

---

# Amazon Elastic File System

## 用户指南



## Amazon Elastic File System: 用户指南

Copyright © 2018 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

# Table of Contents

什么是 Amazon Elastic File System ? .....	1
您是 Amazon EFS 的新用户吗 ? .....	1
工作原理 .....	3
概述 .....	3
Amazon EFS 如何与 Amazon EC2 协同工作 .....	4
Amazon EFS 如何与 AWS Direct Connect 协同工作 .....	4
实现摘要 .....	5
身份验证和访问控制 .....	6
Amazon EFS 中的数据一致性 .....	6
设置 .....	7
注册 AWS .....	7
创建 IAM 用户 .....	7
入门 .....	9
假设 .....	9
相关主题 .....	9
第 1 步：创建您的 EC2 资源并启动您的 EC2 实例 .....	9
第 2 步：创建您的 Amazon EFS 文件系统 .....	12
第 3 步：连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统 .....	13
步骤 4：使用 EFS 文件同步将文件从现有的文件系统同步到 Amazon EFS .....	14
步骤 5：清理资源并保护您的 AWS 账户 .....	15
为 Amazon EFS 创建资源 .....	16
创建文件系统 .....	16
要求 .....	17
所需权限 .....	17
创建文件系统 .....	17
创建挂载目标 .....	20
使用 Amazon EFS 控制台创建挂载目标 .....	21
使用 AWS CLI 创建挂载目标 .....	24
创建安全组 .....	24
使用 AWS 管理控制台创建安全组 .....	25
使用 AWS CLI 创建安全组 .....	25
使用文件系统 .....	26
相关主题 .....	26
NFS 级别用户、组和权限 .....	26
示例 Amazon EFS 文件系统使用案例和权限 .....	27
对文件系统中的文件和目录的用户和组 ID 权限 .....	27
无根 Squash .....	28
权限缓存 .....	28
更改文件系统对象所有权 .....	28
Amazon EFS 文件同步 .....	28
EFS 文件同步要求 .....	29
EFS 文件同步架构 .....	31
EFS 文件同步如何传输文件 .....	32
使用 amazon-efs-utils .....	34
概述 .....	34
在 Amazon Linux 上安装 amazon-efs-utils 软件包 .....	35
在其他 Linux 发行版上安装 amazon-efs-utils 软件包 .....	35
升级 stunnel .....	36
EFS 挂载帮助程序 .....	37
如何使用 .....	37
使用 EFS 挂载帮助程序 .....	38
获取支持日志 .....	38
将 amazon-efs-utils 与 AWS Direct Connect 一起使用 .....	38
相关主题 .....	38

管理文件系统	39
管理网络可访问性	39
在 VPC 中创建或删除挂载目标	41
在另一个 VPC 中创建挂载目标	43
更新挂载目标配置	44
管理标签	45
使用控制台	45
使用 AWS CLI	46
计量文件系统和对象大小	46
计量 Amazon EFS 文件系统对象	46
计量 Amazon EFS 文件系统	47
管理 EFS 文件同步	47
删除同步代理	47
删除同步任务	48
了解同步代理状态	48
了解同步任务状态	48
在 EFS 文件同步虚拟机本地控制台上执行任务	49
在 Amazon EC2 本地控制台上为文件同步执行维护任务	54
删除文件系统	56
使用控制台	56
使用 CLI	57
相关主题	57
管理对加密的文件系统的访问	57
对 Amazon EFS 客户主密钥执行管理操作	57
相关主题	58
挂载文件系统	59
AMI 和内核版本故障排除	59
安装 amazon-efs-utils	59
使用 EFS 挂载帮助程序进行挂载	59
使用 EFS 挂载帮助程序在 EC2 上挂载	59
在本地 Linux 客户端上使用 EFS 挂载帮助程序通过 AWS Direct Connect 挂载	60
自动挂载	61
将现有 EC2 实例更新为自动挂载	61
将 EFS 文件系统配置为在 EC2 实例启动时自动挂载	62
其他挂载注意事项	63
卸载文件系统	63
监控文件系统	65
监控工具	65
自动化工具	65
手动监控工具	66
监控 CloudWatch	66
Amazon EFS 的 Amazon CloudWatch 指标	66
在 CloudWatch 中报告的字节数	68
Amazon EFS 维度	68
如何使用 Amazon EFS 指标？	69
监控 EFS 文件同步	69
访问 CloudWatch 指标	70
创建警报	70
将指标数学与 Amazon EFS 一起使用	71
使用 AWS CloudTrail 记录 Amazon EFS API 调用	74
CloudTrail 中的 Amazon EFS 信息	74
了解 Amazon EFS 日志文件条目	75
静态加密的文件系统的 Amazon EFS 日志文件条目	76
性能	78
性能概述	78
Amazon EFS 使用案例	78
大数据与分析	78

媒体处理 workflows .....	79
内容管理和 Web 服务 .....	79
主目录 .....	79
将文件系统同步到 Amazon EFS .....	79
性能模式 .....	79
通用性能模式 .....	79
最大 I/O 性能模式 .....	79
使用合适的性能模式 .....	79
吞吐量模式 .....	80
激增 .....	80
预置吞吐量 .....	82
使用合适的吞吐量模式 .....	82
本地性能注意事项 .....	82
针对高可用性设计 .....	82
Amazon EFS 性能提示 .....	83
相关主题 .....	83
安全性 .....	84
API 调用的 AWS Identity and Access Management (IAM) 权限 .....	84
Amazon EC2 实例和挂载目标的安全组 .....	84
挂载 Amazon EFS 文件系统的安全注意事项 .....	85
EFS 文件和目录的读取、写入和执行权限 .....	86
源端口 .....	86
在 EFS 中加密数据和元数据 .....	86
何时使用加密 .....	86
加密传输中的数据 .....	86
静态加密数据 .....	88
相关主题 .....	89
限制 .....	90
您可以提高的 Amazon EFS 限制 .....	90
资源限制 .....	91
客户端 EC2 实例的限制 .....	91
Amazon EFS 文件系统的限制 .....	92
EFS 文件同步限制 .....	92
不支持的 NFSv4 功能 .....	92
其他注意事项 .....	93
Amazon EFS 故障排除 .....	94
排查一般问题 .....	94
Amazon EC2 实例挂起 .....	94
写入大量数据的应用程序挂起 .....	94
打开和关闭操作被序列化 .....	95
自定义 NFS 设置导致写入延迟 .....	95
使用 Oracle Recovery Manager 创建备份的速度很慢 .....	96
解决文件操作错误 .....	96
命令失败，并显示“Disk quota exceeded”错误 .....	96
命令失败，并显示“I/O error” .....	96
命令失败，并显示“File name is too long”错误 .....	97
命令失败，并显示“Too many links”错误 .....	97
命令失败，并显示“File too large”错误 .....	97
命令失败，并显示“Try again”错误 .....	97
解决 AMI 和内核问题 .....	97
无法更改所有权 .....	98
由于客户端错误，文件系统重复执行操作 .....	98
客户端发生死锁 .....	98
列出大型目录中的文件需要很长时间 .....	98
解决挂载问题 .....	99
在 Windows 实例上挂载文件系统失败 .....	99
自动挂载失败，并且实例没有响应 .....	99

在 /etc/fstab 中挂载多个 Amazon EFS 文件系统失败 .....	99
挂载命令失败，并显示“wrong fs type”错误消息 .....	100
挂载命令失败，并显示“incorrect mount option”错误消息 .....	100
在创建文件系统后文件系统挂载立即失败 .....	100
文件系统挂载挂起，然后失败，并显示超时错误 .....	101
使用 DNS 名称的文件系统挂载失败 .....	101
挂载目标生命周期状态停滞 .....	101
挂载没有响应 .....	101
针对新挂载的文件系统的操作返回“bad file handle”错误 .....	102
卸载文件系统失败 .....	102
排除加密故障 .....	102
具有传输中的数据加密的挂载失败 .....	103
具有传输中的数据加密的挂载中断 .....	103
无法创建静态加密的文件系统 .....	103
无法使用的加密文件系统 .....	103
EFS 文件同步故障排除 .....	105
本地源文件系统停滞在“正在挂载”状态 .....	105
Amazon EC2 源文件系统停滞在“正在挂载”状态 .....	105
同步任务停滞在“正在启动”状态 .....	106
同步任务失败，并显示“权限被拒绝”错误消息 .....	106
完成同步任务的“正在准备”状态需要多长时间？ .....	106
完成同步任务的“正在验证”状态需要多长时间？ .....	106
允许 AWS Support 帮助解决您的 EFS 文件同步问题 .....	107
允许 AWS Support 帮助解决您的 EC2 EFS 文件同步问题 .....	108
演练 .....	110
演练 1：使用 AWS CLI 创建和挂载文件系统 .....	110
开始前的准备工作 .....	110
设置工具 .....	111
步骤 1：创建 Amazon EC2 资源 .....	112
步骤 2：创建 Amazon EFS 资源 .....	115
步骤 3：挂载并测试文件系统 .....	117
步骤 4：清除 .....	120
演练 2：设置 Apache Web 服务器并提供文件服务 .....	121
提供文件的单个 EC2 实例 .....	121
提供文件服务的多个 EC2 实例 .....	123
演练 3：创建可写的每用户子目录 .....	126
重启时自动重新挂载 .....	127
演练 4：Amazon EFS 文件系统的备份解决方案 .....	127
演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 .....	127
开始前的准备工作 .....	128
步骤 1：创建您的 Amazon Elastic File System 资源 .....	128
步骤 2：下载并安装 amazon-efs-utils .....	129
步骤 3：在本地客户端上挂载 Amazon EFS 文件系统 .....	130
第 4 步：清理资源并保护您的 AWS 账户 .....	131
演练 6：在 Amazon EFS 文件系统上实施静态加密 .....	132
实施静态加密 .....	132
演练 7：使用 EFS 文件同步从本地同步文件 .....	134
开始前的准备工作 .....	134
步骤 1：创建同步代理 .....	134
步骤 2：创建同步任务 .....	135
步骤 3：将源文件系统同步到 Amazon EFS .....	137
步骤 4：访问文件 .....	138
第 5 步：清除 .....	138
演练 8：使用 EFS 文件同步将文件系统从 Amazon EC2 同步到 Amazon EFS .....	138
开始前的准备工作 .....	138
步骤 1：创建同步代理 .....	139
步骤 2：创建同步任务 .....	140

步骤 3：将源文件系统同步到 Amazon EFS .....	142
步骤 4：访问文件 .....	143
步骤 4：清除 .....	143
身份验证和访问控制 .....	144
身份验证 .....	144
访问控制 .....	145
访问管理概述 .....	145
Amazon Elastic File System 资源和操作 .....	145
了解资源所有权 .....	146
管理对资源的访问 .....	146
指定策略元素：操作、效果和委托人 .....	147
在策略中指定条件 .....	148
使用基于身份的策略 (IAM 策略) .....	148
使用 Amazon EFS 控制台所需要的权限 .....	149
适用于 Amazon EFS 的 AWS 托管 (预定义) 策略 .....	150
客户托管策略示例 .....	150
Amazon EFS API 权限参考 .....	151
Amazon EFS API .....	154
API 终端节点 .....	154
API 版本 .....	154
相关主题 .....	155
使用 Amazon EFS 的查询 API 请求速率 .....	155
轮询 .....	155
重试或批处理 .....	155
计算睡眠间隔 .....	155
Actions .....	155
CreateFileSystem .....	157
CreateMountTarget .....	164
CreateTags .....	171
DeleteFileSystem .....	174
DeleteMountTarget .....	176
DeleteTags .....	179
DescribeFileSystems .....	181
DescribeMountTargets .....	185
DescribeMountTargetSecurityGroups .....	188
DescribeTags .....	191
ModifyMountTargetSecurityGroups .....	194
UpdateFileSystem .....	197
Data Types .....	201
FileSystemDescription .....	202
FileSystemSize .....	205
MountTargetDescription .....	206
Tag .....	208
附加信息 .....	209
使用 AWS Data Pipeline 备份 .....	209
使用 AWS Data Pipeline 的 Amazon EFS 备份的性能 .....	210
使用 AWS Data Pipeline 的 Amazon EFS 备份的注意事项 .....	210
使用 AWS Data Pipeline 的 Amazon EFS 备份假设 .....	210
如何使用 AWS Data Pipeline 备份 Amazon EFS 文件系统 .....	211
其他备份资源 .....	216
在没有 EFS 挂载帮助程序的情况下挂载文件系统 .....	220
NFS 支持 .....	220
安装 NFS 客户端 .....	221
使用 DNS 名称在 Amazon EC2 上挂载 .....	222
使用 IP 地址挂载 .....	222
自动挂载 .....	223
文档历史记录 .....	226

