能模式。

类型：字符串

有效值：generalPurpose | maxIO

[ProvisionedThroughputInMibps](#)

文件系统的预配置吞吐量（以 MiBps 衡量单位）。对使用将 ThroughputMode 设置为 provisioned 的文件系统有效。

类型：双精度

有效范围：最小值为 1.0。

[SizeInBytes](#)

文件系统中存储的数据的最新已知计量大小（以字节为单位）（在其 Value 字段中），以及确定该大小的时间（在其 Timestamp 字段中）。Timestamp 值是自 1970-01-01T00:00:00Z 以来的整数秒数。SizeInBytes 值并不代表文件系统一致快照的大小，但是当没有写入文件系统时，该值最终会保持一致。也就是说，只有在超过几个小时的时间内未修改文件系统时，SizeInBytes 才表示实际大小。否则，该值不是文件系统在任何时间点的确切大小。

类型：[FileSystemSize](#) 对象

[Tags](#)

与文件系统关联的标签，以 Tag 对象数组形式呈现。

类型：[Tag](#) 对象数组

[ThroughputMode](#)

显示文件系统的吞吐量模式。有关更多信息，请参阅《Amazon EFS 用户指南》中的[吞吐量模式](#)。

类型：字符串

有效值：bursting | provisioned | elastic

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemAlreadyExists

如果尝试创建的文件系统已存在，且具有所提供的创建令牌，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

FileSystemLimitExceeded

如果 AWS 账户 已经创建了每个账户允许的最大文件系统数，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：403

InsufficientThroughputCapacity

如果没有足够的容量来预置额外的吞吐量，则返回此内容。尝试在预配置吞吐量模式下创建文件系统，尝试增加现有文件系统的预配置吞吐量，或尝试将现有文件系统从突增吞吐量模式更改为预配置吞吐量模式时，可能会返回此值。请稍后重试。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：503

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ThroughputLimitExceeded

如果由于已达到 1024 的吞吐量限制而无法更改吞吐量模式或预配置吞吐量 MiB/s，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

UnsupportedAvailabilityZone

如果请求的 Amazon EFS 功能在指定的可用区中不可用，则返回此值。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

示例

创建加密 EFS 文件系统

以下示例发送 POST 请求，要求在启用自动备份的 us-west-2 区域中创建文件系统。该请求指定 myFileSystem1 为幂等性的创建令牌。

示例请求

```
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"
  }
]
```

示例响应

```
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"
}
```

创建具有单区可用性的加密 EFS 文件系统

以下示例发送 POST 请求，要求在启用自动备份的 us-west-2 区域中创建文件系统。该文件系统在 us-west-2b 可用区中将具有单区存储。

示例请求

```
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"
    }
  ]
}
```

示例响应

```
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"
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

CreateMountTarget

创建文件系统的挂载目标。然后，您可以使用挂载目标将文件系统挂载到 EC2 实例上。有关挂载目标的更多信息，请参阅[管理挂载目标](#)。

要为 EFS 文件系统创建挂载目标，文件系统的生命周期状态必须为 `available`。有关更多信息，请参阅[DescribeFileSystems](#)。

创建挂载目标后，Amazon EFS 将返回一个响应，其中包括 `MountTargetId` 和一个 IP 地址（`IpAddress` 或 `Ipv6Address`）。在 EC2 实例中装载文件系统时，您可以使用此 IP 地址。您还可以在挂载文件系统时使用挂载目标的 DNS 名称。使用挂载目标挂载文件系统的 EC2 实例可以将挂载目标的 DNS 名称解析为其 IP 地址。有关更多信息，请参阅[Amazon EFS 的工作原理](#)。

请注意，您只能在一个 VPC 中为文件系统创建挂载目标，而且每个可用区只能有一个挂载目标。有关更多信息，请参阅[创建挂载目标](#)。

如果请求满足要求，则 Amazon EFS 将执行以下操作：

- 在指定的子网中创建新的挂载目标。
- 在子网中创建新的网络接口，并包含以下配置：
 - 描述 Mount target *fsmt-id* for file system *fs-id*，其中 *fsmt-id* 是挂载目标 ID，*fs-id* 是 `FileSystemId`。
 - 将网络接口的 `requesterManaged` 属性设置为 `true`，并将 `requesterId` 值设置为 EFS。

每个挂载目标都有一个相应的请求者管理的 EC2 网络接口。创建网络接口后，Amazon EFS 会将挂载目标描述中的 `NetworkInterfaceId` 字段设置为网络接口 ID，并将 IP 地址设置为其地址。如果网络接口创建失败，则整个 `CreateMountTarget` 操作将失败。

Note

`CreateMountTarget` 调用仅在创建网络接口后返回，但在挂载目标状态仍为 `creating` 时，可以通过调用[DescribeMountTargets](#) 操作来检查挂载目标创建状态，除此之外，该操作还将返回挂载目标状态。

此操作需要对文件系统执行以下操作的权限：

- `elasticfilesystem:CreateMountTarget`

此操作还需要权限才能执行以下 Amazon EC2 操作：

- ec2:DescribeSubnets
- ec2:DescribeNetworkInterfaces
- ec2:CreateNetworkInterface

请求语法

```
POST /2015-02-01/mount-targets HTTP/1.1
Content-type: application/json
```

```
{
  "FileSystemId": "string",
  "IpAddress": "string",
  "IpAddressType": "string",
  "Ipv6Address": "string",
  "SecurityGroups": [ "string" ],
  "SubnetId": "string"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

FileSystemId

要为其创建挂载目标的文件系统的 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

是否必需：是

[IpAddress](#)

如果挂载目标IpAddressType的是 IPv4 (IPV4_ONLY或DUAL_STACK) , 则指定要使用 IPv4 的地址。如果没有指定 IpAddress , 则 Amazon EFS 会从为 SubnetId 指定的子网中选择未使用的 IP 地址。

类型 : 字符串

长度限制 : 最小长度为 7。最大长度为 15。

模式 : `^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

必需 : 否

[IpAddressType](#)

挂载目标的 IP 地址类型。可能的值是IPV4_ONLY (仅限 IPv4 地址)、IPV6_ONLY (仅限 IPv6 地址) 和DUAL_STACK (双堆栈 , 均 IPv4 为 IPv6 地址)。如果未指定 IpAddressType , 则使用 IPV4_ONLY。

Note

IpAddressType 必须与子网的 IP 类型相匹配。此外 , IpAddressType 参数会覆盖 VPC 中设置为子网默认 IP 地址的值。例如 , 如果 IpAddressType is IPV4_ONLY 和 AssignIpv6AddressOnCreation is true , IPv4 则用于挂载目标。有关更多信息 , 请参阅[修改子网的 IP 寻址属性](#)。

类型 : 字符串

有效值 : IPV4_ONLY | IPV6_ONLY | DUAL_STACK

必需 : 否

[Ipv6Address](#)

如果挂载目标IpAddressType的是 IPv6 (IPV6_ONLY或DUAL_STACK) , 则指定要使用 IPv6 的地址。如果没有指定 Ipv6Address , 则 Amazon EFS 会从为 SubnetId 指定的子网中选择未使用的 IP 地址。

类型 : 字符串

长度限制：最小长度为 3。最大长度为 39。

必需：否

SecurityGroups

VPC 安全组 IDs，形式为 `sg-xxxxxxxx`。它们必须用于与指定子网相同的 VPC。安全组的最大数量取决于账户配额。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [Amazon VPC 配额](#)（参阅安全组表）。如果未指定安全组，则 Amazon EFS 将对子网的 VPC 使用默认安全组。

类型：字符串数组

数组成员：最多 100 项。

长度限制：最小长度为 11。最大长度为 43。

模式：`^sg-[0-9a-f]{8,40}`

必需：否

SubnetId

要在其中添加挂载目标的子网的 ID。对于单区文件系统，请使用与文件系统的可用区关联的子网。子网类型必须与 `IpAddressType` 同样的类型。

类型：字符串

长度限制：最小长度为 15。最大长度为 47。

模式：`^subnet-[0-9a-f]{8,40}$`

必需：是

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "AvailabilityZoneId": "string",
  "AvailabilityZoneName": "string",
  "FileSystemId": "string",
  "IpAddress": "string",
```

```
"Ipv6Address": "string",  
"LifeCycleState": "string",  
"MountTargetId": "string",  
"NetworkInterfaceId": "string",  
"OwnerId": "string",  
"SubnetId": "string",  
"VpcId": "string"  
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[AvailabilityZoneId](#)

挂载目标所在可用区的唯一且一致的标识符。例如，use1-az1是 us-east-1 区域的可用区 ID，并且每个区域的位置都相同。AWS 账户

类型：字符串

[AvailabilityZoneName](#)

挂载目标所在的可用区名称。可用区独立映射到每个可用区的名称 AWS 账户。例如，您的可用区us-east-1a AWS 账户 可能与其他可用区不同 AWS 账户。us-east-1a

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：.+

[FileSystemId](#)

挂载目标要用于的文件系统 ID。

类型：字符串

长度限制：最大长度为 128。

模式：^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})\$

[IpAddress](#)

挂载目标 IPv4 的地址。

类型：字符串

长度限制：最小长度为 7。最大长度为 15。

模式：`^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

[Ipv6Address](#)

挂载目标 IPv6 的地址。

类型：字符串

长度限制：最小长度为 3。最大长度为 39。

[LifeCycleState](#)

挂载目标的生命周期状态。

类型：字符串

有效值：`creating` | `available` | `updating` | `deleting` | `deleted` | `error`

[MountTargetId](#)

系统分配的挂载目标 ID。

类型：字符串

长度限制：最小长度为 13。最大长度为 45。

模式：`^fsmt-[0-9a-f]{8,40}$`

[NetworkInterfaceId](#)

Amazon EFS 在创建挂载目标时创建的网络接口 ID。

类型：字符串

[OwnerId](#)

AWS 账户 拥有资源的 ID。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\d{12})|(\d{4}-\d{4}-\d{4})$`

SubnetId

挂载目标子网的 ID。

类型：字符串

长度限制：最小长度为 15。最大长度为 47。

模式：`^subnet-[0-9a-f]{8,40}$`

VpcId

在其中配置挂载目标的虚拟私有云 (VPC) ID。

类型：字符串

错误

AvailabilityZonesMismatch

如果为挂载目标指定的可用区与文件系统的可用区不同，将返回此错误。有关更多信息，请参阅[区域和单区存储冗余](#)。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifecycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

IpAddressInUse

如果请求指定了子网中已在使用的 IP 地址 (`IpAddress` 或 `Ipv6Address`)，将返回此错误。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

MountTargetConflict

如果挂载目标违反了基于文件系统现有挂载目标的指定限制之一，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

NetworkInterfaceLimitExceeded

调用账户已达到特定 AWS 区域弹性网络接口的限制。要么删除一些网络接口，要么请求提高账户配额。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [Amazon VPC 配额](#)（请参阅“网络接口”表中的“每个区域的网络接口”）。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

NoFreeAddressesInSubnet

如果请求中未指定 `IpAddress` 或 `Ipv6Address` 且指定的子网中没有空闲的 IP 地址，则返回此错误。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

SecurityGroupLimitExceeded

当请求中指定的 `SecurityGroups` 数量超过基于账户配额设定的限制时，将返回此错误。要么删除一些安全组，要么请求提高账户配额。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [Amazon VPC 配额](#)（参阅安全组表）。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

SecurityGroupNotFound

如果子网的虚拟私有云 (VPC) 中不存在指定的安全组，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

SubnetNotFound

如果请求中没有提供 ID 为 SubnetId 的子网，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

UnsupportedAvailabilityZone

如果请求的 Amazon EFS 功能在指定的可用区中不可用，则返回此值。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

示例

在子网的可用 IPv4 地址上创建挂载目标

以下请求仅指定挂载目标的文件系统和子网。目标是在指定子网的可用 IPv4 地址上创建的，默认安全组与 VPC 关联。

示例请求

```
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"}
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252
```

```
{
  "OwnerId": "231243201240",
  "MountTargetId": "fsmt-55a4413c",
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-01234567",
  "LifecycleState": "available",
  "IpAddress": "172.31.22.183"
  "NetworkInterfaceId": "eni-1bcb7772"
  "AvailabilityZoneId": "eus1-az2",
  "AvailabilityZoneName": "eu-south-1b",
  "VpcId": "vpc-08d45b31fa009a15e"
}
```

在特定 IPv4 地址创建挂载目标

以下请求指定要用于挂载目标的文件系统、子网、安全组 and IPv4地址。

示例请求

```
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"
  ]
}
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
```

```
"OwnerId": "251839141158",
"MountTargetId": "fsmt-9a13661e",
"FileSystemId": "fs-01234567",
"SubnetId": "subnet-fd04ff94",
"LifeCycleState": "available",
"IpAddress": "10.0.2.42",
"NetworkInterfaceId": "eni-1bcb7772"
"AvailabilityZoneId": "eus1-az2",
"AvailabilityZoneName": "eu-south-1b",
"VpcId": "vpc-08d45b31fa009a15e"
}
```

在特定 IPv6 地址创建挂载目标

以下请求指定要用于挂载目标的文件系统、子网、安全组 and IPv6 地址。

示例请求

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-01234567",
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "IpAddressType": "IPV6_ONLY",
  "SecurityGroups": [
    "sg-01234567"
  ]
}
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
```

```
"OwnerId": "251839141158",
"MountTargetId": "fsmt-9a13661e",
"FileSystemId": "fs-01234567",
"SubnetId": "subnet-fd04ff94",
"LifeCycleState": "available",
"Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
"NetworkInterfaceId": "eni-1bcb7772"
"AvailabilityZoneId": "eus1-az2",
"AvailabilityZoneName": "eu-south-1b",
"VpcId": "vpc-08d45b31fa009a15e"
}
```

在双栈子网的可用 IPv4 地址上创建挂载目标

以下请求为挂载目标指定了文件系统、子网和地址类型。目标是在指定双栈子网上的可用 IPv4 地址创建的，默认安全组与 VPC 关联。

示例请求

```
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"}
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
  "OwnerId": "251839141158",
  "MountTargetId": "fsmt-55a4413c",
  "FileSystemId": "fs-01234567",
  "SubnetId": "subnet-fd04ff94",
  "LifeCycleState": "available",
  "IpAddress": "172.31.22.183"
  "Ipv6Address": "2a05:d01a:419:8611:3e57:75ab:5719:b238",
}
```



```
"NetworkInterfaceId": "eni-01234567",  
"AvailabilityZoneId": "eus1-az2",  
"AvailabilityZoneName": "eu-south-1b",  
"VpcId": "vpc-08d45b31fa009a15e"  
}
```

在特定 IPv4 上创建挂载目标，在双栈子 IPv6 网上创建地址

以下请求为挂载目标指定文件系统、子网、安全组、IPv4 IPv6 地址和地址。目标是在双栈子网上的指定 IPv4 IPv6地址创建的。

示例请求