# 什么是 Amazon Elastic File System ?

Amazon Elastic File System (Amazon EFS) 提供简单的可扩展文件存储以供与 Amazon EC2 配合使用。使用 Amazon EFS，存储容量会随着您添加和删除文件而自动弹性增长和收缩，因此您的应用程序可在需要时获得所需存储。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 支持网络文件系统版本 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 无缝融合。多个 Amazon EC2 实例可以同时访问 Amazon EFS 文件系统，为在多个实例或服务器上运行的工作负载和应用程序提供通用数据源。

有了 Amazon EFS，您仅需为文件系统使用的存储付费，无最低费用或设置费用。与预配置吞吐量有关的成本由与您指定的吞吐量值决定。有关更多信息，请参阅 [Amazon EFS 定价](#)。

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 文件系统将数据和元数据存储在一个 AWS 区域内的多个可用区中。EFS 文件系统可以扩展到 PB 级，提高吞吐量，并允许从 Amazon EC2 实例对您的数据进行大规模并行访问。

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 \(p. 6\)](#)。Amazon EFS 还允许您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 [安全性 \(p. 84\)](#)。

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 Amazon 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。有关更多信息，请参阅 [在 EFS 中加密数据和元数据 \(p. 86\)](#)。

Amazon EFS 旨在提供各种工作负载所需的吞吐量、IOPS 和低延迟。有了 Amazon EFS，您可以从两种性能模式和两种吞吐量模式中进行选择：

- 默认通用性能模式非常适合对延迟敏感的使用案例，如 Web 服务环境、内容管理系统、主目录和一般文件服务。最大 I/O 模式下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，但代价是稍高的文件操作延迟。有关更多信息，请参阅 [性能模式 \(p. 79\)](#)。
- 使用默认突发吞吐量模式，吞吐量随着文件系统的增长而扩展。使用预置吞吐量模式，您可以指定与存储的数据量无关的文件系统的吞吐量。有关更多信息，请参阅 [Amazon EFS 性能 \(p. 78\)](#)。

## Note

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

## 您是 Amazon EFS 的新用户吗？

如果您是首次接触 Amazon EFS 的用户，建议您按顺序阅读以下内容：

1. 有关 Amazon EFS 产品和定价概述，请参阅 [Amazon EFS](#)。
2. 有关 Amazon EFS 技术概述，请参阅 [Amazon EFS：工作原理 \(p. 3\)](#)。
3. 尝试入门练习：
  - [入门 \(p. 9\)](#)
  - [演练 \(p. 110\)](#)

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：



- [为 Amazon EFS 创建资源 \(p. 16\)](#)
- [管理 Amazon EFS 文件系统 \(p. 39\)](#)
- [Amazon EFS API \(p. 154\)](#)

# Amazon EFS : 工作原理

下面介绍了 Amazon EFS 的工作原理、其实施细节和安全注意事项。

## 主题

- [概述](#) (p. 3)
- [Amazon EFS 如何与 Amazon EC2 协同工作](#) (p. 4)
- [Amazon EFS 如何与 AWS Direct Connect 协同工作](#) (p. 4)
- [实现摘要](#) (p. 5)
- [身份验证和访问控制](#) (p. 6)
- [Amazon EFS 中的数据一致性](#) (p. 6)

## 概述

Amazon EFS 在 AWS 云中提供文件存储。使用 Amazon EFS，您可以创建文件系统，将文件系统挂载到 Amazon EC2 实例上，然后与文件系统之间读取和写入数据。您可以通过网络文件系统 4.0 和 4.1 版 (NFSv4) 协议在 VPC 中挂载 Amazon EFS 文件系统。

有关支持此协议的 Amazon EC2 Linux Amazon 系统映像 (AMI) 的列表，请参阅[NFS 支持](#) (p. 220)。我们建议使用当前一代 Linux NFSv4.1 客户端，如 Amazon Linux 和 Ubuntu AMI 中的客户端。对于某些 AMI，则需要安装 NFS 客户端以便将文件系统挂载到 Amazon EC2 实例上。有关说明，请参阅[安装 NFS 客户端](#) (p. 221)。

您可以从 Amazon VPC 中的 Amazon EC2 实例并发访问 Amazon 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。在同一区域内的多个可用区中运行的 Amazon EC2 实例可以访问文件系统，以便许多用户可以访问和共享通用数据源。

注意以下限制：

- 一次只能在一个 VPC 中的实例上挂载 Amazon EFS 文件系统。
- 文件系统和 VPC 必须位于同一个 AWS 区域。

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

要在 VPC 中访问 Amazon 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 Amazon EFS 文件系统的 NFSv4 终端节点的 IP 地址。您使用其 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。您可以在一个区域内的每个可用区中创建一个挂载目标。如果 VPC 内的可用区中有多个子网，则可以在其中一个子网中创建挂载目标，该可用区中的所有 EC2 实例都将共享该挂载目标。

挂载目标本身设计为具有高可用性。在设计应用程序以实现高可用性和故障转移到其他可用区的能力时，请记住，每个可用区中的挂载目标的 IP 地址和 DNS 都是静态的。

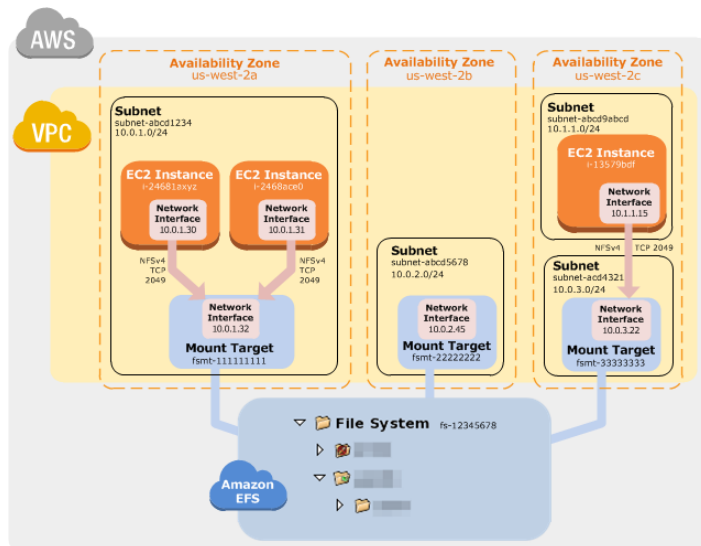
通过挂载目标挂载文件系统后，可以像使用任何其他符合 POSIX 标准的文件系统一样使用它。有关 NFS 级别的权限和相关注意事项的信息，请参阅[网络文件系统 \(NFS\) 级别用户、组和权限](#) (p. 26)。

使用 AWS Direct Connect 连接到 Amazon VPC 时，可以将 Amazon EFS 文件系统挂载到本地数据中心的服务器上。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 EFS。

Amazon EFS 文件系统可以通过 AWS Direct Connect 连接挂载到 Amazon EC2 实例或本地部署中。

## Amazon EFS 如何与 Amazon EC2 协同工作

下图显示了一个访问 Amazon EFS 文件系统的示例 VPC。在这里，VPC 中的 EC2 实例挂载有文件系统。



在此图中，VPC 有三个可用区，每个可用区中都创建了一个挂载目标。我们建议您从同一可用区内的挂载目标访问文件系统。请注意，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。创建此设置的方式如下所示：

1. 创建您的 Amazon EC2 资源并启动您的 Amazon EC2 实例。有关 Amazon EC2 的更多信息，请参阅 [Amazon EC2 - 虚拟服务器托管](#)。
2. 创建您的 Amazon EFS 文件系统。
3. 连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统。

有关详细步骤，请参阅[Amazon Elastic File System 入门 \(p. 9\)](#)。

## Amazon EFS 如何与 AWS Direct Connect 协同工作

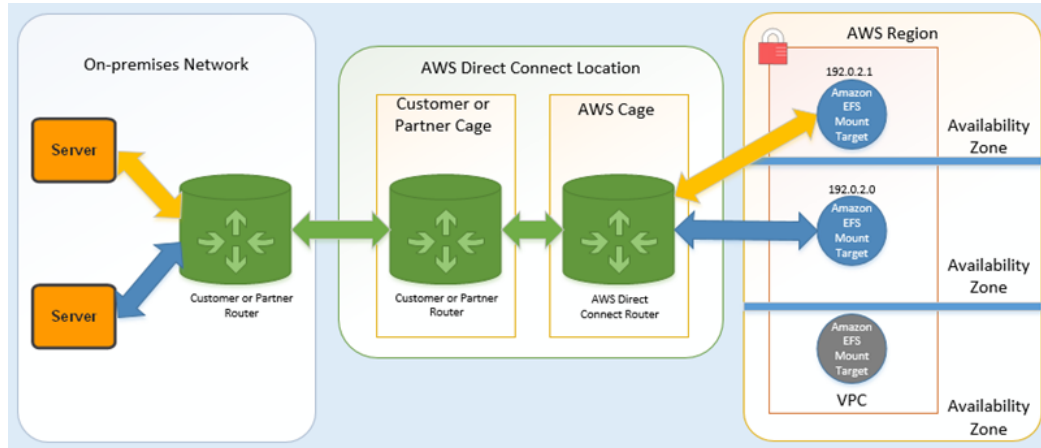
通过使用挂载在本地服务器上的 Amazon EFS 文件系统，可以将本地数据迁移到 Amazon EFS 文件系统中托管的 AWS 云中。您也可以利用突发，这意味着您可以将数据从本地服务器移到 Amazon EFS，并在 Amazon VPC 内的 Amazon EC2 实例队列中进行分析，然后将结果永久存储在文件系统中或将结果移回您的本地服务器。

在将 Amazon EFS 与 AWS Direct Connect 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。
- 不支持使用 AWS VPN 从本地服务器访问 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。请注意，将向您收取 AWS Direct Connect 与 Amazon VPC 的连接费用。有关更多信息，请参阅 [AWS 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。



只要使用您的本地服务器和 Amazon VPC 之间的 AWS Direct Connect 连接可以访问挂载目标的子网，就可以使用 VPC 中的任何一个挂载目标。要从本地服务器访问 Amazon EFS，您需要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。

要创建类似设置，您需要执行以下操作：

1. 在您的本地数据中心和 Amazon VPC 之间建立 AWS Direct Connect 连接。有关 AWS Direct Connect 的更多信息，请参阅 [AWS Direct Connect](#)。
2. 创建您的 Amazon EFS 文件系统。
3. 将 Amazon EFS 文件系统挂载在本地服务器上。

有关详细步骤，请参阅 [演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 \(p. 127\)](#)。