```
POST /2015-02-01/mount-targets HTTP/1.1  
Host: elasticfilesystem.us-west-2.amazonaws.com  
x-amz-date: 20140620T221118Z  
Authorization: <...>  
Content-Type: application/json  
Content-Length: 160  
  
{  
  "FileSystemId": "fs-01234567",  
  "SubnetId": "subnet-01234567",  
  "IpAddressType": "DUAL_STACK",  
  "IpAddress": "10.0.1.25",  
  "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",  
  "SecurityGroups": [  
    "sg-01234567"  
  ]  
}
```

示例响应

```
HTTP/1.1 200 OK  
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef  
Content-Type: application/json  
Content-Length: 252  
  
{  
  "OwnerId": "231243201240",  
  "MountTargetId": "fsmt-55a4413c",  
  "FileSystemId": "fs-01234567",  
  "SubnetId": "subnet-01234567",  
}
```

```
"LifecycleState": "available",  
"IpAddress": "10.0.1.25",  
"Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",  
"NetworkInterfaceId": "eni-01234567",  
"AvailabilityZoneId": "eus1-az2",  
"AvailabilityZoneName": "eu-south-1b",  
"VpcId": "vpc-08d45b31fa009a15e"  
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

CreateReplicationConfiguration

为新的或现有的 EFS 文件系统创建复制配置。有关更多信息，请参阅《Amazon EFS 用户指南》中的 [Amazon EFS 复制](#)。此复制配置指定以下内容：

- 源文件系统 - 要复制的 EFS 文件系统。
- 目标文件系统 - 要将源文件系统复制到的目标文件系统。复制配置中只能有一个目标文件系统。

Note

一个文件系统只能属于一个复制配置。

复制配置的目标参数取决于您是要复制到新文件系统还是复制到现有文件系统，以及是否跨 AWS 账户复制。请参阅 [DestinationToCreate](#) 了解更多信息。

此操作需要 `elasticfilesystem:CreateReplicationConfiguration` 操作的权限。此外，还需要其他权限，具体取决于您复制文件系统的方式。有关更多信息，请参阅《Amazon EFS 用户指南》中的 [执行复制时所需的权限](#)。

请求语法

```
POST /2015-02-01/file-systems/SourceFileSystemId/replication-configuration HTTP/1.1
Content-type: application/json
```

```
{
  "Destinations": [
    {
      "AvailabilityZoneName": "string",
      "FileSystemId": "string",
      "KmsKeyId": "string",
      "Region": "string",
      "RoleArn": "string"
    }
  ]
}
```

URI 请求参数

请求使用以下 URI 参数。

[SourceFileSystemId](#)

指定要复制的 Amazon EFS 文件系统。此文件系统不能已经是另一个复制配置中的源文件系统或目标文件系统。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

[Destinations](#)

目标配置对象的数组。仅支持一个目标配置对象。

类型：[DestinationToCreate](#) 对象数组

是否必需：是

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "CreationTime": number,
  "Destinations": [
    {
      "FileSystemId": "string",
      "LastReplicatedTimestamp": number,
      "OwnerId": "string",
      "Region": "string",
      "RoleArn": "string",
      "Status": "string",
      "StatusMessage": "string"
    }
  ],
}
```

```
"OriginalSourceFileSystemArn": "string",  
"SourceFileSystemArn": "string",  
"SourceFileSystemId": "string",  
"SourceFileSystemOwnerId": "string",  
"SourceFileSystemRegion": "string"  
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

CreationTime

描述复制配置的创建时间。

类型：时间戳

Destinations

目标对象的数组。仅支持一个目标对象。

类型：[Destination](#) 对象数组

OriginalSourceFileSystemArn

复制配置中原始源 EFS 文件系统的 Amazon 资源名称 (ARN)。

类型：字符串

SourceFileSystemArn

复制配置中当前源文件系统的 Amazon 资源名称 (ARN)。

类型：字符串

SourceFileSystemId

正在复制的源 Amazon EFS 文件系统 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

SourceFileSystemOwnerId

源文件系统所在的 ID。AWS 账户

类型：字符串

长度限制：最大长度为 14。

模式：`^(\d{12})|(\d{4}-\d{4}-\d{4})$`

SourceFileSystemRegion

源 EFS 文件系统所在的。AWS 区域

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：`^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

ConflictException

如果复制中的源文件系统已加密，但目标文件系统未加密，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

FileSystemLimitExceeded

如果 AWS 账户 已经创建了每个账户允许的最大文件系统数，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：403

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifeCycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InsufficientThroughputCapacity

如果没有足够的容量来预置额外的吞吐量，则返回此内容。尝试在预配置吞吐量模式下创建文件系统，尝试增加现有文件系统的预配置吞吐量，或尝试将现有文件系统从突增吞吐量模式更改为预配置吞吐量模式时，可能会返回此值。请稍后重试。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：503

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ReplicationNotFound

如果指定的文件系统没有复制配置，则返回此内容。

ErrorCode

ReplicationNotFound

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

ThroughputLimitExceeded

如果由于已达到 1024 的吞吐量限制而无法更改吞吐量模式或预配置吞吐量 MiB/s，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

UnsupportedAvailabilityZone

如果请求的 Amazon EFS 功能在指定的可用区中不可用，则返回此值。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

ValidationException

如果请求所在的 AWS Backup 服务不可用 AWS 区域，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

CreateTags

Note

已弃用 - 已弃用 CreateTags 且未维护。要为 EFS 资源创建标签，请使用 [TagResource](#) API 操作。

创建或覆盖与文件系统关联的标签。每个标签都是一个键-值对。如果请求中指定的标签键已存在于文件系统中，则此操作将使用请求中提供的值覆盖其值。如果将 Name 标签添加到文件系统，Amazon EFS 会在对 [DescribeFileSystems](#) 操作的响应中将其返回。

此操作需要 elasticfilesystem:CreateTags 操作权限。

请求语法

```
POST /2015-02-01/create-tags/FileSystemId HTTP/1.1
Content-type: application/json
```

```
{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

URI 请求参数

请求使用以下 URI 参数。

[FileSystemId](#)

要修改其标签的文件系统 ID (字符串)。此操作仅修改标签，不修改文件系统。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

[Tags](#)

要添加的 Tag 对象的数组。每个 Tag 对象都是一个键值对。

类型：[Tag](#) 对象数组

是否必需：是

响应语法

```
HTTP/1.1 204
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 204 响应。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)

- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DeleteAccessPoint

删除指定的访问点。删除完成后，新客户端将无法再连接到接入点。删除时连接到接入点的客户端将继续运行，直到它们终止连接。

此操作需要 `elasticfilesystem:DeleteAccessPoint` 操作的权限。

请求语法

```
DELETE /2015-02-01/access-points/AccessPointId HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

AccessPointId

要删除的接入点 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 204
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 204 响应。

错误

AccessPointNotFound

如果请求者的指定AccessPointId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DeleteFileSystem

删除文件系统，永久中止访问其内容。返回后，文件系统不再存在，无法访问已删除文件系统的任何内容。

在删除 EFS 文件系统之前，需要手动删除挂载到文件系统的挂载目标。此步骤是在您使用 AWS 控制台删除文件系统时为您执行的。

Note

无法删除属于 EFS 复制配置的文件系统。需要先删除复制配置。

无法删除正在使用的文件系统。也就是说，如果文件系统有任何挂载目标，必须先删除挂载目标。有关更多信息，请参阅[DescribeMountTargets](#)和[DeleteMountTarget](#)。

Note

当文件系统状态仍为 `deleting` 时，将返回 `DeleteFileSystem` 调用。可以通过调用 [DescribeFileSystems](#) 操作来检查文件系统的删除状态，该操作会返回账户中的文件系统列表。如果传递已删除文件系统的文件系统 ID 或创建令牌，[DescribeFileSystems](#) 会返回 `404 FileSystemNotFound` 错误。

此操作需要 `elasticfilesystem:DeleteFileSystem` 操作的权限。

请求语法

```
DELETE /2015-02-01/file-systems/FileSystemId HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

[FileSystemId](#)

要删除的文件系统 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 204
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 204 响应。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemInUse

如果文件系统有挂载目标，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

示例

删除文件系统

以下示例向 file-systems 端点 (elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems/fs-01234567) 发送 DELETE 请求，以删除 ID 为 fs-01234567 的文件系统。

示例请求

```
DELETE /2015-02-01/file-systems/fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T233021Z
Authorization: <...>
```

示例响应

```
HTTP/1.1 204 No Content
x-amzn-RequestId: a2d125b3-7ebd-4d6a-ab3d-5548630bff33
Content-Length: 0
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DeleteFileSystemPolicy

删除指定文件系统的 FileSystemPolicy。删除现有策略后，默认 FileSystemPolicy 即生效。有关默认文件系统策略的更多信息，请参阅[在 EFS 中使用基于资源的策略](#)。

此操作需要 elasticfilesystem:DeleteFileSystemPolicy 操作的权限。

请求语法

```
DELETE /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

FileSystemId

指定要删除 FileSystemPolicy 的 EFS 文件系统。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifecycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DeleteMountTarget

删除指定的挂载目标。

此操作通过使用要删除的挂载目标强制中断文件系统的任何挂载，这可能会中断使用这些挂载的实例或应用程序。为避免应用程序突然中断，如果可行，可以考虑卸载挂载目标的所有挂载。此操作还会删除关联的网络接口。未提交的写入可能会丢失，但是使用此操作中断挂载目标不会损坏文件系统本身。将保留所创建的文件系统。您可以使用另一个挂载目标在您的 VPC 中挂载 EC2 实例。

此操作需要对文件系统执行以下操作的权限：

- `elasticfilesystem:DeleteMountTarget`

Note

当挂载目标状态仍为 `deleting` 时，将返回 `DeleteMountTarget` 调用。可以通过调用 [DescribeMountTargets](#) 操作来检查挂载目标的删除情况，此操作会返回给定文件系统的挂载目标描述列表。

该操作还需要在挂载目标的网络接口上执行以下 Amazon EC2 操作的权限：

- `ec2:DeleteNetworkInterface`

请求语法

```
DELETE /2015-02-01/mount-targets/MountTargetId HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

MountTargetId

要删除的挂载目标 ID (字符串)。

长度限制：最小长度为 13。最大长度为 45。

模式：`^fsmt-[0-9a-f]{8,40}$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 204
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 204 响应。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

DependencyTimeout

服务在尝试完成请求时超时，客户端应重试调用。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：504

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

MountTargetNotFound

如果在调用者的 AWS 账户中未找到具有指定 ID 的挂载目标，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

示例

移除文件系统的挂载目标

以下示例发送 DELETE 请求以删除特定挂载目标。

示例请求

```
DELETE /2015-02-01/mount-targets/fsmt-9a13661e HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T232908Z
Authorization: <...>
```

示例响应

```
HTTP/1.1 204 No Content
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DeleteReplicationConfiguration

删除复制配置。删除复制配置会结束复制过程。删除复制配置后，目标文件系统变为 Writeable，其复制覆盖保护功能将重新启用。有关更多信息，请参阅[删除复制配置](#)。

此操作需要 `elasticfilesystem:DeleteReplicationConfiguration` 操作的权限。

请求语法

```
DELETE /2015-02-01/file-systems/SourceFileSystemId/replication-configuration?
deletionMode=DeletionMode HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

[DeletionMode](#)

默认情况下，跨 AWS 账户 或跨区域复制时 AWS 区域，Amazon EFS 会从源账户和目标账户或区域 (ALL_CONFIGURATIONS) 中删除复制配置。如果存在配置或权限问题，导致 Amazon EFS 无法同时从两端删除复制配置，则可以使用 LOCAL_CONFIGURATION_ONLY 模式仅从本地端（执行删除操作的账户或区域）删除配置。

Note

仅在 Amazon EFS 无法同时删除源和目标账户或区域中的复制配置时才使用 LOCAL_CONFIGURATION_ONLY 模式。删除本地配置会使其他账户或区域中的配置无法恢复。

此外，不要将此模式用于同账户、同区域的复制，因为这样做会导致 BadRequest 异常错误。

有效值：ALL_CONFIGURATIONS | LOCAL_CONFIGURATION_ONLY

[SourceFileSystemId](#)

复制配置中源文件系统的 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 204
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 204 响应。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ReplicationNotFound

如果指定的文件系统没有复制配置，则返回此内容。

ErrorCode

ReplicationNotFound

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DeleteTags

Note

已弃用 - 已弃用 DeleteTags 且未维护。要从 EFS 资源中删除标签，请使用 [UntagResource](#) API 操作。

从文件系统中删除指定的标签。如果 DeleteTags 请求中包含不存在的标签键，Amazon EFS 会忽略它并且不会导致错误。有关标签和相关限制的更多信息，请参阅《AWS 账单与成本管理 用户指南》中的[标签限制](#)。

此操作需要 elasticfilesystem:DeleteTags 操作的权限。

请求语法

```
POST /2015-02-01/delete-tags/FileSystemId HTTP/1.1
Content-type: application/json

{
  "TagKeys": [ "string" ]
}
```

URI 请求参数

请求使用以下 URI 参数。

[FileSystemId](#)

要删除其标签的文件系统 ID (字符串)。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

TagKeys

要删除的标签键列表。

类型：字符串数组

数组成员：最少 1 个物品。最多 50 项。

长度限制：最小长度为 1。最大长度为 128。

模式：`^(?!([aA]{1}[wW]{1}[sS]{1}:)([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]+))$`

必需：是

响应语法

```
HTTP/1.1 204
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 204 响应。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)

- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeAccessPoints

返回特定 Amazon EFS 接入点的描述 (如果提供 `AccessPointId`)。如果提供 `EFS FileSystemId`，它将返回该文件系统的所有接入点描述。可以在请求中提供 `AccessPointId` 或 `FileSystemId`，但不能同时提供两者。

此操作需要 `elasticfilesystem:DescribeAccessPoints` 操作的权限。

请求语法

```
GET /2015-02-01/access-points?  
AccessPointId=AccessPointId&FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken  
HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

AccessPointId

(可选) 指定要在响应中描述的 EFS 接入点；与 `FileSystemId` 互斥。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

FileSystemId

(可选) 如果提供了 `FileSystemId`，EFS 将返回该文件系统的所有接入点；与 `AccessPointId` 互斥。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

MaxResults

(可选) 检索文件系统的所有接入点时，可以选择指定 `MaxItems` 参数来限制响应中返回的对象数量。默认值是 100。

有效范围：最小值为 1。

NextToken

如果对响应进行分页，则显示 NextToken。可以在后续请求中使用 NextMarker 提取下一页接入点描述。

长度限制：最小长度为 1。最大长度为 128。

模式：`.+`

请求正文

该请求没有请求正文。

响应语法

```
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,
          "OwnerId": number,
          "Permissions": "string"
        },
        "Path": "string"
      },
      "Tags": [
```

```
{
  {
    "Key": "string",
    "Value": "string"
  }
],
"NextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[AccessPoints](#)

一系列接入点描述。

类型：[AccessPointDescription](#) 对象数组

[NextToken](#)

如果接入点数量超过响应中返回的数量，则显示。您可以在后续请求 NextMarker 中使用来获取其他描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：`.+`