## 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，以及文件系统状态。有关更多信息，请参阅 [CreateFileSystem \(p. 157\)](#)。

Amazon EFS 还支持使用其他资源来配置主要资源，其中包括挂载目标和标签：

- 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。每个挂载目标都具有以下属性：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,wsz=1048576,hard,timeo=600,retrans=2,noresvport file-system-  
DNS-name:/ ~/efs-mount-point
```

有关更多信息，请参阅 [创建挂载目标 \(p. 20\)](#)。首先，您需要在 EC2 实例上安装 NFS 客户端。[入门 \(p. 9\)](#) 练习提供了分步说明。

- 标签 – 为了帮助组织文件系统，您可以将自己的元数据分配给您创建的每个文件系统。每个标签都是一个键-值对。

您可以将挂载目标和标签视为仅在与文件系统相关联时才存在的子资源。

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，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[入门 \(p. 9\)](#)。
- 使用 Amazon EFS 命令行界面 (CLI) – 有关示例，请参阅[演练 1：使用 AWS CLI 创建 Amazon EFS 文件系统并将其挂载到 EC2 实例上 \(p. 110\)](#)。
- 也可以通过编程方式管理这些资源，如下所示：
  - 使用 AWS 软件开发工具包 - AWS 软件开发工具包通过包装底层 Amazon EFS API 来简化编程任务。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](#)。
  - 直接从应用程序调用 Amazon EFS API - 如果由于某种原因无法使用软件开发工具包，您可以直接从应用程序调用 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 \(p. 154\)](#)。

## 身份验证和访问控制

您必须具有有效的凭证来发起 Amazon EFS API 请求，例如创建文件系统。此外，您还必须具有创建或访问资源的权限。默认情况下，当您使用 AWS 账户的根账户凭证时，可以创建和访问该账户拥有的资源。但是，我们不建议使用根账户凭证。此外，还必须向您在账户中创建的任何 AWS Identity and Access Management (IAM) 用户和角色授予创建或访问资源的权限。有关许可的更多信息，请参阅[Amazon EFS 的身份验证和访问控制 \(p. 144\)](#)。

## Amazon EFS 中的数据一致性

Amazon EFS 提供了应用程序期望从 NFS 获得的关闭后打开一致性语义。

在 Amazon EFS 中，写入操作在以下情况下将持久存储在可用区中：

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

根据访问模式，Amazon EFS 提供了比关闭后打开一致性语义更强大的一致性保证。执行同步数据访问和执行非附加写入的应用程序将具有写入后读取数据访问一致性。

# 设置

首次使用 Amazon EFS 前，请完成以下任务：

1. 注册 AWS (p. 7)
2. 创建 IAM 用户 (p. 7)

## 注册 AWS

当您注册 Amazon Web Services (AWS) 时，您的 AWS 账户会自动注册 AWS 中的所有服务，包括 Amazon EFS。您只需为使用的服务付费。

借助 Amazon EFS，您仅需为实际使用的存储付费。有关 Amazon EFS 使用费率的更多信息，请参阅 [Amazon Elastic File System 定价](#)。如果您是 AWS 新客户，还可以免费试用 Amazon EFS。有关更多信息，请参阅 [AWS 免费使用套餐](#)。

如果您已有一个 AWS 账户，请跳到下一个任务。如果您还没有 AWS 账户，请使用以下步骤创建。

### 创建 AWS 账户

1. 打开 <https://aws.amazon.com/>，然后选择 Create an AWS Account。

#### Note

如果您之前已登录 AWS 管理控制台，则可能无法在浏览器中执行此操作。在此情况下，请选择 Sign in to a different account，然后选择 Create a new AWS account。

2. 按照屏幕上的说明进行操作。

作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

请记住您的 AWS 账号，因为在下一个任务中您会用到它。

## 创建 IAM 用户

AWS 中的服务 (例如 Amazon EFS) 要求您在访问时提供凭证，以便服务可以确定您是否有权访问其资源。AWS 建议不要使用 AWS 账户的根凭证发起请求。而应创建一个 IAM 用户并授予该用户完全访问权限。我们将这些用户称为管理员用户。您可以使用管理员用户凭证而不是您账户的根凭证来与 AWS 交互和执行任务，例如创建存储桶、创建用户和为用户授予权限。有关更多信息，请参阅 AWS 一般参考 中的 [根账户凭证与 IAM 用户凭证](#) 和 IAM 用户指南中的 [IAM 最佳实践](#)。

如果您已注册 AWS 但尚未为自己创建一个 IAM 用户，则可以使用 IAM 控制台自行创建。

为您自己创建一个 IAM 用户并将该用户添加到管理员组

1. 使用 AWS 账户电子邮件地址和密码，以 [AWS 账户根用户](#) 身份登录到 IAM 控制台 (<https://console.aws.amazon.com/iam/>)。

#### Note

强烈建议您遵守以下使用 **Administrator** IAM 用户的最佳实践，妥善保存根用户凭证。只在执行少数 [账户和服务管理任务](#) 时才作为根用户登录。

2. 在控制台的导航窗格中，选择 **Users**，然后选择 **Add user**。
3. 对于 **User name**，键入 **Administrator**。
4. 选中 AWS 管理控制台 **access** 旁边的复选框，选择 **Custom password**，然后在文本框中键入新用户的密码。您可以选择 **Require password reset** (需要重置密码) 以强制用户在下次登录时创建新密码。
5. 选择 **Next: Permissions**。
6. 在设置权限页面上，选择将用户添加到组。
7. 选择 **Create group**。
8. 在 **Create group** (创建组) 对话框中，对于 **Group name** (组名称)，键入 **Administrators**。
9. 对于 **Filter policies** (筛选策略)，选中 **AWS managed - job function** (AWS 托管 - 工作职能) 的复选框。
10. 在策略列表中，选中 **AdministratorAccess** 的复选框。然后选择 **Create group**。
11. 返回到组列表中，选中您的新组所对应的复选框。如有必要，选择 **Refresh** 以在列表中查看该组。
12. 选择 **Next: Review** 以查看要添加到新用户的组成员资格的列表。如果您已准备好继续，请选择 **Create user**。

您可使用此相同的流程创建更多的组和用户，并允许您的用户访问 AWS 账户资源。要了解有关使用策略限制用户对特定 AWS 资源的权限的信息，请参阅[访问管理](#)和[示例策略](#)。

要以该新 IAM 用户的身份登录，请从 AWS 管理控制台退出，然后使用以下 URL，其中 `your_aws_account_id` 是您的不带连字符的 AWS 账号 (例如，如果您的 AWS 账号是 1234-5678-9012，则您的 AWS 账户 ID 是 123456789012)：

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

输入您刚创建的 IAM 用户名和密码。登录后，导航栏显示 `your_user_name@your_aws_account_id`。

如果您不希望您的登录页面 URL 包含 AWS 账户 ID，可以创建账户别名。从 IAM 控制面板中，单击 **Create Account Alias** (创建账户别名)，然后输入一个别名，例如您的公司名称。要在创建账户别名后登录，请使用以下 URL：

```
https://your_account_alias.signin.aws.amazon.com/console/
```

要为您的账户验证 IAM 用户的登录链接，请打开 IAM 控制台并在控制面板的 **AWS Account Alias** 下进行检查。



# Amazon Elastic File System 入门

## 主题

- [假设 \(p. 9\)](#)
- [相关主题 \(p. 9\)](#)
- [第 1 步：创建您的 EC2 资源并启动您的 EC2 实例 \(p. 9\)](#)
- [第 2 步：创建您的 Amazon EFS 文件系统 \(p. 12\)](#)
- [第 3 步：连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统 \(p. 13\)](#)
- [步骤 4：使用 EFS 文件同步将文件从现有的文件系统同步到 Amazon EFS \(p. 14\)](#)
- [步骤 5：清理资源并保护您的 AWS 账户 \(p. 15\)](#)

本入门练习将向您演示如何快速创建 Amazon Elastic File System (Amazon EFS) 文件系统、如何将其挂载到您的 VPC 中的 Amazon Elastic Compute Cloud (Amazon EC2) 实例上，以及如何测试端到端设置。

您需要执行四个步骤来创建和使用您的首个 Amazon EFS 文件系统：

- 创建您的 Amazon EC2 资源并启动您的实例。
- 创建您的 Amazon EFS 文件系统。
- 连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统。
- 清理您的资源并保护您的 AWS 账户。

## 假设

在本练习中，我们假设满足以下条件：

- 您已经熟悉如何使用 Amazon EC2 控制台来启动实例。
- 您的 Amazon VPC、Amazon EC2 和 Amazon EFS 资源全部在同一区域中。本指南使用美国西部（俄勒冈）区域（us-west-2）。
- 您在用于本入门练习的区域中有默认 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 中挂载您的文件系统，您仍然可以使用本入门练习。为此，请配置[Amazon EC2 实例和挂载目标的安全组 \(p. 84\)](#)。
- 您没有更改默认安全组的默认入站访问规则。

您可以使用 AWS 账户的根凭证登录到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 AWS 账户的根凭证。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[设置 \(p. 7\)](#)。

## 相关主题

本指南还提供了一个演练，使用 AWS Command Line Interface (AWS CLI) 命令调用 Amazon EFS API，以执行类似的入门练习。有关更多信息，请参阅[演练 1：使用 AWS CLI 创建 Amazon EFS 文件系统并将其挂载到 EC2 实例上 \(p. 110\)](#)。

## 第 1 步：创建您的 EC2 资源并启动您的 EC2 实例

在启动并连接到 Amazon EC2 实例之前，如果您还没有密钥对，则需要创建一个密钥对。您可以使用 Amazon EC2 控制台创建密钥对，然后即可启动您的 EC2 实例。



## Note

不支持将 Amazon EFS 与 Microsoft Windows Amazon EC2 实例结合使用。

## 创建密钥对

- 按照 Amazon EC2 用户指南（适用于 Linux 实例）中的 [使用 Amazon EC2 进行设置](#) 中的步骤创建密钥对。如果您已有密钥对，则不需要创建新密钥对，可使用现有密钥对来完成此练习。

## 启动 EC2 实例

1. 打开 Amazon EC2 控制台 <https://console.aws.amazon.com/ec2/>。
2. 选择 Launch Instance。
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. 在步骤 3：配置实例详细信息中，选择网络，然后选择您的默认 VPC 的条目。它应该类似于 vpc-xxxxxxx (172.31.0.0/16) (default)。
  - a. 选择子网，然后在任何可用区中选择一个子网。
  - b. 选择 Next: Add Storage。
6. 选择 Next: Tag Instance。
7. 命名您的实例，然后选择下一步：配置安全组。
8. 在步骤 6：配置安全组中，检查该页的内容，确保分配安全组设置为创建一个新安全组，并验证创建的入站规则是否具有以下默认值。
  - Type : SSH
  - Protocol : TCP
  - Port Range : 22
  - Source : Anywhere 0.0.0.0/0

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0

Add Rule

**Warning**

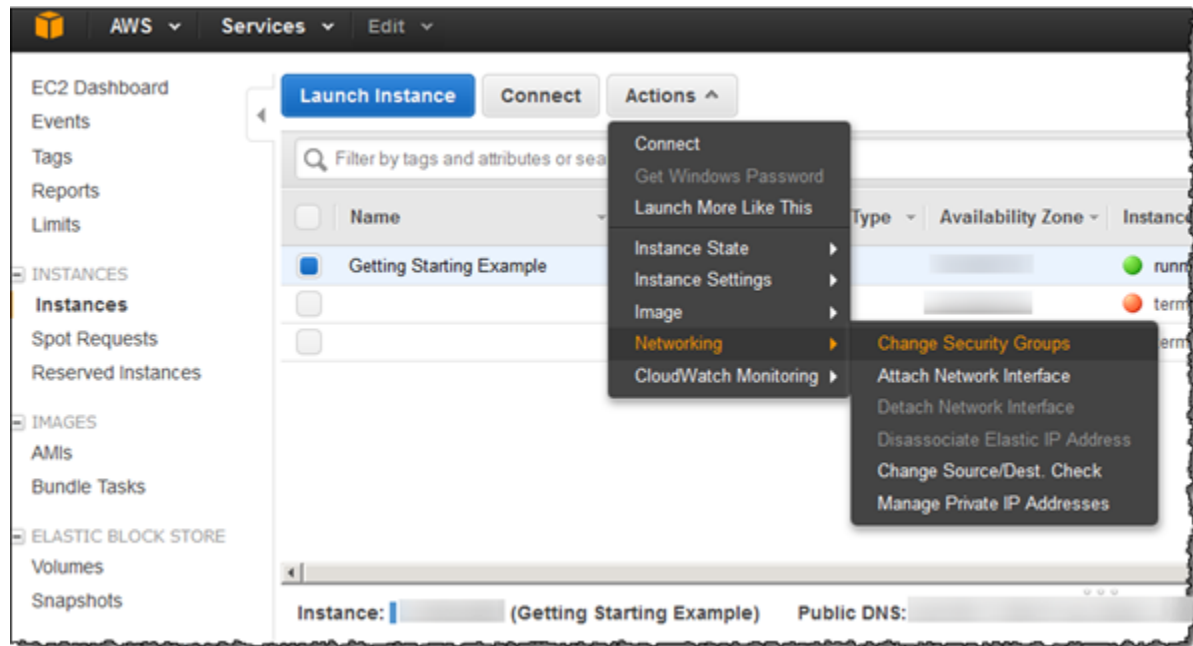
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous **Review and Launch**

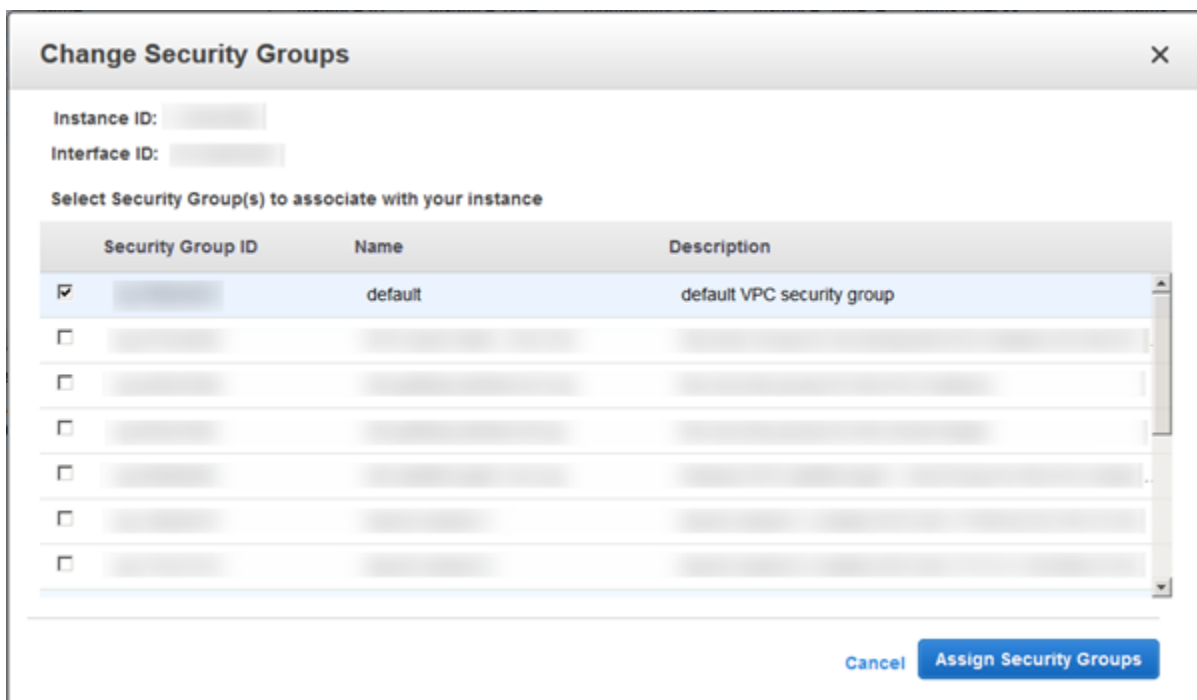
## Note

您可以将 EFS 文件系统配置为自动挂载到您的 EC2 实例。有关更多信息，请参阅 [将 EFS 文件系统配置为在 EC2 实例启动时自动挂载 \(p. 62\)](#)。

9. 选择 Review and Launch。
10. 选择 Launch。
11. 选中您创建的密钥对的复选框，然后选择启动实例。
12. 选择查看实例。
13. 从列表中选择刚创建的实例的名称，然后选择操作。
  - a. 从打开的菜单中选择网络，然后选择更改安全组。



- b. 选中具有默认 VPC 安全组说明的安全组旁边的复选框。
  - c. 选择 Assign Security Groups。



#### Note

在该步骤中，您需要将 VPC 的默认安全组分配给 Amazon EC2 实例。这样做可以确保该实例属于 Amazon EFS 文件系统挂载目标在[第 2 步：创建您的 Amazon EFS 文件系统 \(p. 12\)](#)中为连接授权的安全组。

通过使用 VPC 的默认安全组以及其默认入站和出站规则，该实例以及该文件系统可能会受到 VPC 中的潜在威胁。确保您在本入门练习结束时按照[步骤 5：清理资源并保护您的 AWS 账户 \(p. 15\)](#)中的说明进行操作，以删除该示例分配给您的 VPC 的默认安全组的资源。有关更多信息，请参阅[Amazon EC2 实例和挂载目标的安全组 \(p. 84\)](#)。

14. 从列表中选择您的实例。
15. 在描述选项卡中，确保在安全组旁边列出了两个条目 — 一个条目针对默认 VPC 安全组，另一个条目针对在启动实例时创建的安全组。
16. 记下在 VPC ID 和公有 DNS 旁边列出的值。在本练习后面，您需要使用这些值。

## 第 2 步：创建您的 Amazon EFS 文件系统

在该步骤中，您将创建您的 Amazon EFS 文件系统。

#### 创建 Amazon EFS 文件系统

1. 打开 Amazon EFS 管理控制台 (<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。它具有与您在[第 1 步：创建您的 EC2 资源并启动您的 EC2 实例 \(p. 9\)](#)结束时记录的相同 VPC ID。
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[创建挂载目标 \(p. 20\)](#)。

The screenshot shows the 'Create file system' console page, specifically Step 1: Configure file system access. The page has a sidebar with three steps: Step 1: Configure file system access (selected), Step 2: Add tags, and Step 3: Review and create. The main content area is titled 'Configure file system access' and includes a description: 'An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system via a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.' Below this is a 'VPC' dropdown menu. The 'Create mount targets' section explains that instances connect via mount targets and recommends creating one in each VPC's Availability Zone. It features a table with columns: Availability Zone, Subnet, IP address, and Security group. There are three rows, each with a checkbox, a dropdown for Availability Zone, a dropdown for Subnet (all set to 'default'), 'Automatic' for IP address, and a dropdown for Security group (all set to 'default'). At the bottom right are 'Cancel' and 'Next Step' buttons.

5. 选择 Next Step。
6. 命名您的文件系统，选择 General Purpose (通用型) 和 Bursting (突增) 作为您的默认性能模式，然后选择 Next Step (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，并记下文件系统 ID 值。在下一个步骤中，您需要用到该值。

## 第 3 步：连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统

您可以通过运行 Windows 或 Linux 的计算机连接到您的 Amazon EC2 实例。要连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统，您需要使用 Amazon EFS 文件系统的挂载目标的文件系统 ID 值。您在[第 2 步：创建您的 Amazon EFS 文件系统 \(p. 12\)](#)的结尾记录了该值。

连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统

1. 连接到您的 Amazon EC2 实例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[使用 PuTTY 从 Windows 连接到您的 Linux 实例](#)或[使用 SSH 连接到您的 Linux 实例](#)。
2. 在连接后，请安装 amazon-efs-utils 软件包，它具有 Amazon EFS 挂载帮助程序。

运行以下命令以安装 amazon-efs-utils。

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

### Note

有关 amazon-efs-utils 软件包的更多信息（包括其他 Linux 发行版的安装说明），请参阅[使用 amazon-efs-utils 工具 \(p. 34\)](#)。

3. 使用以下命令为挂载点创建目录。