错误

AccessPointNotFound

如果请求者的指定 AccessPointId 值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeAccountPreferences

返回当前与发出请求的用户 AWS 账户 关联的账户偏好设置 AWS 区域。

请求语法

```
GET /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

MaxResults

(可选) 检索账户首选项时，可以选择指定 MaxItems 参数以限制响应中返回的对象数量。默认值是 100。

类型：整数

有效范围：最小值为 1。

必需：否

NextToken

(可选) 如果响应负载已分页，则可以在后续请求中使用 NextToken 获取下一页 AWS 账户 首选项。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

必需：否

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "ResourceIdPreference": {
    "ResourceIdType": "string",
    "Resources": [ "string" ]
  }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[NextToken](#)

如果存在的记录多于响应中返回的记录，则显示。可以在随后的请求中使用 NextToken 来获取其他描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

[ResourceIdPreference](#)

描述当前与发出请求的用户 AWS 账户 关联的资源 ID 首选项设置 AWS 区域。

类型：[ResourceIdPreference](#) 对象

错误

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeBackupPolicy

返回指定 EFS 文件系统的备份策略。

请求语法

```
GET /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

[FileSystemId](#)

指定要检索其 BackupPolicy 的 EFS 文件系统。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json
```

```
{
  "BackupPolicy": {
    "Status": "string"
  }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[BackupPolicy](#)

描述文件系统的备份策略，指明是开启还是关闭自动备份。

类型：[BackupPolicy](#) 对象

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

PolicyNotFound

在无法找到或访问指定的 IAM 策略或资源策略时返回此错误。根据正在执行的操作，如果默认文件系统策略对 EFS 文件系统有效，或者如果未指定备份策略，也可能返回此错误。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

ValidationException

如果请求所在的 AWS Backup 服务不可用 AWS 区域，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeFileSystemPolicy

返回指定 EFS 文件系统的 FileSystemPolicy。

此操作需要 elasticfilesystem:DescribeFileSystemPolicy 操作的权限。

请求语法

```
GET /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

FileSystemId

指定要检索 FileSystemPolicy 的 EFS 文件系统。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "FileSystemId": "string",
  "Policy": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[FileSystemId](#)

指定 FileSystemPolicy 适用的 EFS 文件系统。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[Policy](#)

EFS 文件系统的 JSON 格式的 FileSystemPolicy。

类型：字符串

长度限制：最小长度为 1。最大长度为 20000。

模式：`[\s\S]+`

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定 FileSystemId 值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

PolicyNotFound

在无法找到或访问指定的 IAM 策略或资源策略时返回此错误。根据正在执行的操作，如果默认文件系统策略对 EFS 文件系统有效，或者如果未指定备份策略，也可能返回此错误。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

示例

示例

此示例说明了的一种用法 DescribeFileSystemPolicy。

示例请求

```
GET /2015-02-01/file-systems/fs-01234567/policy HTTP/1.1
```

示例响应

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

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeFileSystems

返回特定 Amazon EFS 文件系统的描述 (如果提供了文件系统 `CreationToken` 或 `FileSystemId`)。否则, 它将返回您正在调用的端点 AWS 账户 中调用方拥有的所有文件系统的描述。AWS 区域

检索所有文件系统描述时, 可以选择指定 `MaxItems` 参数来限制响应中的描述数量。此数字自动设置为 100。如果还有更多文件系统描述, Amazon EFS 将在响应中返回一个不透明的令牌 `NextMarker`。在这种情况下, 应发送一个后续请求, 并将 `Marker` 请求参数的值设置为 `NextMarker`。

要检索文件系统描述列表, 可以在迭代过程中使用此操作, 在此过程中, 首先在没有 `Marker` 的情况下调用 `DescribeFileSystems`, 然后操作继续调用此内容, 并将 `Marker` 参数设置为前一响应中的 `NextMarker` 值, 直到响应没有 `NextMarker` 为止。

未指定在一次 `DescribeFileSystems` 调用的响应中返回的文件系统顺序, 以及多调用迭代响应中返回的文件系统顺序。

此操作需要 `elasticfilesystem:DescribeFileSystems` 操作的权限。

请求语法

```
GET /2015-02-01/file-systems?
CreationToken=CreationToken&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems
HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

CreationToken

(可选) 使用此创建令牌 (字符串) 将列表限制为文件系统。创建 Amazon EFS 文件系统时, 需要指定创建令牌。

长度限制: 最小长度为 1。长度上限为 64。

模式: `.+`

FileSystemId

(可选) 要检索其描述的文件系统 ID (字符串)。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Marker

(可选) 从以前的 `DescribeFileSystems` 操作返回的不透明分页标记 (字符串)。如果存在，则指定从返回调用中断的位置继续显示列表。

长度限制：最小长度为 1。最大长度为 128。

模式：`.+`

MaxItems

(可选) 指定要在响应中返回的最大文件系统数 (整数)。此数字自动设置为 100。如果文件系统超过 100 个，则响应按每页 100 项进行分页。

有效范围：最小值为 1。

请求正文

该请求没有请求正文。

响应语法

```
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"
}

```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[FileSystems](#)

一组文件系统描述。

类型：[FileSystemDescription](#) 对象数组

[Marker](#)

如果请求中的调用者提供，则显示（字符串）。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

[NextMarker](#)

如果文件系统数量超过响应中返回的文件系统数量（字符串），则显示。可以在随后的请求中使用 NextMarker 来获取这些描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述（英语）。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定 FileSystemId 值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述（英语）。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

示例

检索包含 10 个文件系统的列表

以下示例向 file-systems 端点 (elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems) 发送 GET 请求。该请求指定一个 MaxItems 查询参数，将文件系统描述的数量限制为 10 个。

示例请求

```
GET /2015-02-01/file-systems?MaxItems=10 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191208Z
Authorization: <...>
```

示例响应

```
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
    }
  ]
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeLifecycleConfiguration

返回指定 EFS 文件系统的当前 LifecycleConfiguration 对象。生命周期管理使用 LifecycleConfiguration 对象来确定何时在存储类之间移动文件。对于没有 LifecycleConfiguration 对象的文件系统，该调用将在响应中返回一个空数组。

此操作需要 elasticfilesystem:DescribeLifecycleConfiguration 操作的权限。

请求语法

```
GET /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

FileSystemId

要检索其 LifecycleConfiguration 对象的文件系统 ID (字符串)。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "LifecyclePolicies": [
    {
```

```
    "TransitionToArchive": "string",  
    "TransitionToIA": "string",  
    "TransitionToPrimaryStorageClass": "string"  
  }  
]  
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[LifecyclePolicies](#)

一系列生命周期管理策略。EFS 支持每个文件系统最多一个策略。

类型：[LifecyclePolicy](#) 对象数组

数组成员：最多 3 项。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

示例

检索文件系统的生命周期配置

以下请求检索指定文件系统的 LifecycleConfiguration 对象。

示例请求

```
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: <...>
```

示例响应

```
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"
    }
  ]
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeMountTargets

返回文件系统的所有当前挂载目标或特定挂载目标的描述。请求所有当前挂载目标时，未指定响应中返回的挂载目标顺序。

此操作需要对您在 `FileSystemId` 中指定的文件系统 ID，或您在 `MountTargetId` 中指定的挂载目标的文件系统拥有 `elasticfilesystem:DescribeMountTargets` 操作权限。

对应的 CLI 命令为 `describe-mount-targets`。

请求语法

```
GET /2015-02-01/mount-targets?
AccessPointId=AccessPointId&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems&MountTargetId=MountTargetId
HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

[AccessPointId](#)

(可选) 要列出其挂载目标的接入点 ID。如果请求中未包含 `FileSystemId` 或 `MountTargetId`，则必须将其包含在请求中。接受接入点 ID 或 ARN 作为输入。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

[FileSystemId](#)

(可选) 要列出其挂载目标的文件系统 ID (字符串)。如果请求中未包含 `AccessPointId` 或 `MountTargetId`，则必须将其包含在请求中。接受文件系统 ID 或 ARN 作为输入。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[Marker](#)

(可选) 从以前的 `DescribeMountTargets` 操作返回的不透明分页标记 (字符串)。如果存在，则它指定从上一个返回的调用中断的位置继续列表。

长度限制：最小长度为 1。最大长度为 128。

模式：.+

MaxItems

(可选) 要在响应中返回的挂载目标的最大数量。当前，此数字自动设置为 10，其他值被忽略。如果挂载目标超过 100 个，则响应按每页 100 项进行分页。

有效范围：最小值为 1。

MountTargetId

(可选) 您要描述的挂载目标 ID (字符串)。如果请求中未包含 FileSystemId，则必须将其包含在请求中。接受挂载目标 ID 或 ARN 作为输入。

长度限制：最小长度为 13。最大长度为 45。

模式：`^fsmt-[0-9a-f]{8,40}$`

请求正文

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "Marker": "string",
  "MountTargets": [
    {
      "AvailabilityZoneId": "string",
      "AvailabilityZoneName": "string",
      "FileSystemId": "string",
      "IpAddress": "string",
      "Ipv6Address": "string",
      "LifeCycleState": "string",
      "MountTargetId": "string",
      "NetworkInterfaceId": "string",
      "OwnerId": "string",
      "SubnetId": "string",
    }
  ]
}
```

```
    "VpcId": "string"
  },
  "NextMarker": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[Marker](#)

如果请求包含 Marker，则响应会在此字段中返回该值。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

[MountTargets](#)

以 MountTargetDescription 对象数组的形式返回文件系统的挂载目标。

类型：[MountTargetDescription](#) 对象数组

[NextMarker](#)

如果存在值，则会返回更多挂载目标。在后续请求中，可以在请求中提供具有此值的 Marker 以检索下一组挂载目标。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

错误

AccessPointNotFound

如果请求者的指定AccessPointId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

MountTargetNotFound

如果在调用者的 AWS 账户中未找到具有指定 ID 的挂载目标，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

示例

检索为文件系统创建的挂载目标的描述

以下请求检索为指定文件系统创建的挂载目标的描述。

示例请求

```
GET /2015-02-01/mount-targets?FileSystemId=fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191252Z
```

Authorization: <...>

示例响应

```
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"
    }
  ]
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeMountTargetSecurityGroups

返回挂载目标的当前有效的安全组。此操作需要已创建挂载目标的网络接口，并且挂载目标的生命周期状态不是 `deleted`。

此操作需要以下操作的权限：

- 对挂载目标的文件系统执行 `elasticfilesystem:DescribeMountTargetSecurityGroups` 操作。
- 在挂载目标的网络接口上执行 `ec2:DescribeNetworkInterfaceAttribute` 操作。

请求语法

```
GET /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

MountTargetId

要检索其安全组的挂载目标 ID。

长度限制：最小长度为 13。最大长度为 45。

模式：`^fsmt-[0-9a-f]{8,40}$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
```

```
"SecurityGroups": [ "string" ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

SecurityGroups

一组安全组。

类型：字符串数组

数组成员：最多 100 项。

长度限制：最小长度为 11。最大长度为 43。

模式：`^sg-[0-9a-f]{8,40}`

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

IncorrectMountTargetState

如果挂载目标未处于执行操作的正确状态，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

MountTargetNotFound

如果在调用者的 AWS 账户中未找到具有指定 ID 的挂载目标，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

示例

检索文件系统的有效安全组

以下示例检索对与挂载目标关联的网络接口有效的安全组。

示例请求

```
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: <...>
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Length: 57

{
  "SecurityGroups" : [
    "sg-188d9f74"
  ]
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版 SDK](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)

- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeReplicationConfigurations

检索特定文件系统的复制配置。如果未指定文件系统，则会检索 AWS 账户 中的所有复制配置。AWS 区域

请求语法

```
GET /2015-02-01/file-systems/replication-configurations?  
FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

[FileSystemId](#)

可以通过提供特定文件系统的文件系统 ID 来检索其复制配置。对于跨账户、跨区域复制，账户只能描述自己所在区域中文件系统的复制配置。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

[MaxResults](#)

(可选) 要限制响应中返回的对象数，可以指定 `MaxItems` 参数。默认值是 100。

有效范围：最小值为 1。

[NextToken](#)

如果对响应进行分页，则显示 `NextToken`。可以在后续请求中使用 `NextToken` 获取下一页输出。

长度限制：最小长度为 1。最大长度为 128。

模式：`.+`

请求正文

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "Replications": [
    {
      "CreationTime": number,
      "Destinations": [
        {
          "FileSystemId": "string",
          "LastReplicatedTimestamp": number,
          "OwnerId": "string",
          "Region": "string",
          "RoleArn": "string",
          "Status": "string",
          "StatusMessage": "string"
        }
      ],
      "OriginalSourceFileSystemArn": "string",
      "SourceFileSystemArn": "string",
      "SourceFileSystemId": "string",
      "SourceFileSystemOwnerId": "string",
      "SourceFileSystemRegion": "string"
    }
  ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[NextToken](#)

可以在后续请求中使用上一响应中的 NextToken 来获取其他描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

Replications

返回的复制配置集合。

类型：[ReplicationConfigurationDescription](#) 对象数组

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ReplicationNotFound

如果指定的文件系统没有复制配置，则返回此内容。

ErrorCode

ReplicationNotFound

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

ValidationException

如果请求所在的 AWS Backup 服务不可用 AWS 区域，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

DescribeTags

Note

已弃用 - DescribeTags 操作已弃用且未维护。要查看与 EFS 资源关联的标签，请使用 ListTagsForResource API 操作。

返回与文件系统关联的标签。未指定在一个 DescribeTags 调用的响应中返回标记的顺序和在多个调用迭代的响应中（当使用分页时）返回标记的顺序。

此操作需要 elasticfilesystem:DescribeTags 操作的权限。

请求语法

```
GET /2015-02-01/tags/FileSystemId?Marker=Marker&MaxItems=MaxItems HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

FileSystemId

要检索其标签集的文件系统 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

是否必需：是

Marker

（可选）从以前的 DescribeTags 操作返回的不透明分页标记（字符串）。如果存在，则它指定从上一调用中断的位置继续列表。

长度限制：最小长度为 1。最大长度为 128。

模式：`.+`

MaxItems

(可选) 在响应中返回的最大文件系统标签数。当前，此数字自动设置为 100，其他值被忽略。如果标签超过 100 个，则响应按每页 100 项进行分页。

有效范围：最小值为 1。

请求正文

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "Marker": "string",
  "NextMarker": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Marker

如果请求包含 Marker，则响应会在此字段中返回该值。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

[NextMarker](#)

如果存在值，则会返回更多标签。在后续请求中，可以在下一请求中提供值 `NextMarker` 作为 `Marker` 参数的值，以检索下一组标签。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：`.+`

[Tags](#)

以 `Tag` 对象数组的形式返回与文件系统关联的标签。

类型：[Tag](#) 对象数组

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定 `FileSystemId` 值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

示例

检索与文件系统关联的标签

以下请求检索与指定文件系统关联的标签（键值对）。

示例请求

```
GET /2015-02-01/tags/fs-01234567/ HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215404Z
Authorization: <...>
```

示例响应