```
$ sudo mkdir efs
```

4. 将 Amazon EFS 文件系统挂载到您所创建的目录中。请使用以下命令，并将 `file-system-id` 替换为您的文件系统 ID 值。

```
sudo mount -t efs fs-12345678:/ /mnt/efs
```

#### 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。

5. 使用以下命令将目录更改为您创建的新目录。

```
$ cd efs
```

6. 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
```

7. 使用以下命令创建一个文本文件。

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

8. 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 ec2-user ec2-user 0 Aug 15 15:32 test-file.txt
```

## 步骤 4：使用 EFS 文件同步将文件从现有的文件系统同步到 Amazon EFS

现在，您已创建一个正常工作的 Amazon EFS 文件系统，您可以使用 EFS 文件同步将文件从现有的文件系统同步到 Amazon EFS。EFS 文件同步可以同步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。

在该步骤中，我们假定您具有以下内容：

- 您可以从中同步的源 NFS 文件系统。需要能够通过 NFS 版本 3 或版本 4 访问该源系统。源文件系统可能位于本地或 Amazon EC2 上。
- 要同步到的目标 Amazon EFS 文件系统。如果没有 Amazon EFS 文件系统，请创建一个文件系统。有关更多信息，请参阅 [Amazon Elastic File System 入门 \(p. 9\)](#)。
- DNS 访问权限，如下详述：
  - 对于在 Amazon EC2 上托管的同步代理，您的同步代理需要访问在您的 Amazon VPC 中配置的 DNS 服务器。该服务器可能是默认 Amazon DNS 服务器。有关更多信息，请参阅 Amazon VPC 用户指南中的 [在您的 VPC 中使用 DNS](#)。
  - 对于在本地托管的同步代理，您的同步代理需要访问可与 AWS 通信的正常工作的 DNS 服务器。

将文件从现有的文件系统同步到 Amazon EFS

1. 打开 Amazon EFS 管理控制台 (<https://console.aws.amazon.com/efs/>)。
2. 下载并部署一个同步代理。对于本地部署，同步代理是作为 VMware ESXi 的虚拟机 (VM) 映像提供的。对于 AWS 云部署，您可以通过社区 Amazon 系统映像 (AMI) 创建一个 Amazon EC2 实例。
3. 创建一个同步任务，并配置源和目标文件系统。
4. 启动同步任务以开始将文件从源文件系统同步到 Amazon EFS 文件系统。
5. 在 Amazon EFS 控制台上或从 Amazon CloudWatch 中监控同步任务。有关更多信息，请参阅 [使用 Amazon CloudWatch 监控 EFS 文件同步](#) (p. 69)。

有关 EFS 文件同步进程的更多详细信息，请参阅以下内容：

- 有关如何将文件从本地文件系统同步到 Amazon EFS 的信息，请参阅[演练 7：使用 EFS 文件同步将文件从本地文件系统同步到 Amazon EFS](#) (p. 134)。
- 有关如何将文件从 Amazon EC2 同步到 Amazon EFS 的信息，请参阅[演练 8：使用 EFS 文件同步将文件从 Amazon EC2 同步到 Amazon EFS](#) (p. 138)。

## 步骤 5：清理资源并保护您的 AWS 账户

您可以使用本指南中包含的演练来进一步探索 Amazon EFS。在执行此清理步骤之前，可以在这些演练中使用通过本入门练习创建和连接的资源。有关更多信息，请参阅 [演练](#) (p. 110)。完成演练后，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. 连接到您的 Amazon EC2 实例。
2. 使用以下命令卸载 Amazon EFS 文件系统。

```
$ sudo umount efs
```

3. 在 <https://console.aws.amazon.com/efs/> 处打开 Amazon EFS 控制台。
4. 选择要从文件系统列表中删除的 Amazon EFS 文件系统。
5. 对于 Actions，选择 Delete file system。
6. 在永久删除文件系统对话框中，键入要删除的 Amazon EFS 文件系统的文件系统 ID，然后选择删除文件系统。
7. 打开 Amazon EC2 控制台 <https://console.aws.amazon.com/ec2/>。
8. 从实例列表中选择您要终止的 Amazon EC2 实例。
9. 对于操作，请选择实例状态，然后选择终止。
10. 在终止实例中，选择是，请终止以终止您为本入门练习创建的实例。
11. 在导航窗格中，选择 Security Groups。
12. 选择您在 [第 1 步：创建您的 EC2 资源并启动您的 EC2 实例](#) (p. 9) 中 (作为 Amazon EC2 实例启动向导的一部分) 为本入门练习创建的安全组的名称。

### Warning

不要删除您的 VPC 的默认安全组。

13. 对于操作，请选择删除安全组。
14. 在删除安全组中，选择是，删除以删除您为本入门练习创建的安全组。



# 为 Amazon EFS 创建资源

Amazon EFS 提供符合 POSIX 标准的弹性共享文件存储。您创建的文件系统支持来自多个 Amazon EC2 实例的并行读写访问，并且可从创建它的 AWS 区域中的所有可用区访问。

您可以使用网络文件系统 4.0 和 4.1 版协议 (NFSv4) 在 Amazon Virtual Private Cloud (Amazon VPC) 中的 EC2 实例上挂载 Amazon EFS 文件系统。有关更多信息，请参阅 [Amazon EFS：工作原理 \(p. 3\)](#)。

## 主题

- [创建 Amazon Elastic File System \(p. 16\)](#)
- [创建挂载目标 \(p. 20\)](#)
- [创建安全组 \(p. 24\)](#)

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 文件系统需要执行的典型步骤：

- 创建 Amazon EFS 文件系统 – 在创建文件系统时，我们建议您考虑使用名称标签，因为名称标签值显示在控制台中，从而更容易识别文件系统。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 实例和挂载目标都需要有关联的安全组。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您的文件系统的入站流量，方法是向挂载目标安全组添加一条入站规则，以允许来自特定 EC2 实例的访问。然后，您可以将文件系统仅挂载到该 EC2 实例上。

如果您不熟悉 Amazon EFS，我们建议您尝试以下练习，这些练习提供了使用 Amazon EFS 文件系统的第一手端到端体验：

- [入门 \(p. 9\)](#) – 入门练习提供了基于控制台的端到端设置，您可以创建文件系统，将它挂载到 EC2 实例上，然后测试设置。控制台会替您处理很多任务，并帮助您快速设置端到端体验。
- [演练 1：使用 AWS CLI 创建 Amazon EFS 文件系统并将其挂载到 EC2 实例上 \(p. 110\)](#) – 该演练类似于入门练习，只是它使用 AWS Command Line Interface (AWS CLI) 来执行大部分任务。因为 AWS CLI 命令紧密地映射到 Amazon EFS API，所以该演练可以帮助您熟悉 Amazon EFS API 操作。

有关创建和访问文件系统的更多信息，请参阅以下主题。

## 主题

- [创建 Amazon Elastic File System \(p. 16\)](#)
- [创建挂载目标 \(p. 20\)](#)
- [创建安全组 \(p. 24\)](#)

## 创建 Amazon Elastic File System

下面介绍了如何创建 Amazon EFS 文件系统并为其创建可选标签。本节阐述如何使用控制台和 AWS Command Line Interface (AWS CLI) 来创建这些资源。

### Note

如果您不熟悉 Amazon EFS，我们建议您先完成入门练习，该练习提供了如何使用控制台在 VPC 中创建和访问文件系统的端到端说明。有关更多信息，请参阅 [入门 \(p. 9\)](#)。

#### 主题

- [要求 \(p. 17\)](#)
- [所需权限 \(p. 17\)](#)
- [创建文件系统 \(p. 17\)](#)

## 要求

要创建文件系统，唯一的要求是创建一个令牌来确保幂等操作。如果您使用控制台，它会替您生成令牌。有关更多信息，请参阅 [CreateFileSystem \(p. 157\)](#)。创建文件系统后，Amazon EFS 将以 JSON 形式返回文件系统描述。以下是一个示例。

```
{
  "SizeInBytes": {
    "Value": 6144
  },
  "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",
  "CreationTime": 1422823614.0,
  "FileSystemId": "fs-c7a0456e",
  "PerformanceMode": "generalPurpose",
  "NumberOfMountTargets": 0,
  "LifeCycleState": "available",
  "OwnerId": "231243201240"
}
```

如果您使用控制台，控制台会在用户界面中显示此信息。

创建文件系统后，即可为该文件系统创建可选标签。最初，文件系统没有名称。您可以创建一个名称标签以指定文件系统名称。Amazon EFS 提供 [CreateTags \(p. 171\)](#) 操作来创建标签。每个标签都是一个键-值对。

## 所需权限

对于所有操作，例如创建文件和标签，用户都必须具有相应 API 操作和资源的 AWS Identity and Access Management 权限。

您可以使用 AWS 账户的根凭证执行任何 Amazon EFS 操作，但不建议这样做。如果您在您的账户中创建了 IAM 用户，可通过用户策略授予他们对 Amazon EFS 操作的权限。也可以使用角色来授予跨账户权限。有关管理 API 操作权限的更多信息，请参阅 [Amazon EFS 的身份验证和访问控制 \(p. 144\)](#)。

## 创建文件系统

可以使用 Amazon EFS 控制台或 AWS Command Line Interface 创建文件系统。也可以使用 AWS 软件开发工具包以编程方式创建文件系统。

### 使用 Amazon EFS 控制台创建文件系统

Amazon EFS 控制台提供了集成的体验。在创建文件系统时，您可以在控制台中指定 VPC 子网以创建挂载目标和可选的文件系统标签。

要在 VPC 中创建文件系统挂载目标，您必须指定 VPC 子网。控制台会预填充您的账户中位于选定 AWS 区域的 VPC 列表。首先，您要选择 VPC，然后控制台会列出 VPC 中的可用区。对于每个可用区，您都可以从列表中选择子网。选择子网后，您可以指定子网中的可用 IP 地址，也可以让 Amazon EFS 选择一个地址。

在创建文件系统时，还要选择性能模式。有两种性能模式可供选择，分别是通用模式和最大 I/O 模式。对于绝大多数使用案例，我们建议对文件系统使用通用性能模式。有关不同性能模式的更多信息，请参阅 [性能模式 \(p. 79\)](#)。



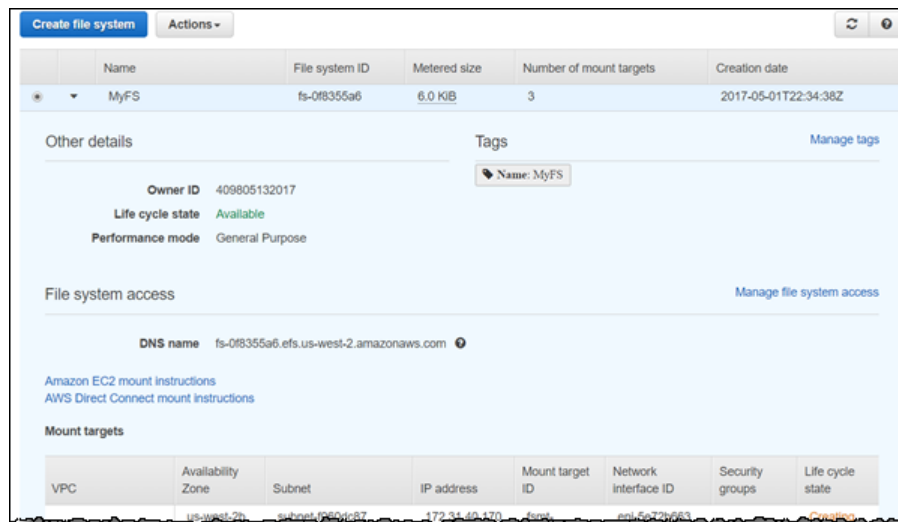
除性能模式外，您还可以选择吞吐量模式。有两种吞吐量模式可供选择：突增和预配置。默认的突增吞吐量模式易于使用，适用于大多数应用程序和各种性能要求。预配置模式适用于需要比突增吞吐量模式允许的吞吐量与存储容量之比更大比率的应用程序。有关更多信息，请参阅 [通过预配置模式指定吞吐量 \(p. 82\)](#)。

#### Note

使用预配置吞吐量模式会产生相关的额外费用。有关更多信息，请参阅 <https://aws.amazon.com/efs/pricing>。

您可以在创建文件系统时启用静态加密。如果您为文件系统启用静态加密，则会加密其中存储的所有数据和元数据。您可以在以后挂载文件系统时启用传输中加密。有关 Amazon EFS 加密的更多信息，请参阅 [安全性 \(p. 84\)](#)。

在选择创建文件系统时，控制台将发送一系列 API 请求以创建文件系统。然后，控制台发送 API 请求，以便为文件系统创建标签和挂载目标。在以下示例中，控制台显示 MyFS 文件系统。为该文件系统创建了名称标签和三个挂载目标。挂载目标生命周期状态必须为可用，然后才能使用挂载目标在 EC2 实例上挂载文件系统。



有关如何使用控制台创建 Amazon EFS 文件系统的说明，请参阅 [第 1 步：创建您的 EC2 资源并启动您的 EC2 实例 \(p. 9\)](#)。

## 使用 AWS CLI 创建文件系统

在使用 AWS CLI 时，您将按顺序创建这些资源。首先，创建一个文件系统。然后，可以使用相应的 AWS CLI 命令为该文件系统创建挂载目标和可选标签。

以下示例使用 `adminuser` 作为 `profile` 参数值。您需要使用适当的用户配置文件来提供您的凭证。有关 AWS CLI 的信息，请参阅 [AWS Command Line Interface 用户指南](#) 中的 [使用 AWS 命令行界面进行设置](#)。

- 要创建文件系统，请使用 Amazon EFS `create-file-system` CLI 命令 (相应操作为 [CreateFileSystem \(p. 157\)](#))，如下所示。

```
$ aws efs create-file-system \
--creation-token creation-token \
--performance-mode generalPurpose \
--throughput-mode bursting \
--region aws-region \
--profile adminuser
```

例如，以下 `create-file-system` 命令在 **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。有关您可以在其中创建 Amazon EFS 文件系统的 AWS 区域的列表，请参阅 [Amazon Web Services 一般参考](#)。

```
$ aws efs create-file-system \
--creation-token MyFirstFS \
--performance-mode generalPurpose \
--throughput-mode bursting \
--region us-west-2 \
--profile adminuser
```

在成功创建文件系统后，Amazon EFS 以 JSON 形式返回文件系统描述，如下示例所示。

```
{
  "SizeInBytes": {
    "Value": 6144
  },
  "CreationToken": "MyFirstFS",
  "CreationTime": 1422823614.0,
  "FileSystemId": "fs-c7a0456e",
  "PerformanceMode": "generalPurpose",
  "ThroughputMode": "bursting",
  "NumberOfMountTargets": 0,
  "LifecycleState": "available",
  "OwnerId": "231243201240"
}
```

Amazon EFS 还提供 `describe-file-systems` CLI 命令 (相应操作为 [DescribeFileSystems \(p. 181\)](#))，可以用来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 返回您的 AWS 账户中在指定区域创建的文件系统的列表。

- 要创建标签，请使用 Amazon EFS `create-tags` CLI 命令 (相应的 API 操作为 [CreateTags \(p. 171\)](#))。以下示例命令向文件系统添加 Name 标签。

```
aws efs create-tags \
--file-system-id File-System-ID \
--tags Key=Name,Value=SomeExampleNameValue \
--region aws-region \
--profile adminuser
```

您可以使用 `describe-tags` CLI 命令 (相应操作为 [DescribeTags \(p. 191\)](#)) 检索为文件系统创建的标签列表，如下所示。

```
aws efs describe-tags \
--file-system-id File-System-ID \
--region aws-region \
--profile adminuser
```

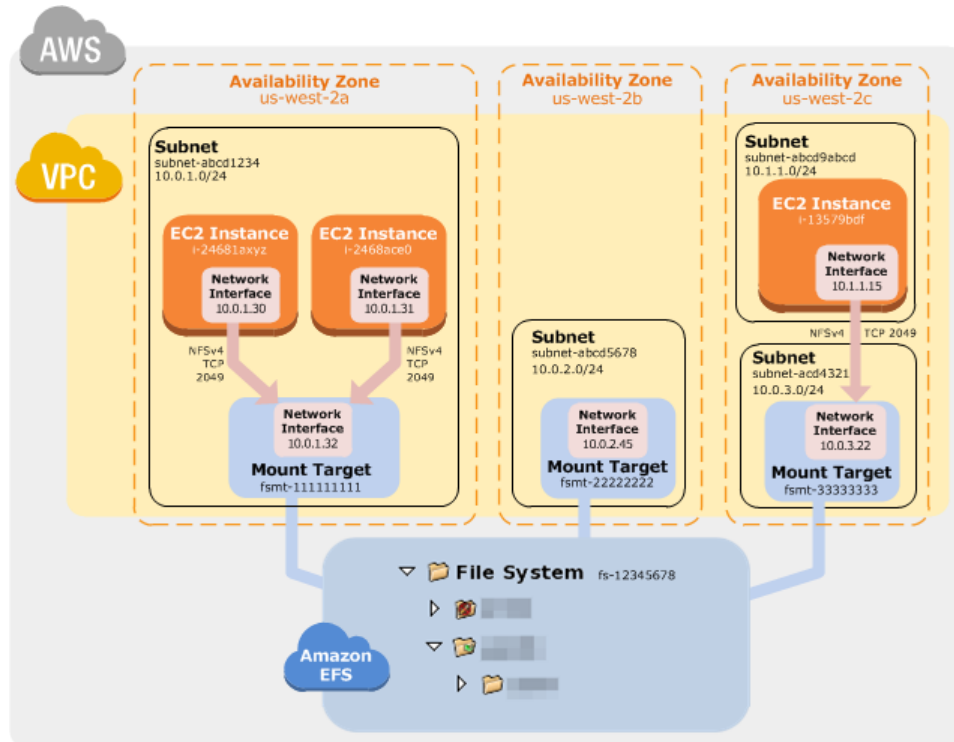
Amazon EFS 以 JSON 形式返回这些描述。下面是 `DescribeTags` 操作返回的标签示例。它显示文件系统仅有 Name 标签。