```
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"
    }
  ]
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

ListTagsForResource

列出顶级 EFS 资源的所有标签。必须提供要检索其标签的资源 ID。

此操作需要 `elasticfilesystem:DescribeAccessPoints` 操作的权限。

请求语法

```
GET /2015-02-01/resource-tags/ResourceId?MaxResults=MaxResults&NextToken=NextToken
HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

MaxResults

(可选) 指定要在响应中返回的标签对象的最大数量。默认值是 100。

有效范围：最小值为 1。

NextToken

(可选) 如果响应负载已分页，则可以在后续请求中使用 NextToken 获取下一页接入点。

长度限制：最小长度为 1。最大长度为 128。

模式：.+

ResourceId

指定要检索其标签的 EFS 资源。可以使用此 API 端点检索 EFS 文件系统和接入点的标签。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:(access-point/fsap|file-system/fs)-[0-9a-f]{8,40}|fs(ap)?-[0-9a-f]{8,40})$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[NextToken](#)

如果已对响应负载进行分页，则显示 NextToken。可以在后续请求中使用 NextToken 提取下一页接入点描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：.+

[Tags](#)

指定 EFS 资源的标签数组。

类型：[Tag](#) 对象数组

错误

AccessPointNotFound

如果请求者的指定 AccessPointId 值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

ModifyMountTargetSecurityGroups

修改挂载目标的一组有效安全组。

创建挂载目标时，Amazon EFS 还会创建新网络接口。有关更多信息，请参阅 [CreateMountTarget](#)。此操作将与挂载目标关联的网络接口的有效安全组替换为请求中提供的 SecurityGroups。此操作需要已创建挂载目标的网络接口，并且挂载目标的生命周期状态不是 deleted。

此操作需要以下操作的权限：

- 对挂载目标的文件系统执行 elasticfilesystem:ModifyMountTargetSecurityGroups 操作。
- 在挂载目标的网络接口上执行 ec2:ModifyNetworkInterfaceAttribute 操作。

请求语法

```
PUT /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
Content-type: application/json

{
  "SecurityGroups": [ "string" ]
}
```

URI 请求参数

请求使用以下 URI 参数。

[MountTargetId](#)

要修改其安全组的挂载目标 ID。

长度限制：最小长度为 13。最大长度为 45。

模式：`^fsmt-[0-9a-f]{8,40}$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

SecurityGroups

一组 VPC 安全组 IDs。

类型：字符串数组

数组成员：最多 100 项。

长度限制：最小长度为 11。最大长度为 43。

模式：`^sg-[0-9a-f]{8,40}`

必需：否

响应语法

```
HTTP/1.1 204
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 204 响应。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

IncorrectMountTargetState

如果挂载目标未处于执行操作的正确状态，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

MountTargetNotFound

如果在调用者的 AWS 账户中未找到具有指定 ID 的挂载目标，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

SecurityGroupLimitExceeded

当请求中指定的 SecurityGroups 数量超过基于账户配额设定的限制时，将返回此错误。要么删除一些安全组，要么请求提高账户配额。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [Amazon VPC 配额](#)（参阅安全组表）。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

SecurityGroupNotFound

如果子网的虚拟私有云 (VPC) 中不存在指定的安全组，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

示例

替换挂载目标的安全组

以下示例替换对与挂载目标关联的网络接口有效的安全组。

示例请求

```
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"
  ]
}
```

示例响应

```
HTTP/1.1 204 No Content
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

PutAccountPreferences

使用此操作将当前 AWS 区域 帐户首选项设置为使用长 17 个字符 (63 位) 或短 8 个字符 (32 位) 的资源 IDs 来存储新 EFS 文件系统和挂载目标资源。所有现有资源 IDs 都不会受到您所做的任何更改的影响。在 EFS 过渡到长资源 IDs 时，您可以在选择加入期内设置 ID 首选项。有关更多信息，请参阅[管理 Amazon EFS 资源 IDs](#)。

Note

从 2021 年 10 月开始，如果尝试将帐户首选项设置为使用短 8 个字符格式的资源 ID，则会收到错误。如果您收到错误并且必须使用文件系统和挂载目标资源的缩写 IDs，请联系 AWS 支持人员。

请求语法

```
PUT /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json
```

```
{
  "ResourceIdType": "string"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

[ResourceIdType](#)

指定要为用户设置的 EFS 资源 ID 首选项 AWS 账户 AWS 区域，当前格式为 LONG_ID (17 个字符) 或 SHORT_ID (8 个字符)。

Note

从 2021 年 10 月开始，将帐户首选项设置为 SHORT_ID 时会收到错误。如果您收到错误并且必须使用文件系统和挂载目标资源的缩写 IDs，请联系 AWS 支持人员。

类型：字符串

有效值：LONG_ID | SHORT_ID

是否必需：是

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "ResourceIdPreference": {
    "ResourceIdType": "string",
    "Resources": [ "string" ]
  }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[ResourceIdPreference](#)

描述当前的资源类型及其用户的 AWS 账户 ID 首选项 AWS 区域。

类型：[ResourceIdPreference](#) 对象

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

PutBackupPolicy

更新文件系统的备份策略。可使用此操作启动或停止文件系统的自动备份。

请求语法

```
PUT /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
Content-type: application/json

{
  "BackupPolicy": {
    "Status": "string"
  }
}
```

URI 请求参数

请求使用以下 URI 参数。

FileSystemId

指定要为哪个 EFS 文件系统更新备份策略。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

BackupPolicy

备份策略包含在 PutBackupPolicy 请求中。

类型：[BackupPolicy](#) 对象

是否必需：是

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "BackupPolicy": {
    "Status": "string"
  }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[BackupPolicy](#)

描述文件系统的备份策略，指明是开启还是关闭自动备份。

类型：[BackupPolicy](#) 对象

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifecycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ValidationException

如果请求所在的 AWS Backup 服务不可用 AWS 区域，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

PutFileSystemPolicy

将 Amazon EFS FileSystemPolicy 应用于 Amazon EFS 文件系统。文件系统策略是一种基于 IAM 资源的策略，可以包含多个策略声明。一个文件系统始终只有一个文件系统策略，该策略可以是默认策略，也可以是使用此 API 操作设置或更新的显式策略。EFS 文件系统策略有 2 万个字符的限制。设置显式策略后，它会覆盖默认策略。有关默认文件系统策略的更多信息，请参阅[默认 EFS 文件系统策略](#)。

Note

EFS 文件系统策略有 2 万个字符的限制。

此操作需要 `elasticfilesystem:PutFileSystemPolicy` 操作的权限。

请求语法

```
PUT /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
Content-type: application/json

{
  "BypassPolicyLockoutSafetyCheck": boolean,
  "Policy": "string"
}
```

URI 请求参数

请求使用以下 URI 参数。

[FileSystemId](#)

要为其创建或更新 FileSystemPolicy 的 EFS 文件系统 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

[BypassPolicyLockoutSafetyCheck](#)

(可选) 指定是否绕过 FileSystemPolicy 锁定安全检查的布尔值。锁定安全检查确定请求中的策略是否将锁定或阻止发出请求的 IAM 主体在此文件系统上发出未来的 PutFileSystemPolicy 请求。仅当您打算阻止发出请求的 IAM 主体在此文件系统上发出后续 PutFileSystemPolicy 请求时才将 BypassPolicyLockoutSafetyCheck 设置为 True。默认值为 False。

类型：布尔值

必需：否

[Policy](#)

正在创建的 FileSystemPolicy。接受 JSON 格式的策略定义。EFS 文件系统策略有 2 万个字符的限制。要详细了解构成文件系统策略的元素，请参阅 [Amazon EFS 中基于资源的策略](#)。

类型：字符串

长度限制：最小长度为 1。最大长度为 20000。

模式：`[\s\S]+`

必需：是

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "FileSystemId": "string",
  "Policy": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

FileSystemId

指定 FileSystemPolicy 适用的 EFS 文件系统。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Policy

EFS 文件系统的 JSON 格式的 FileSystemPolicy。

类型：字符串

长度限制：最小长度为 1。最大长度为 20000。

模式：`[\s\S]+`

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定 FileSystemId 值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifecycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

InvalidPolicyException

如果 `FileSystemPolicy` 格式错误或包含错误（例如参数值无效或缺少必填参数），则返回。如果出现策略锁定安全检查错误，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述（英语）。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

示例

创建 EFS `FileSystemPolicy`

以下请求创建一个 `FileSystemPolicy`，允许所有 AWS 委托人以读写权限挂载指定的 EFS 文件系统。

示例请求

```
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": ["*"]
      },
    }
  ]
}
```

示例响应

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

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

PutLifecycleConfiguration

使用此操作可以管理文件系统的存储。LifecycleConfiguration 由一个或多个定义以下内容的 LifecyclePolicy 对象组成：

- **TransitionToIA** – 何时将文件系统上的文件从主存储（标准存储类）移到不频繁访问 (IA) 存储。
- **TransitionToArchive** – 何时将文件系统上的文件从其当前存储类别（IA 或标准存储）移到存档存储。

在转换为 IA 存储之前，文件系统无法转换为归档存储。因此，TransitionToArchive 要么不能设置，要么必须晚于 TransitionTo IA。

Note

归档存储类仅适用于使用弹性吞吐量模式和通用性能模式的文件系统。

- **TransitionToPrimaryStorageClass** – 在 IA 或归档存储中访问文件后，是否将文件系统上的文件移回主存储（标准存储类）。

有关更多信息，请参阅[管理文件系统存储](#)。

每个 Amazon EFS 文件系统都支持一种生命周期配置，该配置适用于该文件系统中的所有文件。如果指定文件系统的 LifecycleConfiguration 对象已存在，PutLifecycleConfiguration 调用会修改现有配置。请求正文中包含空 LifecyclePolicies 数组的 PutLifecycleConfiguration 调用会删除任何现有 LifecycleConfiguration。在该请求中，指定以下项：

- 您要启用、禁用或修改其生命周期管理的文件系统的 ID。
- LifecyclePolicy 对象的 LifecyclePolicies 数组，这些对象定义何时将文件移至 IA 存储、存档存储以及何时移回主存储。

Note

Amazon EFS 要求每个 LifecyclePolicy 对象只有一次转换，因此需要使用单独的 LifecyclePolicy 对象来构造 LifecyclePolicies 数组。有关更多信息，请参阅以下部分中的示例请求。

此操作需要 `elasticfilesystem:PutLifecycleConfiguration` 操作的权限。

要将 LifecycleConfiguration 对象应用于加密文件系统，您需要与创建加密文件系统时相同的 AWS Key Management Service 权限。

请求语法

```
PUT /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
Content-type: application/json

{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "string",
      "TransitionToIA": "string",
      "TransitionToPrimaryStorageClass": "string"
    }
  ]
}
```

URI 请求参数

请求使用以下 URI 参数。

FileSystemId

要为其创建 LifecycleConfiguration 对象的文件系统 ID (字符串)。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

LifecyclePolicies

定义文件系统 LifecycleConfiguration 对象的 LifecyclePolicy 对象数组。LifecycleConfiguration 对象向生命周期管理通知以下内容：

- **TransitionToIA** – 何时将文件系统中的文件从主存储（标准存储类）移到不频繁访问 (IA) 存储。
- **TransitionToArchive** - 何时将文件系统中的文件从其当前存储类别（IA 或标准存储）移到存档存储。

在转换为 IA 存储之前，文件系统无法转换为归档存储。因此，TransitionToArchive 要么不能设置，要么必须晚于 TransitionTo IA。

 Note

归档存储类仅适用于使用弹性吞吐量模式和通用性能模式的文件系统。

- **TransitionToPrimaryStorageClass** – 在 IA 或归档存储中访问文件后，是否将文件系统中的文件移回主存储（标准存储类）。

 Note

使用 put-lifecycle-configuration CLI 命令或 PutLifecycleConfiguration API 操作时，Amazon EFS 要求每个 LifecyclePolicy 对象只有一次转换。这意味着在请求正文中，LifecyclePolicies 必须构造为 LifecyclePolicy 对象的数组，每个存储转换对应于一个对象。有关更多信息，请参阅以下部分中的示例请求。

类型：[LifecyclePolicy](#) 对象数组

数组成员：最多 3 项。

是否必需：是

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "string",
      "TransitionToIA": "string",
      "TransitionToPrimaryStorageClass": "string"
    }
  ]
}
```

```
    }  
  ]  
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[LifecyclePolicies](#)

一系列生命周期管理策略。EFS 支持每个文件系统最多一个策略。

类型：[LifecyclePolicy](#) 对象数组

数组成员：最多 3 项。

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifecycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

示例

创建生命周期配置

以下示例使用 `PutLifecycleConfiguration` 操作创建 `LifecyclePolicy` 对象。此示例将创建一个生命周期策略，指示 EFS 执行以下操作：

- 将文件系统中过去 30 天内未在标准存储中访问过的所有文件移到 IA 存储。
- 将文件系统中过去 90 天内未在标准存储中访问过的所有文件移到存档存储。
- 在 IA 或存档存储中访问文件后，将文件移回标准存储。归档存储类仅适用于使用弹性吞吐量模式和通用性能模式的文件系统。

有关更多信息，请参阅 [EFS 存储类](#) 和 [管理文件系统存储](#)。

示例请求

```
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"
    }
  ]
}
```

示例响应

```
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-configurationCLI 请求示例

此示例说明了的一种用法 PutLifecycleConfiguration。

示例请求

```
aws efs put-lifecycle-configuration \
  --file-system-id fs-0123456789abcdfb \
  --lifecycle-policies "[{"TransitionToArchive":"AFTER_90_DAYS"},
  {"TransitionToIA":"AFTER_30_DAYS"},
  {"TransitionToPrimaryStorageClass":"AFTER_1_ACCESS"}]"
  --region us-west-2 \
  --profile adminuser
```

示例响应

```
{
  "LifecyclePolicies": [
    {
      "TransitionToArchive": "AFTER_90_DAYS"
    },
    {
      "TransitionToIA": "AFTER_30_DAYS"
    },
  ],
}
```

```
{
  "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
}
]
```

禁用生命周期管理

以下示例禁用指定文件系统的生命周期管理。

示例请求

```
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": [ ]
}
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-type: application/json
Content-Length: 86

{
  "LifecyclePolicies": [ ]
}
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

TagResource

为 EFS 资源创建标签。可以使用此 API 操作创建 EFS 文件系统和接入点的标签。

此操作需要 `elasticfilesystem:TagResource` 操作的权限。

请求语法

```
POST /2015-02-01/resource-tags/ResourceId HTTP/1.1
Content-type: application/json
```

```
{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

URI 请求参数

请求使用以下 URI 参数。

ResourceId

指定要为其创建标签的 EFS 资源的 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:(access-point/fsap|file-system/fs)-[0-9a-f]{8,40}|fs(ap)?-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

Tags

要添加的 Tag 对象的数组。每个 Tag 对象都是一个键值对。

类型：[Tag](#) 对象数组

是否必需：是

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

AccessPointNotFound

如果请求者的指定AccessPointId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

示例

在文件系统上创建标签

以下请求在指定的文件系统上创建三个标签 ("key1"、"key2" 和 "key3")。

示例请求