```
{
```

```
"Tags": [
  {
    "Key": "Name",
    "Value": "MyFS"
  }
]
```

## 创建挂载目标

创建文件系统后，即可创建挂载目标，然后再将文件系统挂载到 VPC 中的 EC2 实例上，如下图所示。



有关创建文件系统的更多信息，请参阅[创建 Amazon Elastic File System \(p. 16\)](#)。

挂载目标安全组充当控制流量的虚拟防火墙。例如，它判断哪些 Amazon EC2 实例可以访问文件系统。本节介绍以下内容：

- 挂载目标安全组如何启用流量。
- 如何将文件系统挂载到 Amazon EC2 实例上。
- NFS 级权限注意事项。

最初，只有 Amazon EC2 实例上的根用户才具有对文件系统的读写执行权限。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 \(p. 26\)](#)。

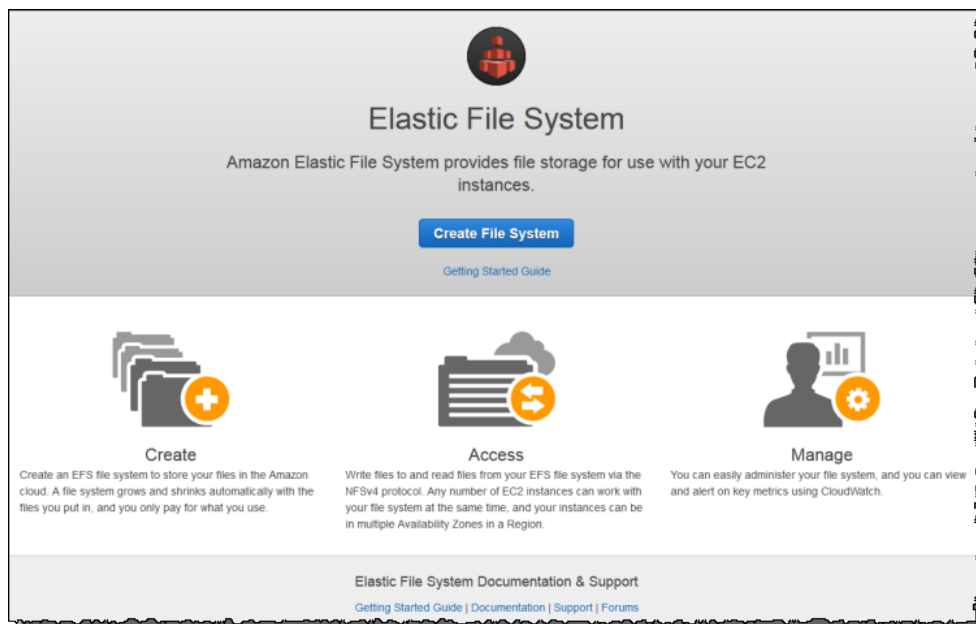
您可以使用控制台、AWS Command Line Interface 或使用 AWS 软件开发工具包以编程方式为文件系统创建挂载目标。使用控制台时，您可以在首次创建文件系统时或创建文件系统之后创建挂载目标。

## 使用 Amazon EFS 控制台创建挂载目标

执行以下过程中的步骤通过使用控制台创建挂载目标。当您按照控制台步骤执行时，也可以创建一个或多个挂载目标。可以为 VPC 中的每个可用区分别创建一个挂载目标。

### 创建 Amazon EFS 文件系统 (控制台)

1. 登录 AWS 管理控制台并通过以下网址打开 Amazon EFS 控制台：<https://console.aws.amazon.com/efs/>。
2. 选择创建文件系统。



### Note

仅当您还没有任何 Amazon EFS 文件系统时，控制台才显示上述页面。如果您已创建了文件系统，则控制台会显示文件系统列表。在列表页面上，选择创建文件系统。

3. 在步骤 1：配置文件系统访问页上，选择您希望控制台在其中为创建的文件系统创建一个或多个挂载目标的 VPC 和 VPC 可用区。此 VPC 应和您在上一节创建 Amazon EC2 实例的 Amazon VPC 相同。
  - a. 从 VPC 列表选择一个 Amazon VPC。

### Warning

如果您想要的 Amazon VPC 没有列出，请在 Amazon EFS 控制台的全局导航中验证该区域。

- b. 在创建挂载目标部分中，选择列出的所有可用区。

我们建议您在所有可用区中创建挂载目标。然后，您可以将文件系统挂载到在任何 Amazon VPC 子网中创建的 Amazon EC2 实例上。

### Note

您可以使用在一个可用区中创建的挂载目标访问另一个可用区中的 Amazon EC2 实例上的文件系统，但跨可用区访问会产生相应的成本。

对于每个可用区，请执行以下操作：

- 从列表选择一个要在其中创建挂载目标的子网。

您可以在每个可用区中创建一个挂载目标。如果在您启动了 Amazon EC2 实例的可用区中有多个子网，则不必非得在同一子网中创建挂载目标，而是可以在该可用区内的任何一个子网中创建。

- 将 IP 地址保留为自动。Amazon EFS 将为挂载目标选择一个可用 IP 地址。
- 指定专门为挂载目标创建的安全组，或指定默认 VPC 的默认安全组。两种安全组都将具有必要的入站规则以允许来自 EC2 实例安全组的入站访问。

单击安全组框，控制台将显示可用的安全组。您可以在其中选择特定的安全组并删除默认安全组，或者保留默认值，具体取决于您配置 Amazon EC2 实例的方式。

**Create File System**

Step 1: Configure File System Access  
Step 2: Add Tags  
Step 3: Review and Create

**Configure File System Access**

An EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system via a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC: vpc-460fba23 ⓘ

**Create Mount Targets**

Instances connect to a file system via mount targets you create. We recommend creating a mount target in each of your VPC's availability zones so that EC2 instances across your VPC can access the file system.

Availability Zone	Subnet ⓘ	IP Address ⓘ	Security Group ⓘ
<input checked="" type="checkbox"/> eu-west-1a	subnet-f5c2a39d	Automatic ⚙	× sg-2291ce47 - efs-getting-started-ml-sg
<input checked="" type="checkbox"/> eu-west-1b	subnet-8c41c8fb	Automatic ⚙	× sg-2291ce47 - efs-getting-started-ml-sg
<input checked="" type="checkbox"/> eu-west-1c	subnet-0dc36154	Automatic ⚙	× sg-2291ce47 - efs-getting-started-ml-sg

Cancel Next Step

4. 在步骤 2：配置可选设置页上，指定名称标签的值 (**MyExampleFileSystem**) 并选择性能模式。

控制台将预填充名称标签，因为 Amazon EFS 将其值作为文件系统显示名称。

The screenshot shows the 'Create file system' console page, specifically Step 2: Configure optional settings. The left sidebar shows the progress: Step 1: Configure file system access, Step 2: Configure optional settings (active), and Step 3: Review and create. The main content area is titled 'Configure optional settings' and includes an 'Add tags' section with a text box for a key-value pair (Key: Name, Value: MyExampleFileSystem) and a 'Choose performance mode' section with radio buttons for 'General Purpose (default)' and 'Max I/O'. At the bottom right are 'Cancel', 'Previous', and 'Next Step' buttons.

### Create file system

Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

#### Configure optional settings

##### Add tags

You can add tags to describe your file system. A tag consists of a case-sensitive key-value pair. (For example, you can define a tag with key-value pair with key = Corporate Department and value = Sales and Marketing.) At a minimum, we recommend a tag with key = Name.

Key	Value	Remove
Name	MyExampleFileSystem	
<input type="text" value="Add New Key"/>		

##### Choose performance mode

We recommend **General Purpose** performance mode for most file systems. **Max I/O** performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system — it scales to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

☒ General Purpose (default)

☐ Max I/O

Cancel Previous Next Step

5. 在步骤 3：审核和创建页上，选择创建文件系统。

The screenshot shows the 'Create file system' console page, specifically Step 3: Review and create. The left sidebar shows the progress: Step 1: Configure file system access, Step 2: Configure optional settings, and Step 3: Review and create (active). The main content area is titled 'Review and create' and includes a 'File system access' table with columns for VPC, Availability Zone, Subnet, IP address, and Security groups. Below the table is an 'Optional settings' section with a 'Tags' field (Name: MyExampleFileSystem) and a 'Performance mode' dropdown (General Purpose (default)). At the bottom right are 'Cancel', 'Previous', and 'Create File System' buttons.

### Create file system

Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

#### Review and create

Review the configuration below before proceeding to create your file system.

##### File system access

VPC	Availability Zone	Subnet	IP address	Security groups
vpc- (default)	us-west-2a		Automatic	
	us-west-2b		Automatic	
	us-west-2c		Automatic	

##### Optional settings

Tags

Performance mode

Cancel Previous Create File System

6. 控制台将在文件系统页上显示新创建的文件系统。确认所有挂载目标将生命周期状态显示为可用。可能需要过一会儿挂载目标才变得可用 (您可以在 EFS 控制台中展开/折叠文件系统以强制它刷新)。

- 在文件系统访问下面，将会看到文件系统的 DNS 名称。记下这个 DNS 名称。在下一节中，您将使用此 DNS 名称通过挂载目标将文件系统挂载到 Amazon EC2 实例上。挂载了文件系统的 Amazon 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。

现在，您已准备就绪，可以在 Amazon EC2 实例上挂载 Amazon EFS 文件系统了。

## 使用 AWS CLI 创建挂载目标

要使用 AWS CLI 创建挂载目标，请使用 `create-mount-target` CLI 命令 (相应操作为 [CreateMountTarget](#) (p. 164))，如下所示。

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

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。

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

您也可以使用 `describe-mount-targets` CLI 命令 (相应操作为 [DescribeMountTargets](#) (p. 185)) 检索为文件系统创建的挂载目标列表，如下所示。

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

有关示例，请参阅[演练 1：使用 AWS CLI 创建 Amazon EFS 文件系统并将其挂载到 EC2 实例上](#) (p. 110)。

## 创建安全组

### Note

下一节针对的是 Amazon EC2，并介绍了如何创建安全组，以便使用安全 Shell (SSH) 连接到任何挂载了 Amazon EFS 文件系统的实例。如果不使用 SSH 连接到您的 Amazon EC2 实例，则可以跳过该节。

Amazon EC2 实例和挂载目标都有关联的安全组。这些安全组充当虚拟防火墙，控制它们之间的流量。如果您在创建挂载目标时未提供安全组，Amazon EFS 则将 VPC 的默认安全组与之关联。

无论如何，要启用 EC2 实例和挂载目标 (以后随后的文件系统) 之间的流量，您必须在这些安全组中配置以下规则：

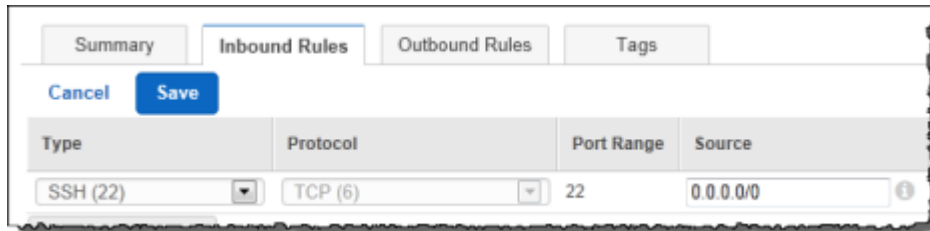
- 与挂载目标关联的安全组必须允许来自您要挂载文件系统的所有 EC2 实例在 NFS 端口上对 TCP 协议的入站访问。
- 每个挂载文件系统的 EC2 实例都必须有一个安全组，以便允许在 NFS 端口上对挂载目标的出站访问。

有关安全组的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 [Amazon EC2 安全组](#)。

## 使用 AWS 管理控制台创建安全组

您可以使用 AWS 管理控制台在 VPC 中创建安全组。要连接 Amazon EFS 文件系统与 Amazon EC2 实例，需要创建两个安全组：一个用于 Amazon EC2 实例，另一个用于 Amazon EFS 挂载目标。

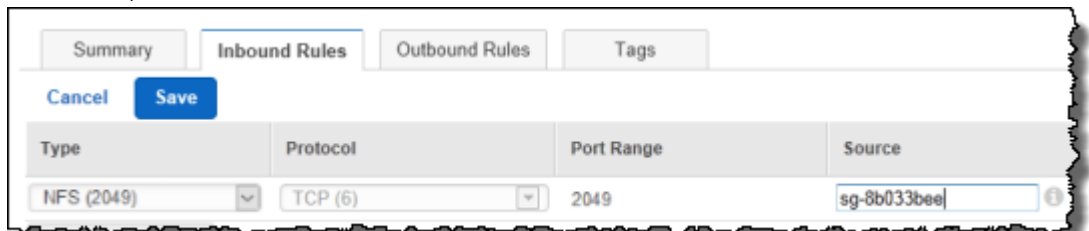
1. 在 VPC 中创建两个安全组。有关说明，请参阅 Amazon VPC 用户指南 中的 [创建安全组](#)。
2. 在 VPC 控制台中，验证这些安全组的默认规则。两个安全组都应当只有一条允许出站流量的出站规则。
3. 您需要向安全组授予额外访问权限，如下所示：
  - a. 在 EC2 安全组中添加一个规则以允许入站访问，如下所示。或者，您可以限制源地址。



Type	Protocol	Port Range	Source
SSH (22)	TCP (6)	22	0.0.0.0/0

有关说明，请参阅 Amazon VPC 用户指南 中的 [添加和删除规则](#)。

- b. 在挂载目标安全组中添加一个规则以允许来自 EC2 安全组的入站访问，如下所示（其中 EC2 安全组被标识为源）：



Type	Protocol	Port Range	Source
NFS (2049)	TCP (6)	2049	sg-8b033bee

### Note

您无需添加出站规则，因为默认出站规则允许所有流量离开（否则，您将需要添加出站规则以打开 NFS 端口上的 TCP 连接，从而将挂载目标安全组标识为目标）。

4. 确认两个安全组现在都如本节中所述授权了入站和出站访问。

## 使用 AWS CLI 创建安全组

有关如何使用 AWS CLI 创建安全组的示例，请参阅 [步骤 1：创建 Amazon EC2 资源](#) (p. 112)。



# 使用文件系统

Amazon Elastic File System 提供了一个标准文件系统接口以支持完整文件系统访问语义。通过使用 NFSv4.1，您可以在任何基于 Linux 的 Amazon Elastic Compute Cloud (Amazon EC2) 实例上挂载您的 Amazon EFS 文件系统。在挂载后，您可以像本地文件系统一样使用文件和目录。有关挂载的更多信息，请参阅[挂载文件系统 \(p. 59\)](#)。

您还可以使用 EFS 文件同步将文件从任何文件系统复制到 Amazon EFS。有关挂载的更多信息，请参阅[Amazon EFS 文件同步 \(p. 28\)](#)。

创建文件系统并将其挂载到 EC2 实例上后，为了有效地使用它们，您需要知道以下几点：

- 用户、组和相关 NFS 级别权限管理 – 当您首次创建文件系统时，只有 / 处的一个根目录。默认情况下，只有根用户 (UID 0) 具有读写执行权限。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。有关更多信息，请参阅[网络文件系统 \(NFS\) 级别用户、组和权限 \(p. 26\)](#)。

## 相关主题

[Amazon EFS：工作原理 \(p. 3\)](#)

[入门 \(p. 9\)](#)

[演练 \(p. 110\)](#)

## 网络文件系统 (NFS) 级别用户、组和权限

### 主题

- [示例 Amazon EFS 文件系统使用案例和权限 \(p. 27\)](#)
- [对文件系统上的文件和目录的用户和组 ID 权限 \(p. 27\)](#)
- [无根 Squash \(p. 28\)](#)
- [权限缓存 \(p. 28\)](#)
- [更改文件系统对象所有权 \(p. 28\)](#)

创建文件系统后，默认情况下，只有根用户 (UID 0) 具有读写执行权限。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。

Amazon EFS 文件系统对象具有关联的 Unix 风格模式。这个值定义了对该对象执行操作的权限，熟悉 Unix 风格系统的用户很容易理解 Amazon EFS 对这些权限所表现出来的行为。

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。Amazon EFS 上的文件系统对象 (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，Amazon EFS 使用这些数字 ID 来检查权限。

本节提供权限示例，并讨论特定于 Amazon EFS 的 NFS 权限注意事项。

## 示例 Amazon EFS 文件系统使用案例和权限

创建 Amazon 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。

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,wsize=1048576,hard,timeo=600,retrains=2,noresvport file-system-
id.efs.aws-region.amazonaws.com:/ efs-mount-point
```