```
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"
    }
  ]
}
```

示例响应

```
HTTP/1.1 204 no content
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)

- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

UntagResource

从 EFS 资源中移除标签。可以使用此 API 操作从 EFS 文件系统和接入点移除标签。

此操作需要 `elasticfilesystem:UntagResource` 操作的权限。

请求语法

```
DELETE /2015-02-01/resource-tags/ResourceId?tagKeys=TagKeys HTTP/1.1
```

URI 请求参数

请求使用以下 URI 参数。

ResourceId

指定要从中移除标签的 EFS 资源。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:(access-point/fsap|file-system/fs)-[0-9a-f]{8,40}|fs(ap)?-[0-9a-f]{8,40})$`

是否必需：是

TagKeys

要从指定的 EFS 资源中移除的键值标签对的键。

数组成员：最少 1 个物品。最多 50 项。

长度限制：最小长度为 1。最大长度为 128。

模式：`^(?!([aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_.:/=+\-@]+))$`

必需：是

请求体

该请求没有请求正文。

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

AccessPointNotFound

如果请求者的指定AccessPointId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)

- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

UpdateFileSystem

更新现有文件系统的吞吐量模式或预置吞吐量。

请求语法

```
PUT /2015-02-01/file-systems/FileSystemId HTTP/1.1
Content-type: application/json

{
  "ProvisionedThroughputInMibps": number,
  "ThroughputMode": "string"
}
```

URI 请求参数

请求使用以下 URI 参数。

FileSystemId

要更新的文件系统 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

ProvisionedThroughputInMibps

(可选) 您要为正在创建的文件系统预配置的吞吐量，以兆字节每秒 (MiBps) 为单位。如果将 `ThroughputMode` 设置为 `provisioned`，则是必需的。有效值为 1-3414 MiBps，上限视区域而定。要提高此限制，请联系 支持。有关更多信息，请参阅《Amazon EFS 用户指南》中的[您可以提高的 Amazon EFS 配额](#)。

类型：双精度

有效范围：最小值为 1.0。

必需：否

ThroughputMode

(可选) 更新文件系统的吞吐量模式。如果不更新吞吐量模式，则无需在请求中提供此值。如果将 ThroughputMode 更改为 provisioned，则还必须设置 ProvisionedThroughputInMibps 的值。

类型：字符串

有效值：bursting | provisioned | elastic

必需：否

响应语法

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"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 202 响应。

服务以 JSON 格式返回以下数据。

[AvailabilityZoneId](#)

文件系统所在可用区的唯一且一致的标识符，仅对单区域文件系统有效。例如，use1-az1是 us-east AWS 区域-1 的可用区 ID，它在每个可用区中的位置都相同。AWS 账户

类型：字符串

[AvailabilityZoneName](#)

描述文件系统所在的 AWS 可用区，并且仅对单区域文件系统有效。有关更多信息，请参阅《Amazon EFS 用户指南》中的[使用 EFS 存储类](#)。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：.+

[CreationTime](#)

文件系统的创建时间，以秒为单位（自 1970-01-01T00:00:00Z 起）。

类型：时间戳

[CreationToken](#)

请求中指定的不透明字符串。

类型：字符串

长度限制：最小长度为 1。长度上限为 64。

模式：.+

Encrypted

一个布尔值，如果设为 true，则指示文件系统已加密。

类型：布尔值

FileSystemArn

EFS 文件系统的 Amazon 资源名称 (ARN)，采用 `arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id` 格式。使用示例数据的示例：`arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567`

类型：字符串

FileSystemId

文件系统 ID，由 Amazon EFS 分配。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

FileSystemProtection

描述文件系统上的保护功能。

类型：[FileSystemProtectionDescription](#) 对象

KmsKeyId

AWS KMS key 用于保护加密文件系统的 ID。

类型：字符串

长度约束：最大长度为 2048。

模式：`^([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

文件系统的生命周期阶段。

类型：字符串

有效值：`creating | available | updating | deleting | deleted | error`

Name

可以向文件系统添加标签，包括 Name 标签。有关更多信息，请参阅 [CreateFileSystem](#)。如果文件系统有 Name 标签，Amazon EFS 会返回此字段中的值。

类型：字符串

长度约束：最大长度为 256。

模式：`^([\p{L}\p{Z}\p{N}_.:\/=+\-@]*)$`

NumberOfMountTargets

文件系统当前的挂载目标数。有关更多信息，请参阅 [CreateMountTarget](#)。

类型：整数

有效范围：最小值为 0。

OwnerId

AWS 账户 创建文件系统的。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

PerformanceMode

文件系统的性能模式。

类型：字符串

有效值：generalPurpose | maxIO

[ProvisionedThroughputInMibps](#)

文件系统的预配置吞吐量（以 MiBps 衡量单位）。对使用将 ThroughputMode 设置为 provisioned 的文件系统有效。

类型：双精度

有效范围：最小值为 1.0。

[SizeInBytes](#)

文件系统中存储的数据的最新已知计量大小（以字节为单位）（在其 Value 字段中），以及确定该大小的时间（在其 Timestamp 字段中）。Timestamp 值是自 1970-01-01T00:00:00Z 以来的整数秒数。SizeInBytes 值并不代表文件系统快照的大小，但是当没有写入文件系统时，该值最终会保持一致。也就是说，只有在超过几个小时的时间内未修改文件系统时，SizeInBytes 才表示实际大小。否则，该值不是文件系统在任何时间点的确切大小。

类型：[FileSystemSize](#) 对象

[Tags](#)

与文件系统关联的标签，以 Tag 对象数组形式呈现。

类型：[Tag](#) 对象数组

[ThroughputMode](#)

显示文件系统的吞吐量模式。有关更多信息，请参阅《Amazon EFS 用户指南》中的[吞吐量模式](#)。

类型：字符串

有效值：bursting | provisioned | elastic

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifeCycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InsufficientThroughputCapacity

如果没有足够的容量来预置额外的吞吐量，则返回此内容。尝试在预配置吞吐量模式下创建文件系统，尝试增加现有文件系统的预配置吞吐量，或尝试将现有文件系统从突增吞吐量模式更改为预配置吞吐量模式时，可能会返回此值。请稍后重试。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：503

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ThroughputLimitExceeded

如果由于已达到 1024 的吞吐量限制而无法更改吞吐量模式或预配置吞吐量 MiB/s，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

TooManyRequests

如果在更改吞吐量模式或降低预配置吞吐量值之前没有等待至少 24 小时，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：429

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版软件开发工具包](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

UpdateFileSystemProtection

更新文件系统的保护。

此操作需要 `elasticfilesystem:UpdateFileSystemProtection` 操作的权限。

请求语法

```
PUT /2015-02-01/file-systems/FileSystemId/protection HTTP/1.1
Content-type: application/json

{
  "ReplicationOverwriteProtection": "string"
}
```

URI 请求参数

请求使用以下 URI 参数。

[FileSystemId](#)

要更新的文件系统的 ID。

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

请求体

请求接受采用 JSON 格式的以下数据。

[ReplicationOverwriteProtection](#)

文件系统的复制覆盖保护的状态。

- **ENABLED** – 此文件系统不能用作复制配置中的目标文件系统。文件系统是可写的。默认情况下，复制覆盖保护功能处于 **ENABLED** 状态。
- **DISABLED** – 此文件系统可以用作复制配置中的目标文件系统。文件系统是只读的，只能通过 EFS 复制进行修改。

- REPLICATING – 此文件系统正用作复制配置中的目标文件系统。文件系统是只读的，仅通过 EFS 复制进行修改。

如果删除复制配置，则文件系统的复制覆盖保护功能将重新启用，文件系统将变为可写状态。

类型：字符串

有效值：ENABLED | DISABLED | REPLICATING

必需：否

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
  "ReplicationOverwriteProtection": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

[ReplicationOverwriteProtection](#)

文件系统的复制覆盖保护的状态。

- ENABLED – 此文件系统不能用作复制配置中的目标文件系统。文件系统是可写的。默认情况下，复制覆盖保护功能处于 ENABLED 状态。
- DISABLED – 此文件系统可以用作复制配置中的目标文件系统。文件系统是只读的，只能通过 EFS 复制进行修改。
- REPLICATING – 此文件系统正用作复制配置中的目标文件系统。文件系统是只读的，仅通过 EFS 复制进行修改。

如果删除复制配置，则文件系统的复制覆盖保护功能将重新启用，文件系统将变为可写状态。

类型：字符串

有效值：ENABLED | DISABLED | REPLICATING

错误

BadRequest

如果请求格式错误或包含错误（例如参数值无效或缺少必填参数），则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

FileSystemNotFound

如果请求者的指定FileSystemId值不存在，则返回。AWS 账户

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：404

IncorrectFileSystemLifecycleState

如果文件系统的生命周期状态不是“可用”，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

InsufficientThroughputCapacity

如果没有足够的容量来预置额外的吞吐量，则返回此内容。尝试在预配置吞吐量模式下创建文件系统，尝试增加现有文件系统的预配置吞吐量，或尝试将现有文件系统从突增吞吐量模式更改为预配置吞吐量模式时，可能会返回此值。请稍后重试。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：503

InternalServerError

如果服务器端发生错误，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：500

ReplicationAlreadyExists

如果文件系统已包含在复制配置中，则返回此内容。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：409

ThroughputLimitExceeded

如果由于已达到 1024 的吞吐量限制而无法更改吞吐量模式或预配置吞吐量 MiB/s，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：400

TooManyRequests

如果在更改吞吐量模式或降低预配置吞吐量值之前没有等待至少 24 小时，则返回。

ErrorCode

错误代码是用于唯一标识错误条件的字符串。这意味着按类型检测和处理错误的程序将读取和理解错误代码。

Message

错误消息包含错误条件的常规描述 (英语)。它主要面向用户受众。如果用户遇到不知如何处理或不愿处理的错误条件，简单的程序将直接向最终用户显示消息。支持更详尽的错误处理和适当国际化的复杂程序更容易忽略错误消息。

HTTP 状态代码：429

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWS SDKs，请参阅以下内容：

- [AWS 命令行界面 V2](#)
- [AWS 适用于 .NET 的 SDK V4](#)
- [AWS 适用于 C++ 的 SDK](#)
- [AWS 适用于 Go v2 的 SDK](#)
- [AWS 适用于 Java 的 SDK V2](#)
- [AWS JavaScript V3 版 SDK](#)
- [AWS 适用于 Kotlin 的 SDK](#)
- [AWS 适用于 PHP 的 SDK V3](#)
- [AWS Python 软件开发工具包](#)
- [AWS 适用于 Ruby V3 的 SDK](#)

数据类型

支持以下数据类型：

- [AccessPointDescription](#)
- [BackupPolicy](#)
- [CreationInfo](#)
- [Destination](#)
- [DestinationToCreate](#)
- [FileSystemDescription](#)
- [FileSystemProtectionDescription](#)
- [FileSystemSize](#)
- [LifecyclePolicy](#)
- [MountTargetDescription](#)
- [PosixUser](#)
- [ReplicationConfigurationDescription](#)
- [ResourceIdPreference](#)

- [RootDirectory](#)
- [Tag](#)

AccessPointDescription

提供对 EFS 文件系统接入点的描述。

目录

AccessPointArn

与接入点关联的唯一 Amazon 资源名称 (ARN) 。

类型：字符串

长度限制：最大长度为 128。

模式：`^arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}$`

必需：否

AccessPointId

接入点 ID，由 Amazon EFS 分配。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

必需：否

ClientToken

请求中指定的不透明字符串，以确保幂等创建。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：`.+`

必需：否

FileSystemId

访问点应用到的 EFS 文件系统的 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：否

LifeCycleState

标识接入点的生命周期阶段。

类型：字符串

有效值：`creating | available | updating | deleting | deleted | error`

必需：否

Name

接入点的名称。这是 Name 标签的值。

类型：字符串

必需：否

OwnerId

标识拥有接入点资源的 AWS 账户。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

必需：否

PosixUser

访问点上的完整 POSIX 身份，包括用户 ID、组 ID 和辅助组 ID，由使用访问点的 NFS 客户端用于所有文件操作。

类型：[PosixUser](#) 对象

必需：否

RootDirectory

EFS 文件系统上的目录，接入点将其作为根目录，向使用接入点的 NFS 客户端公开。

类型：[RootDirectory](#) 对象

必需：否

Tags

与接入点关联的标签，以标签对象数组的形式呈现。

类型：[Tag](#) 对象数组

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

BackupPolicy

用于创建每日自动备份的文件系统的备份策略。如果状态值为 `ENABLED`，则表示文件系统正在自动备份。有关更多信息，请参阅[自动备份](#)。

目录

Status

描述文件系统备份策略的状态。

- **ENABLED** – EFS 正在自动备份文件系统。
- **ENABLING** – EFS 正在开启文件系统自动备份。
- **DISABLED** – 已关闭文件系统自动备份。
- **DISABLING** – EFS 正在关闭文件系统自动备份。

类型：字符串

有效值：ENABLED | ENABLING | DISABLED | DISABLING

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

CreationInfo

如果指定的 `RootDirectory > Path` 不存在则必需。指定 POSIX ID 和权限以应用到访问点的 `RootDirectory > Path`。如果访问点根目录不存在，则当客户端连接到访问点时，EFS 使用这些设置创建根目录。指定 `CreationInfo` 时，您必须包含所有属性的值。

只有在您提供 `CreationInfo: OwnUid`、`OwnGID` 以及相关目录权限的情况下，Amazon EFS 才会创建根目录。如果您不提供此信息，Amazon EFS 不会创建根目录。如果根目录不存在，则使用接入点进行挂载的尝试将失败。

Important

如果您不提供 `CreationInfo` 并且指定的 `RootDirectory` 不存在，则使用访问点挂载文件系统的尝试将失败。

目录

OwnerGid

指定要应用到 `RootDirectory` 的 POSIX 组 ID。接受介于 0 到 2^{32} 之间的值 (4294967295)。

类型：长整型

有效范围：最小值为 0。最大值为 4294967295。

必需：是

OwnerUid

指定要应用到 `RootDirectory` 的 POSIX 用户 ID。接受介于 0 到 2^{32} 之间的值 (4294967295)。

类型：长整型

有效范围：最小值为 0。最大值为 4294967295。

必需：是

Permissions

指定要应用到 `RootDirectory` 的 POSIX 权限，格式为表示文件的模式位的八进制数。

类型：字符串

长度约束：最小长度为 3。最大长度为 4。

模式：`^[0-7]{3,4}$`

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

Destination

描述复制配置中的目标文件系统。

目录

FileSystemId

Amazon EFS 文件系统 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

Region

目标文件系统所在的 AWS 区域。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：`^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