请注意，根用户和根组拥有挂载的目录。

```
[ec2-user@ip-172-31-43-70 efs]$ ls -al
total 8
drwxr-xr-x 2 root    root    6144 Aug 29  2016 .
drwx----- 5 ec2-user ec2-user 4096 May  1 21:44 ..
[ec2-user@ip-172-31-43-70 efs]$
```

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

请注意，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。如需分步指导，请参阅 [演练 3：创建可写的每用户子目录以及配置在重启时自动重新挂载 \(p. 126\)](#)。
- 允许用户写入 Amazon 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 Amazon 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
```

该命令为所有挂载了文件系统的 EC2 实例上的所有用户授予读写执行权限。

## 对文件系统上的文件和目录的用户和组 ID 权限

Amazon EFS 文件系统上的文件和目录支持 Unix 风格的标准读/写/执行权限，这些权限基于通过挂载 NFSv4.1 客户端所声明的用户 ID 和组 ID。当用户尝试访问文件和目录时，Amazon EFS 会检查其用户 ID 和组 ID，以验证用户是否有权访问对象。Amazon EFS 还使用这些 ID 作为用户创建的新文件和目录的所有者和根所有者。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 UID 和 GID。数字用户 ID 在 Linux 系统上的 /etc/passwd 文件中设置。数字组 ID 在 /etc/group 文件中。这些文件定义名称与 ID 之间的映射。除 EC2 实例外，Amazon EFS 不对这些 ID 执行任何验证，包括根 ID 0。

如果用户从两个不同的 EC2 实例访问 Amazon 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，并且从两个不同的 EC2 实例访问 Amazon EFS 文件系统的用户体验也不一样。
- 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，如下所示：

```
$ id
uid=502(joe) gid=502(joe) groups=502(joe)
```

## 关闭 ID 映射器

操作系统中的 NFS 实用软件包括一个名为 ID 映射器的守护程序，用于管理用户名与 ID 之间的映射。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 上映射器通常被禁用，在这种情况下不要启用 ID 映射器），如下所示：

```
$ service rpcidmapd status
$ sudo service rpcidmapd stop
```

## 无根 Squash

当根 squash 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。

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）。

## 权限缓存

Amazon EFS 会将文件权限缓存一小段时间。因此，可能存在一个短暂的窗口，允许之前能够访问文件系统对象但最近被撤销访问权限的用户访问该对象。

## 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。尽管根或所有者用户可以更改文件系统对象的所有者组，但除非用户是根用户，否则该组只能更改为所有者用户隶属的组。

# Amazon EFS 文件同步

Amazon EFS 文件同步将文件从现有的本地或云文件系统复制到 Amazon EFS 文件系统。EFS 文件同步通过 Internet 或 AWS Direct Connect 连接安全高效地复制文件。它复制文件数据和文件系统元数据，例如，所有权、时间戳和访问权限。有关如何使用 AWS Direct Connect 的信息，请参阅[演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统](#) (p. 127)。

#### Note

您不需要将 AWS Direct Connect 设置为使用 EFS 文件同步。

#### 主题

- [EFS 文件同步要求 \(p. 29\)](#)
- [EFS 文件同步架构 \(p. 31\)](#)
- [EFS 文件同步如何传输文件 \(p. 32\)](#)

## EFS 文件同步要求

除非另有说明，否则，需要使用以下内容以创建 Amazon EFS 文件同步。

### 硬件要求

在本地部署 Amazon EFS 文件同步时，您必须确保部署文件同步虚拟机的基础硬件可以专门使用以下最少资源：

- 分配给 VM 的四个虚拟处理器。
- 分配给 VM 的 32 GB RAM
- 80GB 磁盘空间，适用于安装虚拟机映像和系统数据。

在 Amazon EC2 上部署 Amazon EFS 文件同步时，实例大小必须至少为 xlarge，Amazon EFS 文件同步才能正常工作。我们建议您使用内存优化的 r4.xlarge 实例类型之一。

### 支持的管理程序和主机要求

您可以选择在本地将 EFS 文件同步作为虚拟机 (VM) 运行，或者在 AWS 中将其作为 Amazon Elastic Compute Cloud (Amazon EC2) 实例运行。

EFS 文件同步支持以下管理程序版本和主机：

- VMware ESXi 管理程序 (4.1、5.0、5.1、5.5、6.0 或 6.5 版) - 可以从 [VMware 网站](#) 中获取免费的 VMware ESXi 管理程序版本。您还需要使用 VMware vSphere 客户端以连接到主机。
- EC2 实例 – EFS 文件同步提供了一个包含 EFS 文件同步虚拟机映像的 Amazon 系统映像 (AMI)。我们建议您使用内存优化的 r4.xlarge 实例类型。

### 支持的 NFS 协议

EFS 文件同步支持 NFS v3.x、NFS v4.0 和 NFS v4.1。

### 允许 EFS 文件同步通过防火墙和路由器进行访问

EFS 文件同步需要访问以下终端节点，以便与 AWS 进行通信。如果使用防火墙或路由器来筛选或限制网络流量，则必须配置防火墙和路由器以允许这些服务终端节点与 AWS 进行出站通信。

EFS 文件同步需要使用以下终端节点。

```
cp-sync.$region.amazonaws.com
activation-sync.$region.amazonaws.com
ec2-*.amazonaws.com
repo.$region.amazonaws.com
repo.default.amazonaws.com
```

```
packages.$region.amazonaws.com
0.amazon.pool.ntp.org
1.amazon.pool.ntp.org
2.amazon.pool.ntp.org
3.amazon.pool.ntp.org
54.201.223.107
169.254.169.123
```

有关支持的 AWS 区域的信息，请参阅《AWS 一般参考》中的 [Amazon Elastic File System](#)。

在激活同步代理之前，需要使用 Amazon CloudFront 终端节点以获取可用的 AWS 区域列表。