必需：是

Status

描述复制配置的状态。有关复制状态的更多信息，请参阅《Amazon EFS 用户指南》中的[查看复制详细信息](#)。

类型：字符串

有效值：ENABLED | ENABLING | DELETING | ERROR | PAUSED | PAUSING

必需：是

LastReplicatedTimestamp

在目标文件系统上成功完成最近一次同步的时间。在此时间之前对源文件系统上的数据所做的任何更改都已成功复制到目标文件系统。在此时间之后发生的任何更改都可能无法完全复制。

类型：时间戳

必需：否

OwnerId

目标文件系统所在的 AWS 账户的 ID。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\d{12})|(\d{4}-\d{4}-\d{4})$`

必需：否

RoleArn

源账户中支持 Amazon EFS 代表其执行复制的 IAM 角色的 Amazon 资源名称 (ARN)。对于同账户复制，此为可选参数；但对于跨账户复制，此为必需参数。

类型：字符串

长度约束：最大长度为 2048。

模式：`arn:(aws[a-zA-Z-]*)?:iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

必需：否

StatusMessage

该消息提供有关复制目标配置的 PAUSED 或 ERROR 状态的详细信息。有关复制状态消息的更多信息，请参阅《Amazon EFS 用户指南》中的[查看复制详细信息](#)。

类型：字符串

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

DestinationToCreate

描述复制配置的新或现有的目标文件系统。

- 如果要复制到新的文件系统，请不要为目标文件系统指定文件系统 ID。Amazon EFS 将创建新的空文件系统。对于单区存储，指定要在其中创建文件系统的可用区。要使用默认 KMS 密钥以外的 AWS Key Management Service 密钥，请指定该密钥。有关更多信息，请参阅《Amazon EFS 用户指南》中的[配置到新 Amazon EFS 文件系统的复制](#)。

Note

创建文件系统后，即无法更改 KMS 密钥或性能模式。

- 如果要复制到与源文件系统同一个账户的现有文件系统，则需要提供要复制到的文件系统的 ID 或 Amazon 资源名称 (ARN)。必须禁用文件系统的复制覆盖保护功能。有关更多信息，请参阅《Amazon EFS 用户指南》中的[复制到现有文件系统](#)。
- 如果要将文件系统复制到与源文件系统不同账户的文件系统（跨账户复制），则需要提供该文件系统的 ARN 和支持 Amazon EFS 对目标账户执行复制的 IAM 角色。必须禁用文件系统的复制覆盖保护功能。有关更多信息，请参阅《Amazon EFS 用户指南》中的[跨 AWS 账户复制](#)。

目录

AvailabilityZoneName

要创建使用单区存储的文件系统，请指定要在其中创建目标文件系统的可用区的名称。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：.+

必需：否

FileSystemId

要用于目标的文件系统的 ID 或 ARN。对于跨账户复制，这必须是 ARN。必须禁用文件系统的复制覆盖复制功能。如果未指定 ID 或 ARN，则会创建一个新的文件系统。

Note

当您最初为现有文件系统配置复制时，Amazon EFS 会在目标文件系统中写入数据或删除现有数据，以与源文件系统中的数据相匹配。如果您不想更改目标文件系统中的数据，则应改为复制到新的文件系统。有关更多信息，请参阅 <https://docs.aws.amazon.com/efs/latest/ug/create-replication.html>。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：否

KmsKeyId

指定要用于加密目标文件系统的 AWS Key Management Service (AWS KMS) 密钥。如果您不指定 KMS 密钥，Amazon EFS 将使用 Amazon EFS 的默认 KMS 密钥 `/aws/elasticfilesystem`。此 ID 可以是下列格式之一：

- 键 ID - 键的唯一标识符，例如 `1234abcd-12ab-34cd-56ef-1234567890ab`。
- ARN - 键的 ARN，例如 `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`。
- 键别名 - 之前为键创建的显示名称，例如 `alias/projectKey1`。
- 键别名 ARN - 键别名的 ARN，例如 `arn:aws:kms:us-west-2:444455556666:alias/projectKey1`。

类型：字符串

长度约束：最大长度为 2048。

模式：`^([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/_-]+))))$`

必需：否

Region

要创建使用区域存储的文件系统，请指定要在其中创建目标文件系统的 AWS 区域。必须为拥有源文件系统的 AWS 账户启用区域。有关更多信息，请参阅《AWS 一般参考指南》中的[管理 AWS 区域](#)。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：`^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

必需：否

RoleArn

源账户中支持 Amazon EFS 代表其执行复制的 IAM 角色的 Amazon 资源名称 (ARN)。对于同账户复制，此为可选参数；但对于跨账户复制，此为必需参数。

类型：字符串

长度约束：最大长度为 2048。

模式：`arn:(aws[a-zA-Z-]*)?:iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FileSystemDescription

操作系统的描述。

目录

CreationTime

文件系统的创建时间，以秒为单位（自 1970-01-01T00:00:00Z 起）。

类型：时间戳

必需：是

CreationToken

请求中指定的不透明字符串。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：.+

必需：是

FileSystemId

文件系统 ID，由 Amazon EFS 分配。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

LifeCycleState

文件系统的生命周期阶段。

类型：字符串

有效值：creating | available | updating | deleting | deleted | error

必需：是

NumberOfMountTargets

文件系统当前的挂载目标数。有关更多信息，请参阅 [CreateMountTarget](#)。

类型：整数

有效范围：最小值为 0。

必需：是

OwnerId

创建文件系统的 AWS 账户。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\d{12})|(\d{4}-\d{4}-\d{4})$`

必需：是

PerformanceMode

文件系统的性能模式。

类型：字符串

有效值：generalPurpose | maxIO

必需：是

SizeInBytes

文件系统中存储的数据的最新已知计量大小（以字节为单位）（在其 Value 字段中），以及确定该大小的时间（在其 Timestamp 字段中）。Timestamp 值是自 1970-01-01T00:00:00Z 以来的整数秒数。SizeInBytes 值并不代表文件系统一致快照的大小，但是当没有写入文件系统时，该值最终会保持一致。也就是说，只有在超过几个小时的时间内未修改文件系统时，SizeInBytes 才表示实际大小。否则，该值不是文件系统在任何时间点的确切大小。

类型：[FileSystemSize](#) 对象

必需：是

Tags

与文件系统关联的标签，以 Tag 对象数组形式呈现。

类型：[Tag](#) 对象数组

必需：是

AvailabilityZoneId

文件系统所在可用区的唯一且一致的标识符，仅对单区域文件系统有效。例如，use1-az1 是 us-east-1 AWS 区域的可用区 ID，它在每个 AWS 账户中的位置均相同。

类型：字符串

必需：否

AvailabilityZoneName

描述文件系统所在的 AWS 可用区，仅对单区文件系统有效。有关更多信息，请参阅《Amazon EFS 用户指南》中的[使用 EFS 存储类](#)。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：.+

必需：否

Encrypted

一个布尔值，如果设为 true，则指示文件系统已加密。

类型：布尔值

必需：否

FileSystemArn

EFS 文件系统的 Amazon 资源名称 (ARN)，采用 `arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id` 格式。使用示例数据的示例：`arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567`

类型：字符串

必需：否

FileSystemProtection

描述文件系统上的保护功能。

类型：[FileSystemProtectionDescription](#) 对象

必需：否

KmsKeyId

用于保护加密文件系统的 AWS KMS key 的 ID。

类型：字符串

长度约束：最大长度为 2048。

模式：`^([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/_-]+))))$`

必需：否

Name

可以向文件系统添加标签，包括 Name 标签。有关更多信息，请参阅 [CreateFileSystem](#)。如果文件系统有 Name 标签，Amazon EFS 会返回此字段中的值。

类型：字符串

长度约束：最大长度为 256。

模式：`^([\p{L}\p{Z}\p{N}_.: /+=\ -@]*)$`

必需：否

ProvisionedThroughputInMibps

为文件系统预置的吞吐量（以 MiBps 为单位）。对使用将 `ThroughputMode` 设置为 `provisioned` 的文件系统有效。

类型：双精度

有效范围：最小值为 1.0。

必需：否

ThroughputMode

显示文件系统的吞吐量模式。有关更多信息，请参阅《Amazon EFS 用户指南》中的[吞吐量模式](#)。

类型：字符串

有效值：bursting | provisioned | elastic

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

FileSystemProtectionDescription

描述文件系统上的保护功能。

目录

ReplicationOverwriteProtection

文件系统的复制覆盖保护的状态。

- **ENABLED** – 此文件系统不能用作复制配置中的目标文件系统。文件系统是可写的。默认情况下，复制覆盖保护功能处于 **ENABLED** 状态。
- **DISABLED** – 此文件系统可以用作复制配置中的目标文件系统。文件系统是只读的，只能通过 EFS 复制进行修改。
- **REPLICATING** – 此文件系统正用作复制配置中的目标文件系统。文件系统是只读的，仅通过 EFS 复制进行修改。

如果删除复制配置，则文件系统的复制覆盖保护功能将重新启用，文件系统将变为可写状态。

类型：字符串

有效值：ENABLED | DISABLED | REPLICATING

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FileSystemSize

文件系统中存储的数据的最新已知计量大小（以字节为单位）（在其 Value 字段中），以及确定该大小的时间（在其 Timestamp 字段中）。该值并不代表文件系统一致快照的大小，但是当没有写入文件系统时，该值最终会保持一致。也就是说，只有在超过几个小时的时间内未修改文件系统时，此值才表示实际大小。否则，此值不一定是文件系统在任何时间点的确切大小。

目录

Value

存储在文件系统中的数据的最新已知计量大小（以字节为单位）。

类型：长整型

有效范围：最小值为 0。

必需：是

Timestamp

确定 Value 字段中返回的数据大小的时间。此值是自 1970-01-01T00:00:00Z 以来的整数秒数。

类型：时间戳

必需：否

ValueInArchive

存储在归档存储类中的数据的最新已知计量大小（以字节为单位）。

类型：长整型

有效范围：最小值为 0。

必需：否

ValueInIA

存储在不频繁访问存储类中的数据的最新已知计量大小（以字节为单位）。

类型：长整型

有效范围：最小值为 0。

必需：否

ValueInStandard

存储在标准存储类中的数据的最新已知计量大小（以字节为单位）。

类型：长整型

有效范围：最小值为 0。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

LifecyclePolicy

描述生命周期管理使用的策略，该策略指定何时将文件转入和转出存储类。有关更多信息，请参阅[管理文件系统存储](#)。

Note

使用 `put-lifecycle-configuration` CLI 命令或 `PutLifecycleConfiguration` API 操作时，Amazon EFS 要求每个 `LifecyclePolicy` 对象只有一次转换。这意味着在请求正文中，`LifecyclePolicies` 必须构造为 `LifecyclePolicy` 对象的数组，每个转换对应于一个对象。有关更多信息，请参阅 [PutLifecycleConfiguration](#) 中的请求示例。

目录

TransitionToArchive

在主存储（标准存储类）中上次访问文件后的天数，在该天数之后将文件移至归档存储。诸如列出目录内容等元数据操作不算作文件访问事件。

类型：字符串

有效值：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

必需：否

TransitionToIA

在主存储（标准存储类）中上次访问文件后的天数，在该天数之后将文件移至不频繁访问 (IA) 存储。诸如列出目录内容等元数据操作不算作文件访问事件。

类型：字符串

有效值：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

必需：否

TransitionToPrimaryStorageClass

在 IA 或归档存储中访问文件后，是否将文件移回主（标准）存储。诸如列出目录内容等元数据操作不算作文件访问事件。

类型：字符串

有效值：AFTER_1_ACCESS

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

MountTargetDescription

提供对挂载目标的描述。

目录

FileSystemId

挂载目标要用于的文件系统 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

LifeCycleState

挂载目标的生命周期状态。

类型：字符串

有效值：`creating | available | updating | deleting | deleted | error`

必需：是

MountTargetId

系统分配的挂载目标 ID。

类型：字符串

长度限制：最小长度为 13。最大长度为 45。

模式：`^fsmt-[0-9a-f]{8,40}$`

必需：是

SubnetId

挂载目标子网的 ID。

类型：字符串

长度限制：最小长度为 15。最大长度为 47。

模式：`^subnet-[0-9a-f]{8,40}$`

必需：是

AvailabilityZoneId

挂载目标所在可用区的唯一且一致的标识符。例如，`use1-az1` 是 `us-east-1` 区域的 AZ ID，它在每个 AWS 账户中的位置均相同。

类型：字符串

必需：否

AvailabilityZoneName

挂载目标所在的可用区名称。可用区独立映射到每个 AWS 账户名称。例如，您 AWS 账户的可用区 `us-east-1a` 可能与另一 AWS 账户的 `us-east-1a` 不在同一位置。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：`.+`

必需：否

IpAddress

挂载目标的 IPv4 地址。

类型：字符串

长度限制：最小长度为 7。最大长度为 15。

模式：`^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

必需：否

Ipv6Address

挂载目标的 IPv6 地址。

类型：字符串

长度约束：最小长度为 3。长度上限为 39。

必需：否

NetworkInterfaceId

Amazon EFS 在创建挂载目标时创建的网络接口 ID。

类型：字符串

必需：否

OwnerId

拥有资源的 AWS 账户 ID。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\d{12})|(\d{4}-\d{4}-\d{4})$`

必需：否

VpcId

在其中配置挂载目标的虚拟私有云 (VPC) ID。

类型：字符串

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PosixUser

访问点上的完整 POSIX 身份，包括用户 ID、组 ID 和任何辅助组 ID，由使用访问点的 NFS 客户端用于执行所有文件系统操作。

目录

Gid

使用此访问点的所有文件系统操作所用的 POSIX 组 ID。

类型：长整型

有效范围：最小值为 0。最大值为 4294967295。

必需：是

Uid

使用此访问点的所有文件系统操作所用的 POSIX 用户 ID。

类型：长整型

有效范围：最小值为 0。最大值为 4294967295。

必需：是

SecondaryGids

使用此访问点的所有文件系统操作所用的辅助 POSIX 组 ID。

类型：长数组

数组成员：最少 0 个物品。最多 16 项。

有效范围：最小值为 0。最大值为 4294967295。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)

- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

ReplicationConfigurationDescription

描述特定文件系统的复制配置。

目录

CreationTime

描述复制配置的创建时间。

类型：时间戳

必需：是

Destinations

目标对象的数组。仅支持一个目标对象。

类型：[Destination](#) 对象数组

必需：是

OriginalSourceFileSystemArn

复制配置中原始源 EFS 文件系统的 Amazon 资源名称 (ARN)。

类型：字符串

必需：是

SourceFileSystemArn

复制配置中当前源文件系统的 Amazon 资源名称 (ARN)。

类型：字符串

必需：是

SourceFileSystemId

正在复制的源 Amazon EFS 文件系统 ID。

类型：字符串

长度限制：最大长度为 128。

模式：`^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

必需：是

SourceFileSystemRegion

源 EFS 文件系统所在的 AWS 区域。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：`^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

必需：是

SourceFileSystemOwnerId

源文件系统所在的 AWS 账户的 ID。

类型：字符串

长度限制：最大长度为 14。

模式：`^(\\d{12})|(\\d{4}-\\d{4}-\\d{4})$`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ResourceIdPreference

描述当前 AWS 区域中用户的 AWS 账户的资源类型及其 ID 首选项。

目录

ResourceIdType

标识 EFS 资源 ID 首选项，可以是 LONG_ID (17 个字符) 或 SHORT_ID (8 个字符) 。

类型：字符串

有效值：LONG_ID | SHORT_ID

必需：否

Resources

标识 ID 首选项设置 FILE_SYSTEM 和 MOUNT_TARGET 适用的 Amazon EFS 资源。

类型：字符串数组

有效值：FILE_SYSTEM | MOUNT_TARGET

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

RootDirectory

指定 Amazon EFS 文件系统上访问点提供了访问权限的目录。访问点将指定的文件系统路径作为您文件系统的根目录，向使用该访问点的应用程序公开。使用接入点的 NFS 客户端只能访问接入点的 RootDirectory 及其子目录中的数据。

目录

CreationInfo

(可选) 指定 POSIX ID 和权限以应用到访问点的 RootDirectory。如果指定的 RootDirectory > Path 不存在，当客户端连接到访问点时，EFS 使用 CreationInfo 设置创建根目录。指定 CreationInfo 时，必须为所有属性提供值。

Important

如果您不提供 CreationInfo 并且指定的 RootDirectory > Path 不存在，则使用访问点挂载文件系统的尝试将失败。

类型：[CreationInfo](#) 对象

必需：否

Path

指定 EFS 文件系统上要作为根目录公开的路径，NFS 客户端使用访问点来访问 EFS 文件系统。一个路径最多可以有四个子目录。如果指定的路径不存在，则需要提供 CreationInfo。

类型：字符串

长度约束：最小长度为 1。最大长度为 100。

模式：`^(\\|\\(?!\\.)([^\s#<> ;`|&?{}^*\\/\\n])+){1,4})$`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

Tag

标签是键值对。允许的字符包括字母、空格、可采用 UTF-8 格式表示的数字以及下列字符： + - = . _ : /。

目录

Key

标签键 (字符串)。键不能以 aws: 开头。

类型：字符串

长度限制：长度下限为 1。最大长度为 128。

模式：`^(?![aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_.:/=+\-@]+)$`

必需：是

Value

标签键的值。

类型：字符串

长度约束：最大长度为 256。

模式：`^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- [适用于 C++ 的 AWS SDK](#)
- [适用于 Java V2 的 AWS SDK](#)
- [适用于 Ruby V3 的 AWS SDK](#)

文档历史记录

- API 版本：2015-02-01
- 上次文档更新日期：2025 年 6 月 9 日

下表介绍了 2018 年 7 月之后对 Amazon Elastic File System 用户指南的重要更改。如需有关文档更新的通知，您可以订阅 RSS 源。

变更	说明	日期
Support IPv6	Amazon EFS APIs 服务和挂载目标支持互联网协议版本 6 (IPv6)。有关 Amazon EFS API 的更多信息，请参阅 Amazon EFS API 。有关挂载目标的更多信息，请参阅 管理挂载目标 。	2025 年 6 月 9 日
一定会增加弹性吞吐量 AWS 区域	以下区域使用 Elastic 吞吐量的文件系统的最大读取吞吐量已从 20 增加 GiBps 到 60 GiBps：亚太地区（孟买）、亚太地区（首尔）、亚太地区（悉尼）和欧洲（伦敦）。有关更多信息，请参阅 您可以提高的 Amazon EFS 配额 。	2025 年 5 月 2 日
一定会增加弹性吞吐量 AWS 区域	在特定区域使用弹性吞吐量的文件系统的最大读取吞吐量为 60 GiBps。在所有其他区域，使用 Elastic 吞吐量的文件系统的读取吞吐量限制已从 10 提高到 20 GiBps。有关更多信息，请参阅 您可以提高的 Amazon EFS 配额 。	2025 年 4 月 10 日

增加了接入点配额	单个文件系统可以拥有的最多接入点数量已从 1,000 个增加到 10,000 个。您还可以请求提高此限制。有关更多信息，请参阅 您可以提高的资源配额 。	2025 年 2 月 10 日
提高了备份和恢复速率	执行备份和恢复的速度有所提高。有关更多信息，请参阅 备份性能 。	2025 年 1 月 8 日
根据请求增加了 IOPS 配额	现在，您可以为采用弹性吞吐量模式的 EFS 文件系统请求提高 IOPS。有关更多信息，请参阅 您可以提高的 Amazon EFS 配额 。	2024 年 11 月 26 日
支持跨账户复制	Amazon EFS 支持跨账户复制。有关更多信息，请参阅 跨 AWS 账户复制 EFS 文件系统 。	2024 年 11 月 19 日
更新了现有的 AWS 托管策略	已将 Replicati onRead 、 Replicati onWrite 、 iam:PassR ole 添加到 AmazonEla sticFileSystemFull Access 策略中。有关更多 信息，请参阅 AmazonEla sticFileSystemFullAccess 。	2024 年 11 月 7 日

更新了现有的 AWS 托管策略	Replicati onRead 并Replicati onWrite 添加到AmazonEla sticFileSystemServ iceRolePolicy 。有关更 多信息，请参阅 AmazonEla sticFileSystemServiceRolePo licy 。	2024 年 11 月 7 日
更新了现有的 AWS 托管策略	已将 ReplicationRead 权限添加到现有 AmazonEla sticFileSystemRead OnlyAccess 策略中。有关 更多信息，请参阅 AmazonEla sticFileSystemReadOnlyAcces s 。	2024 年 11 月 7 日
提高了弹性吞吐量限额	特定 AWS 区域 区域的弹性吞 吐量限制已提高到每秒 60 吉 字节 (GiBps)，所有其他区域 的弹性吞吐量限制已提高到 10 GiBps Gibibytes。有关更多信 息，请参阅 每个 AWS 区域中 所有连接的客户端的默认弹性 吞吐量总计 。	2024 年 10 月 14 日
更新了现有的 AWS 托管策略	可选 Sid (声明 ID) 元素 现已包含在 AmazonEla sticFileSystemRead OnlyAccess 策略声明 中。Sid 值为 ElasticFi leSystemReadOnlyAc cess 。有关 Sid 策略元素的 更多信息，请参阅 IAM JSON 策略元素 : Sid 。	2024 年 8 月 7 日

提高了弹性吞吐量限额	已提高特定 AWS 区域的弹性吞吐量限额。有关更多信息，请参阅 每个 AWS 区域中所有连接的客户端的默认弹性吞吐量总计 。	2024 年 7 月 31 日
提高了挂载目标的限额	每个虚拟私有云 (VPC) 的最大挂载目标数量从 400 个增加到 1400 个。有关更多信息，请参阅 您无法更改的 Amazon EFS 资源配额 。	2024 年 5 月 15 日
提高了弹性文件系统的合并吞吐量限额	MiBps 对于使用弹性吞吐量且使用版本 2.0 或更高版本的 Amazon EFS 客户端 (amazo n-efs-utils 版本) 或 Amazon EFS CSI 驱动程序 (aws-efs-csi-driver) 装载的文件系统，最大读写吞吐量总和为 1,500。有关更多信息，请参阅 Amazon EFS performance 中的性能摘要表。	2024 年 4 月 30 日
提高了弹性吞吐量限额	已提高特定 AWS 区域的弹性吞吐量限额。有关更多信息，请参阅 每个 AWS 区域中所有连接的客户端的默认弹性吞吐量总计 。	2024 年 3 月 13 日
提高 IOPS	使用弹性吞吐量的文件系统对于不频繁访问的数据，最高可以驱动 90000 次读取。有关更多信息，请参阅 性能摘要 。	2024 年 1 月 22 日

更新了现有的 AWS 托管策略	elasticfilesystem: UpdateFileSystemServiceProtection 已向现有 AmazonElasticFileSystemFullAccess 策略添加权限，允许委托人更新文件系统的保护。有关更多信息，请参阅 Amazon EFS 对 AWS 托管策略的更新。	2023 年 11 月 27 日
复制到现有文件系统	现在可以将文件系统复制到现有文件系统，这样可以更轻松地在文件系统之间同步更改以实现失效自动恢复。有关更多信息，请参阅 目标文件系统。	2023 年 11 月 27 日
添加了文件系统保护功能	复制覆盖保护功能已添加到文件系统中，并且默认处于启用状态。该保护功能可防止在复制配置中将文件系统用作目标。有关更多信息，请参阅 文件系统保护。	2023 年 11 月 27 日
新的存储类、文件系统类型和生命周期策略	Amazon EFS 现在提供 EFS 归档存储类、文件系统类型和“转换为归档”生命周期策略。有关更多信息，请参阅 文件系统类型和存储类。	2023 年 11 月 26 日
提高 IOPS	对于不频繁访问的数据，弹性吞吐量文件系统现在最多支持 65000 次读取操作 IOPS 和 50000 次写入操作 IOPS；而对于频繁访问的数据，现在支持 250000 次读取 IOPS。有关更多信息，请参阅 性能摘要。	2023 年 11 月 26 日

从源文件系统中删除复制配置	现在可以从源文件系统中删除复制配置。有关更多信息，请参阅 删除复制配置 。	2023 年 9 月 19 日
添加了其他 AWS 区域支持	Amazon EFS 现已向以色列（特拉维夫）区域的所有用户推出。	2023 年 8 月 7 日
通用模式文件系统的性能提升	Amazon EFS 通用模式文件系统现在支持每秒多达 5.5 万次读取操作和 2.5 万次写入操作。有关更多信息，请参阅 Amazon EFS 文件系统的配额 。	2023 年 8 月 3 日
提高了预置吞吐量限额	具体 AWS 区域而言，预配置吞吐量限制已提高。有关更多信息，请参阅 每个 AWS 区域客户端中所有连接的客户端的默认预配置吞吐量总计 。	2023 年 6 月 21 日
扩展了对 EFS 复制的区域支持	EFS 复制现在适用于所有 AWS 区域 可用 EFS 的地方。有关更多信息，请参阅 Amazon EFS 复制 。	2023 年 4 月 28 日
提高了弹性吞吐量限额	已提高特定 AWS 区域的弹性吞吐量限额。有关更多信息，请参阅表格， 每个客户端中所有连接的客户端的默认弹性吞吐量总计 AWS 区域 。	2023 年 4 月 17 日
弹性取代突发成为默认吞吐量模式	文件系统的默认（也是推荐的）吞吐量模式现在是“弹性”而不是“突发”。有关更多信息，请参阅 吞吐量模式 。	2023 年 4 月 13 日

添加了其他 AWS 区域 支持	Amazon EFS 现已向亚太地区（墨尔本）区域中的所有用户推出。	2023 年 4 月 12 日
增加了对 macOS Ventura 的支持	Amazon EFS 现在可以安装在运行在 macOS Ventura 上的 EC2 Mac 实例上。有关更多信息，请参阅 支持的分配 。	2023 年 4 月 10 日
添加了其他 AWS 区域 支持	Amazon EFS 现已向亚太地区（海得拉巴）区域中的所有用户推出。	2023 年 2 月 16 日
添加了其他 AWS 区域 支持	Amazon EFS 现已向欧洲（西班牙）AWS 区域中的所有用户推出。	2023 年 1 月 19 日
增加了文件系统的接入点限制	单个文件系统可以拥有的最多接入点数量已从 120 个增加到 1 千个。有关更多信息，请参阅 资源配额 。	2023 年 1 月 17 日
添加了其他 AWS 区域 支持	Amazon EFS 现已向欧洲（苏黎世）的所有用户开放 AWS 区域。	2022 年 12 月 15 日
增加了对一日生命周期策略的支持	现在，您可以为“转换为 IA”生命周期策略选择一天。有关更多信息，请参阅 使用生命周期策略 。	2022 年 11 月 27 日
减少了读写延迟时间	对于单区存储文件系统和标准存储文件系统，减少了文件数据读取和写入的延迟时间。有关更多信息，请参阅 性能摘要 。	2022 年 11 月 27 日

添加了更多吞吐量模式	已添加弹性吞吐量模式作为 Amazon EFS 文件系统的吞吐量选项。有关更多信息，请参阅 Elastic throughput 。	2022 年 11 月 27 日
添加了其他 AWS 区域支持	Amazon EFS 现已向中东（阿联酋）区域中的所有用户推出。	2022 年 10 月 17 日
增加了对 EFS 复制的支持	Amazon EFS 已删除之前的限制，即 EFS 复制不支持套接字和命名管道，或 FIFOs。	2022 年 9 月 15 日
增加了每个连接的文件锁数量限制	将每个连接的文件锁数量限制从 8192 个增加到 65536 个。有关更多信息，请参阅 NFS 客户端的配额 。	2022 年 5 月 4 日
移除了对使用文件锁的进程的限制	Amazon EFS 删除了之前的限制，即单个实例上最多有 256 个进程可以同时使用文件锁。有关更多信息，请参阅 NFS 客户端的配额 。	2022 年 5 月 4 日
添加了其他 AWS 区域支持	Amazon EFS 现已向亚太地区（雅加达）AWS 区域中的所有用户推出。	2022 年 1 月 27 日
增加了对 EFS 复制的支持	使用 EFS 复制将 EFS 文件系统上的数据和元数据复制到您选择 AWS 区域的另一个 EFS 文件系统。有关更多信息，请参阅 Amazon EFS 复制 。	2022 年 1 月 25 日

[文件系统和挂载目标资源使用 17 个字符的资源 ID 格式](#)

现在，新的 Amazon EFS 文件系统和挂载目标资源分配了 17 个字符 IDs。有关更多信息，请参阅 [Working with Amazon EFS resources](#)。

2021 年 10 月 22 日

[增加了对 EFS 智能分层的支持](#)

EFS 智能分层使用 EFS 生命周期管理来监控文件访问模式，旨在自动将文件转换为相应的不频繁访问（IA）存储类，以及从相应的不频繁访问（IA）存储类转换成文件。有关更多信息，请参阅 [EFS 智能分层和生命周期管理](#)。

2021 年 9 月 2 日

[增加了对测试 17 字符资源 ID 格式的支持](#)

2021 年 10 月 1 日，Amazon EFS IDs 将从使用 8 IDs 个字符的文件系统和挂载目标过渡到 17 个字符。在此过渡期间，您可以选择加入并开始 AWS 区域 逐个使用 17 个字符 IDs 的资源。有关更多信息，请参阅 [Resource IDs](#)。

2021 年 5 月 5 日

[增加了对使用 Amazon EFS 挂载帮助程序从不同可用区挂载单区文件系统的支持](#)

现在，可以使用 EFS 挂载帮助程序将使用单区存储类的 Amazon EFS 文件系统挂载到位于不同可用区的 EC2 实例。可以使用新 az 选项指定 Amazon EFS 文件系统的可用区。有关更多信息，请参阅 [挂载具有单区存储类的文件系统](#)。

2021 年 4 月 6 日

[增加了对 EFS 单区存储类的支持](#)

Amazon EFS 单区存储类以冗余方式将数据存储在 AWS 区域中的单个可用区中。EFS 单区和单区不频繁访问 (单区 IA) 存储类是一种经济实惠的选择，用于存储不需要 EFS 标准和标准 IA 存储类的多可用区弹性的数据。有关更多信息，请参阅[使用 EFS 存储类](#)。

2021 年 3 月 9 日

[添加了其他 AWS 区域支持](#)

Amazon EFS 现已向亚太地区 (大阪) AWS 区域中的所有用户推出。

2021 年 3 月 3 日

[增加了对运行 macOS Big Sur 的 Amazon EC2 macOS 实例的支持](#)

现在，可以使用 EFS 挂载帮助程序或使用 NFS 挂载命令从运行 macOS Big Sur 的 EC2 macOS 实例挂载您的 Amazon EFS 文件系统。有关更多信息，请参阅[使用 EFS 挂载帮助程序挂载或不使用 EFS 挂载帮助程序挂载文件系统](#)。

2021 年 2 月 23 日

[全新 Amazon EFS 控制台已在各 AWS GovCloud \(US\) 地区推出](#)

新的 Amazon EFS 控制台现已在中推出 AWS GovCloud (US) AWS 区域。

2021 年 2 月 10 日

[增加了对新的 Amazon EFS CloudWatch 指标的支持 MeteredIOBytes](#)

可以使用 MeteredIOBytes 来测量每个文件系统操作 (包括数据读取、数据写入和元数据操作) 的计量字节数。读取操作以其他操作三分之一的速率计量。有关更多信息，请参阅[Amazon EFS 的亚马逊 CloudWatch 指标](#)。

2021 年 1 月 28 日

[Amazon EFS 将文件系统的读取吞吐量增加了 300%](#)

Amazon EFS 文件系统现在以其他请求的三分之一速率计量读取请求。

2021 年 1 月 28 日

[增加了对新的 Amazon EFS CloudWatch 指标的支持 StorageBytes](#)

可以使用 StorageBytes 来测量和监控文件系统的大小（以字节为单位），包括存储在标准和不频繁访问存储类中的数据量。有关更多信息，请参阅 [Amazon EFS 的亚马逊 CloudWatch 指标](#)。

2021 年 1 月 11 日

[用于访问 AWS Transfer Family Amazon EFS 文件系统](#)

您可以使用将文件传 AWS Transfer Family 入和传出您的 Amazon EFS 文件系统。有关更多信息，请参阅[使用 AWS Transfer Family 访问您的 EFS 文件系统](#)中的文件。

2021 年 1 月 8 日

[用于管理 AWS Systems Manager Amazon EFS 客户端 \(amazon-efs-utils\)](#)

您可以使用 AWS Systems Manager 在您的 EC2 实例上自动安装或更新 Amazon EFS 客户端 (amazon-efs-utils)。有关更多信息，请参阅[使用 S AWS ystems Manager 自动安装或更新 Amazon EFS 客户端](#)。

2020 年 9 月 29 日

[强制创建加密 EFS 文件系统](#)

您可以使用 elasticfilesystem:Encrypted AWS Identity and Access Management (IAM) 条件密钥强制用户创建静态加密的 Amazon EFS 文件系统。有关更多信息，请参阅[强制创建静态加密的 Amazon EFS 文件系统](#)。

2020 年 9 月 16 日

Amazon EFS 每客户端吞吐量增加了 100%	EFS 现在支持每台客户机高达 500 MB/s 的吞吐量，比之前的 250 MB/s 限制提高了 100%。有关更多信息，请参阅 Amazon EFS 文件系统的配额 。	2020 年 7 月 23 日
增加了对每日自动备份 Amazon EFS 文件系统的支持	使用 EFS 控制台创建文件系统时，现在默认情况下启用每日自动备份。有关更多信息，请参阅 AWS Backup 与 Amazon EFS 配合使用 。	2020 年 7 月 16 日
全新的快速创建工作流简化了 Amazon EFS 文件系统的创建过程	使用 EFS 控制台中的“快速创建”选项，只需按一下按钮，即可使用服务推荐设置创建 EFS 文件系统。有关更多信息，请参阅 创建 EFS 文件系统 。	2020 年 7 月 16 日
现已推出全新的 Amazon EFS 控制台	全新的 EFS 控制台让您可以更轻松地使用 Amazon EFS，并简化了 EFS 文件系统的管理。	2020 年 7 月 16 日
Amazon EFS 增加了文件系统的最低吞吐量	使用突增吞吐量的 Amazon EFS 文件系统现在的最低吞吐量为 1 MiB/s。有关更多信息，请参阅 吞吐量模式 。	2020 年 6 月 30 日
通用模式文件系统的性能增加	Amazon EFS 通用模式文件系统现在支持每秒最高 3.5 万次读取操作，比之前 7 千次的限制提高了 400%。有关更多信息，请参阅 Amazon EFS 文件系统的配额 。	2020 年 4 月 1 日
添加了其他 AWS 区域支持	Amazon EFS 现已向北京和宁夏的所有用户开放 AWS 区域。	2020 年 1 月 22 日

[增加了对 NFS 客户端的 IAM 授权的支持](#)

现在，您可以使用 AWS Identity and Access Management (IAM) 来管理对 Amazon EFS 文件系统的 NFS 访问权限。有关更多信息，请参阅[使用 AWS IAM 控制对 Amazon EFS 的 NFS 访问权限](#)。

2020 年 1 月 13 日

[增加了对 EFS 接入点的支持](#)

Amazon EFS 接入点是 Amazon EFS 文件系统中特定于应用程序的入口点，便于管理应用程序对共享数据集的访问。有关更多信息，请参阅[使用 Amazon EFS 接入点](#)。

2020 年 1 月 13 日

[添加了 Support 对 AWS Backup 部分还原的支持。](#)

除了还原完整的恢复点之外，现在还可以使用部分还原来还原特定的文件和目录。有关更多信息，请参阅[AWS Backup 与 Amazon EFS 配合使用](#)。

2020 年 1 月 13 日

[增加了对 IAM 服务相关角色的支持](#)

Amazon EFS 现在使用基于 IAM 的服务相关角色，从而通过自动添加必要的权限来更轻松地设置 EFS。有关更多信息，请参阅[使用适用于 Amazon EFS 的服务相关角色](#)。

2019 年 12 月 10 日

[添加了其他 AWS 区域支持](#)

Amazon EFS 现已向欧洲 (斯德哥尔摩) 的所有用户开放 AWS 区域。

2019 年 11 月 20 日

[添加了其他 AWS 区域支持](#)

Amazon EFS 现已向亚太地区 (香港 AWS 区域) 的所有用户开放。

2019 年 11 月 20 日

添加了其他 AWS 区域 支持	Amazon EFS 现已向南美洲（圣保罗）的所有用户开放 AWS 区域。	2019 年 11 月 20 日
添加了其他 AWS 区域 支持	Amazon EFS 现已可供中东（巴林 AWS 区域）的所有用户使用。	2019 年 11 月 20 日
添加了新的 7 天生命周期管理策略	生命周期管理现在有一个额外的策略，可在 7 天后将数据移动到经济高效的“不频繁访问”存储类。有关更多信息，请参阅 EFS 生命周期管理 。	2019 年 11 月 6 日
增加了对接口 VPC 端点的支持	您可以在虚拟私有云和 Amazon EFS 之间建立私有连接，以调用 EFS API。有关更多信息，请参阅 使用 VPC 终端节点 。	2019 年 10 月 22 日
在启动新的 EC2 实例时，装载 EFS 文件系统。	现在，您可以在 EC2 启动实例向导中配置新的 Amazon EC2 实例以在启动时装载 EFS 文件系统。有关更多信息，请参阅 步骤 2。创建您的 EC2 资源并启动您的 EC2 实例 。	2019 年 10 月 17 日
增加了对服务配额的支持	现在，您可以在“服务配额”控制台中查看所有 Amazon EFS 限制。有关更多信息，请参阅 Amazon EFS 限制 。	2019 年 9 月 10 日
添加了新的生命周期管理策略	使用生命周期管理时，您现在可以从四个生命周期策略中选择一个来定义何时将文件转换为经济高效的“不常访问”存储类别。有关更多信息，请参阅 EFS 生命周期管理 。	2019 年 7 月 9 日

EFS 生命周期管理现在可用于所有 EFS 文件系统。	EFS 生命周期管理功能现在可用于所有 EFS 文件系统。现在已删除基于何时创建文件系统的先前限制。有关更多信息，请参阅 EFS 生命周期管理 。	2019 年 7 月 9 日
添加了其他 AWS 区域支持	Amazon EFS 现已向欧洲 (巴黎) 的所有用户开放 AWS 区域。	2019 年 6 月 12 日
添加了其他 AWS 区域支持	Amazon EFS 现已向亚太地区 (孟买) 的所有用户开放 AWS 区域。	2019 年 6 月 5 日
添加了其他 AWS 区域支持	Amazon EFS 现已向加拿大 (中部) 的所有用户开放 AWS 区域。	2019 年 5 月 1 日
API 更新：标签现在是 CreateFileSystem 操作负载的一部分	现在，您可以在使用 AWS API 和 CLI CreateFileSystem 操作创建 Amazon EFS 文件系统时添加标签。有关更多信息，请参阅 CreateFileSystem 和 使用 AWS CLI 创建文件系统 。	2019 年 2 月 19 日
新功能：EFS 不频繁访问存储类和 EFS 生命周期管理	Amazon EFS 不频繁访问是针对不常访问的文件进行了成本优化的存储类。EFS 生命周期管理自动将文件从 Standard 转换为 Infrequent Access 存储。有关更多信息，请参阅 EFS 存储类 。	2019 年 2 月 13 日
添加了其他 AWS 区域支持	Amazon EFS 现已向欧洲 (伦敦) 的所有用户开放 AWS 区域。	2019 年 1 月 23 日

[AWS Backup 与 Amazon EFS 的服务集成](#)

Amazon EFS 文件系统可以使用 AWS Backup 完全托管、集中化、自动化的备份服务进行备份，用于备份云端和本地 AWS 服务的数据。有关更多信息，请参阅 [AWS Backup 和 Amazon EFS](#)。

2019 年 1 月 16 日

[添加了中转网关连接对本地存储系统的支持。](#)

Amazon EFS 文件系统现在可通过与本地存储系统的中转网关连接来访问。有关更多信息，请参阅[从另一个账户或 VPC 挂载](#)和[演练：从不同的 VPC 挂载文件系统](#)。

2018 年 12 月 6 日

[EFS 文件同步现已成为新 AWS DataSync 服务的一部分。](#)

AWS DataSync 是一项托管数据传输服务，可简化本地存储系统和 AWS 存储服务之间大量数据的同步。有关更多信息，请参阅[使用将文件从本地文件系统传输到 Amazon EFS AWS DataSync](#)。

2018 年 11 月 26 日

[增加了对 VPN 和区域间 VPC 对等连接的支持](#)

Amazon EFS 现在可通过 VPN 连接和区域间 VPC 对等连接进行访问。有关更多信息，请参阅[使用将文件从本地文件系统传输到 Amazon EFS AWS DataSync](#)。

2018 年 10 月 23 日

[增加了对 VPN 和区域间 VPC 对等连接的支持](#)

Amazon EFS 文件系统现在可通过 VPN 连接和区域间 VPC 对等连接进行访问。有关更多信息，请参阅[从其他账户或 VPC 挂载](#)以及 [Amazon EFS 如何与 Direct Connect 配合使用](#)和 [VPNs](#)。

2018 年 10 月 23 日

添加了其他 AWS 区域 支持	Amazon EFS 现已向亚太地区（新加坡）AWS 区域的所有用户推出。	2018 年 7 月 13 日
引入预置吞吐量模式	您现在可以使用新的预置吞吐量模式为新文件系统或现有文件系统配置吞吐量。有关更多信息，请参阅 吞吐量模式 。	2018 年 7 月 12 日
添加了其他 AWS 区域 支持	Amazon EFS 现已向亚太地区（东京）AWS 区域中的所有用户推出。	2018 年 7 月 11 日

下表介绍了 2018 年 7 月之前对 Amazon Elastic File System 用户指南的重要更改。

更改	描述	更改日期
添加了其他 AWS 区域 支持	Amazon EFS 现已向亚太地区（首尔）AWS 区域的所有用户推出。	2018 年 5 月 30 日
增加了 CloudWatch 公制数学支持	指标数学使您可以查询多个 CloudWatch 指标，并使用数学表达式根据这些指标创建新的时间序列。有关更多信息，请参阅 将公制数学与 CloudWatch 指标配合使用 。	2018 年 4 月 4 日
添加了 amazon-efs-utils 开源工具集，并添加了传输中加密	amazon-efs-utils 工具是一组开源可执行文件，可以简化使用 Amazon EFS 的各种操作，例如挂载。使用无需支付额外费用amazon-efs-utils，您可以从中下载这些工具 GitHub。有关更多信息，请参阅 安装 Amazon EFS 客户端 。 同样，在该版本中，Amazon EFS 现在支持加密通过传输层安全性协议（TLS）隧道传输的数据。有关更多信息，请参阅 Amazon EFS 中的数据加密 。	2018 年 4 月 4 日

更改	描述	更改日期
更新了每个文件系统的限制 AWS 区域	Amazon EFS 增加了所有 AWS 区域中所有账户的文件系统数限制。有关更多信息，请参阅 您无法更改的 Amazon EFS 资源配额 。	2018 年 3 月 15 日
添加了其他 AWS 区域支持	Amazon EFS 现已向美国西部（加利福尼亚北部）的所有用户开放 AWS 区域。	2018 年 3 月 14 日
静态数据加密	Amazon EFS 现在支持静态数据加密。有关更多信息，请参阅 Amazon EFS 中的数据加密 。	2017 年 8 月 14 日
支持更多区域	Amazon EFS 现已向欧洲地区（法兰克福）区域中的所有用户提供。	2017 年 20 月 7 日
使用域名系统 (DNS) 的文件系统名称	Amazon EFS 现在支持文件系统使用 DNS 名称。文件系统的 DNS 名称自动解析为连接的 Amazon EC2 实例的可用区中挂载目标的 IP 地址。有关更多信息，请参阅 使用 DNS 名称在 Amazon EC2 上挂载 。	2016 年 12 月 20 日
增加了文件系统支持的标签数量	Amazon EFS 现在支持每个文件系统 50 个标签。有关 Amazon EFS 中标签的更多信息，请参阅 标记 EFS 资源 。	2016 年 8 月 29 日
正式发布	Amazon EFS 现已向美国东部（弗吉尼亚州北部）、美国西部（俄勒冈）和欧洲地区（爱尔兰）区域中的所有用户正式推出。	2016 年 6 月 28 日
文件系统限制提升	每个 AWS 区域内的每个账户可以创建的 Amazon EFS 文件系统数量从 5 个提高到 10 个。	2015 年 8 月 21 日
更新了入门练习	入门练习经过更新，以简化入门过程。	2015 年 8 月 17 日
新指南	这是《Amazon Elastic File System 用户指南》的首个版本。	2015 年 5 月 26 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。