```
https://d4kdq0yaxexbo.cloudfront.net/
```

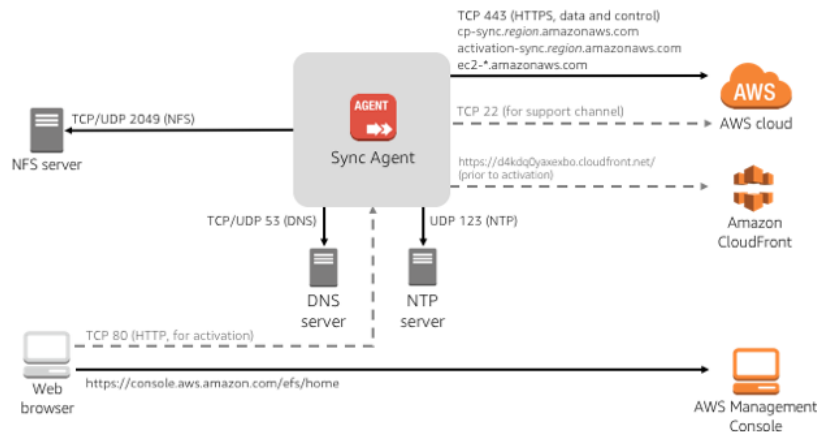
## 网络和端口要求

EFS 文件同步需要使用以下端口才能正常运行。

从	目的	协议	端口	如何使用	
EFS 文件同步虚拟机	AWS	TCP	443 (HTTPS)	用于从 EFS 文件同步虚拟机到 AWS 服务终端节点的通信。有关服务终端节点的信息，请参阅 <a href="#">允许 EFS 文件同步通过防火墙和路由器进行访问 (p. 29)</a> 。	
您的 Web 浏览器	EFS 文件同步虚拟机	TCP	80 (HTTP)	由本地系统使用以获取同步代理激活密钥。仅在激活 EFS 文件同步代理期间使用端口 80。  EFS 文件同步虚拟机不要求可公开访问端口 80。所需的端口 80 访问级别取决于网络配置。如果从 Amazon EFS 管理控制台中激活同步代理，从中连接到控制台的主机必须有权访问端口 80。	
EFS 文件同步虚拟机	域名服务 (DNS) 服务器	TCP/UDP	53 (DNS)	用于 EFS 文件同步虚拟机和 DNS 服务器之间的通信。	

从	目的	协议	端口	如何使用	
EFS 文件同步虚拟机	AWS	TCP	22 (支持渠道)	允许 AWS Support 访问您的 EFS 文件同步以帮助解决 EFS 文件同步问题。您不需要打开该端口即可正常运行，但需要使用该端口以进行故障排除。	
EFS 文件同步虚拟机	NTP 服务器	UDP	123 (NTP)	由本地系统使用以将虚拟机时间同步到主机时间。	
EFS 文件同步虚拟机	NFS 服务器	TCP/UDP	2049 (NFS)	由 EFS 文件同步虚拟机使用以挂载源 NFS 文件系统。  支持 NFS v3.x、NFS v4.0 和 NFS v4.1。	

以下是所需的端口图表，并列出了 EFS 文件同步所需的端口。

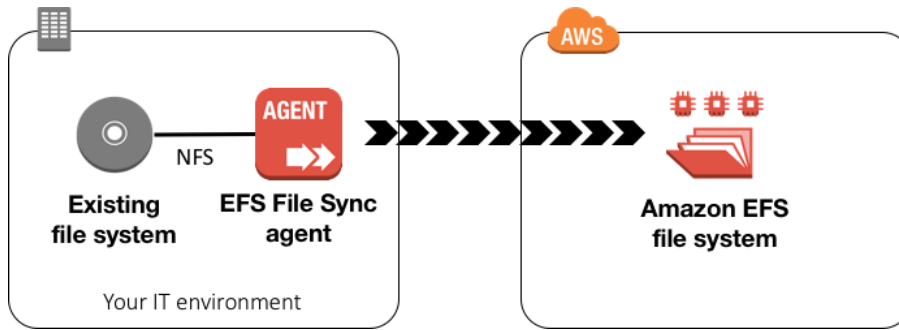


## EFS 文件同步架构

EFS 文件同步具有以下优势：

- 高效、高性能的并行数据传输，可以容忍不可靠和高延迟的网络。
- 加密从您的 IT 环境传输到 AWS 的数据。
- 数据传输速率比标准 Linux 复制工具最多快 5 倍。
- 为重复传输提供完整和增量同步。

下图是 EFS 文件同步架构的简要视图。



将文件从现有的文件系统同步到 Amazon EFS

1. 打开 Amazon EFS 管理控制台 (<https://console.aws.amazon.com/efs/>)。
2. 下载并部署一个同步代理。对于本地部署，同步代理是作为 VMware ESXi 的虚拟机 (VM) 映像提供的。对于 AWS 云部署，您可以通过社区 Amazon 系统映像 (AMI) 创建一个 Amazon EC2 实例。
3. 创建一个同步任务，并配置源和目标文件系统。
4. 启动同步任务以开始将文件从源文件系统同步到 Amazon EFS 文件系统。
5. 在 Amazon EFS 控制台上或从 Amazon CloudWatch 中监控同步任务。有关更多信息，请参阅 [使用 Amazon CloudWatch 监控 EFS 文件同步](#) (p. 69)。

有关 EFS 文件同步进程的更多详细信息，请参阅以下内容：

- 有关如何将文件从本地文件系统同步到 Amazon EFS 的信息，请参阅[演练 7：使用 EFS 文件同步将文件从本地文件系统同步到 Amazon EFS](#) (p. 134)。
- 有关如何将文件从 Amazon EC2 同步到 Amazon EFS 的信息，请参阅[演练 8：使用 EFS 文件同步将文件系统从 Amazon EC2 同步到 Amazon EFS](#) (p. 138)。

## EFS 文件同步如何传输文件

在启动同步任务时，它将经历三种不同的状态：正在准备、正在同步和正在验证。在正在准备状态下，EFS 文件同步检查源和目标文件系统以确定要同步的文件。为此，它递归扫描源和目标文件系统内容以查找差异。它检查的文件包括已修改、删除和添加的文件以及修改了元数据的文件。

在完成扫描并计算差异后，EFS 文件同步将过渡到正在同步状态。此时，EFS 文件同步开始将文件从源文件系统传输到目标 Amazon EFS 文件系统。仅传输已添加、修改或删除的文件。这种增量传输不取决于您使用的同步任务，而是取决于源和目标文件系统内容。在“配置同步设置”对话框中，您可以在源文件系统中选择要保留的元数据。您还可以配置同步任务设置以在目标中保留或删除文件，即使在源文件系统中找不到这些文件。

在同步完成后，EFS 文件同步验证源和目标文件系统之间的一致性。这是正在验证状态。默认情况下，EFS 文件同步在传输文件时使用源和目标完整一致性验证。在同步任务的正在验证阶段，EFS 文件同步重新扫描源和目标内容以查找任何差异。如果找不到任何差异，则该任务成功。否则，将该任务标记为验证失败。有关 EFS 文件同步状态的信息，请参阅[了解同步任务状态](#) (p. 48)。

## 从本地存储阵列传输数据的最佳实践

您可能希望将数据从本地企业存储阵列传输到 Amazon EFS。在这种情况下，在将文件从网络文件系统 (NFS) 传输到 Amazon EFS 时，其他应用程序可能会修改源文件系统中的文件。

要确保 EFS 文件同步成功执行传输并进行完整一致性验证，我们建议源位置指向一个只读快照。该设置确保在传输文件时无法修改源位置中的文件，并确保验证正常工作。

有关如何在企业存储阵列中拍摄快照的信息，请参阅以下内容之一：

- EMC VNX：[如何创建 VNX 快照并将其附加到服务器](#)
- EMC VMAX：[EMC TimeFinder 产品说明指南](#)
- NetApp：[快照管理](#)
- HPE 3PAR：[快照和复制数据管理](#)
- HDS：[Hitachi 写入时复制快照用户指南](#)

## 相关主题

[步骤 2：创建同步任务 \(p. 135\)](#)

[了解同步任务状态 \(p. 48\)](#)



# 使用 amazon-efs-utils 工具

您可以在下文中找到 amazon-efs-utils 说明，这是一个开源 Amazon EFS 工具集。

## 主题

- [概述 \(p. 34\)](#)
- [在 Amazon Linux 上安装 amazon-efs-utils 软件包 \(p. 35\)](#)
- [在其他 Linux 发行版上安装 amazon-efs-utils 软件包 \(p. 35\)](#)
- [升级 stunnel \(p. 36\)](#)
- [EFS 挂载帮助程序 \(p. 37\)](#)

## 概述

amazon-efs-utils 软件包是一个开源 Amazon EFS 工具集。使用 amazon-efs-utils 不会产生额外费用，您可以从 GitHub 中下载这些工具：<https://github.com/aws/efs-utils>。amazon-efs-utils 软件包是在 Amazon Linux 软件包存储库中提供的，您可以在其他 Linux 发行版上构建和安装该软件包。

amazon-efs-utils 软件包附带提供了挂载帮助程序和一些工具，从而为 Amazon EFS 轻松加密传输中的数据。挂载帮助程序是一个在挂载特定类型的文件系统时使用的程序。我们建议您使用 amazon-efs-utils 中包含的挂载帮助程序挂载您的 Amazon EFS 文件系统。

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 ( nfs-utils 软件包 )
- 网络中继 ( stunnel 软件包 4.56 或更高版本 )
- Python ( 2.7 或更高版本 )
- OpenSSL 1.0.2 或更高版本

## Note

默认情况下，在将 Amazon EFS 挂载帮助程序与传输层安全性 (TLS) 一起使用时，挂载帮助程序强制使用在线证书状态协议 (OCSP) 和证书主机名检查。Amazon 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 Amazon EFS 文件系统将失败。

如果已安装 amazon-efs-utils 软件包，要升级您的系统的 stunnel 版本，请参阅[升级 stunnel \(p. 36\)](#)。

有关加密问题，请参阅[排除加密故障 \(p. 102\)](#)。

以下 Linux 发行版支持 amazon-efs-utils：

- Amazon Linux 2
- Amazon Linux
- 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

在以下几节中，您可以了解如何在 Linux 实例上安装 amazon-efs-utils。

## 在 Amazon Linux 上安装 amazon-efs-utils 软件包

可以在 Amazon Linux 以及 Amazon Linux 2 的 Amazon 系统映像 (AMI) 中安装 amazon-efs-utils 软件包。

### Note

如果使用的是 AWS Direct Connect，您可以在[演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 \(p. 127\)](#)中找到安装说明。

### 安装 amazon-efs-utils 软件包

1. 确保您创建了一个 Amazon Linux 或 Amazon Linux 2 EC2 实例。有关如何执行该操作的信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[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。有关如何执行该操作的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[使用 SSH 连接到您的 Linux 实例](#)。
3. 运行以下命令以安装 amazon-efs-utils。

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

## 在其他 Linux 发行版上安装 amazon-efs-utils 软件包

如果不希望从 Amazon Linux 或 Amazon Linux 2 AMI 中获取 amazon-efs-utils 软件包，也可以从 GitHub 中获取 amazon-efs-utils 软件包。

### 从 GitHub 中克隆 amazon-efs-utils

1. 确保您创建了一个支持的 AMI 类型的 Amazon EC2 实例。有关如何执行该操作的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[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。有关如何执行该操作的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[使用 SSH 连接到您的 Linux 实例](#)。
3. 如果尚未安装 git，请使用以下命令进行安装。

```
sudo yum -y install git
```

4. 从终端中，使用以下命令将 amazon-efs-utils 工具从 GitHub 克隆到所选的目录中。

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

由于您需要使用 bash 命令 make，如果您的操作系统尚未安装 bash，您可以使用以下命令进行安装。

```
sudo yum -y install make
```

在克隆该软件包后，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 – Amazon Linux、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。

#### 作为 RPM 软件包构建并安装 amazon-efs-utils

1. 在客户端上打开一个终端，然后导航到具有从 GitHub 克隆的 amazon-efs-utils 软件包的目录（例如，“/home/centos/efs-utils”）。
2. 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. 使用以下命令构建该软件包。

```
sudo make rpm
```

4. 使用以下命令安装 amazon-efs-utils 软件包。

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

#### 作为 DEB 软件包构建并安装 amazon-efs-utils

1. 在客户端上打开一个终端，然后导航到具有从 GitHub 克隆的 amazon-efs-utils 软件包的目录。
2. 安装 binutils 软件包，这是用于构建 DEB 软件包的依赖项。

```
sudo apt-get -y install binutils
```

3. 使用以下命令构建该软件包。

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

4. 使用以下命令安装该软件包。

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

## 升级 stunnel

要将传输中的数据加密与 Amazon EFS 挂载帮助程序一起使用，需要使用 OpenSSL 1.0.2 或更高版本以及支持 OSCP 和证书主机名检查的 stunnel 版本。Amazon 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 版本之一时，使用 TLS 挂载 Amazon EFS 文件系统将失败。

在安装 Amazon 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。

#### 升级 stunnel

1. 在 Linux 客户端上打开一个终端，然后按顺序运行以下命令。
2. `sudo yum install -y gcc openssl-devel tcp_wrappers-devel`
3. `sudo curl -o stunnel-5.46.tar.gz https://www.stunnel.org/downloads/stunnel-5.45.tar.gz`
4. `sudo tar xvfz stunnel-5.46.tar.gz`
5. `cd stunnel-5.46/`
6. `sudo ./configure`

7. `sudo make`
8. 当前 `amazon-efs-utils` 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
```

9. `sudo make install`
10. **Note**

默认 CentOS shell 为 `csh`，它使用与 `bash` shell 不同的语法。以下代码先调用 `bash`，然后运行。

```
bash
```

```
if [[ -f /bin/stunnel ]]; then
sudo mv /bin/stunnel /root
fi
```

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

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

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 OCSP 和证书主机名检查。我们建议您不要在生产环境中禁用这些功能。要禁用 OCSP 和证书主机名检查，请执行以下操作：

1. 使用所选的文本编辑器打开 `/etc/amazon/efs/efs-utils.conf` 文件。
2. 将 `stunnel_check_cert_hostname` 值设置为 `false`。
3. 将 `stunnel_check_cert_validity` 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[挂载文件系统 \(p. 59\)](#)。

## EFS 挂载帮助程序

Amazon EFS 挂载帮助程序简化了挂载文件系统的过程。默认情况下，它包括 Amazon EFS 建议的挂载选项。此外，挂载帮助程序还具有内置的日志记录以进行故障排除。如果您遇到 Amazon EFS 文件系统问题，您可以与 AWS Support 分享这些日志。

### 如何使用

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实例引导时自动使用 `/etc/fstab` 配置文件中的条目挂载 Amazon 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 \(p. 99\)](#)。

在将传输中的数据加密声明为 Amazon EFS 文件系统的挂载选项时，挂载帮助程序初始化客户端 `stunnel` 进程和名为 `amazon-efs-mount-watchdog` 的监管进程。`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[在 EFS 中加密数据和元数据 \(p. 86\)](#)。

## 使用 EFS 挂载帮助程序

挂载帮助程序帮助您在 Linux EC2 实例上挂载您的 EFS 文件系统。有关更多信息，请参阅[挂载文件系统 \(p. 59\)](#)。

## 获取支持日志

挂载帮助程序具有 Amazon EFS 文件系统的内置日志记录。您可以与 AWS Support 分享这些日志以进行故障排除。

对于安装了挂载帮助程序的系统，您可以查找在 `/var/log/amazon/efs` 中存储的日志。这些日志适用于挂载帮助程序、stunnel 进程本身以及监控 stunnel 进程的 `amazon-efs-mount-watchdog` 进程。

### Note

watchdog 进程确保每个挂载的 stunnel 进程正在运行，并在卸载 Amazon 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。

您可以在 `/etc/amazon/efs/amazon-efs-utils.conf` 中更改日志配置。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。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。

## 将 amazon-efs-utils 与 AWS Direct Connect 一起使用

在使用 AWS Direct Connect 连接到您的 Amazon VPC 时，您可以在本地数据中心服务器上挂载 Amazon EFS 文件系统。使用 `amazon-efs-utils` 还可以简化使用挂载帮助程序进行挂载的过程，并允许您启用传输中的数据加密。要了解如何将 `amazon-efs-utils` 与 AWS Direct Connect 一起使用以将 Amazon EFS 文件系统挂载到本地 Linux 客户端上，请参阅[演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 \(p. 127\)](#)。

## 相关主题

有关 Amazon EFS 挂载帮助程序的更多信息，请参阅以下相关主题：

- [在 EFS 中加密数据和元数据 \(p. 86\)](#)
- [挂载文件系统 \(p. 59\)](#)

# 管理 Amazon EFS 文件系统

文件系统管理任务是指创建和删除文件系统，管理标签和管理现有文件的网络可访问性。管理网络可访问性涉及创建和管理挂载目标。

您可以使用 Amazon EFS 控制台、AWS Command Line Interface (AWS CLI) 或以编程方式执行这些文件系统管理任务，如以下各节中所述。

## 主题

- [管理文件系统网络可访问性 \(p. 39\)](#)
- [管理文件系统标签 \(p. 45\)](#)
- [计量 – Amazon EFS 如何报告文件系统和对象大小 \(p. 46\)](#)
- [管理 Amazon EFS 文件同步 \(p. 47\)](#)
- [删除 Amazon EFS 文件系统 \(p. 56\)](#)
- [管理对加密的文件系统的访问 \(p. 57\)](#)

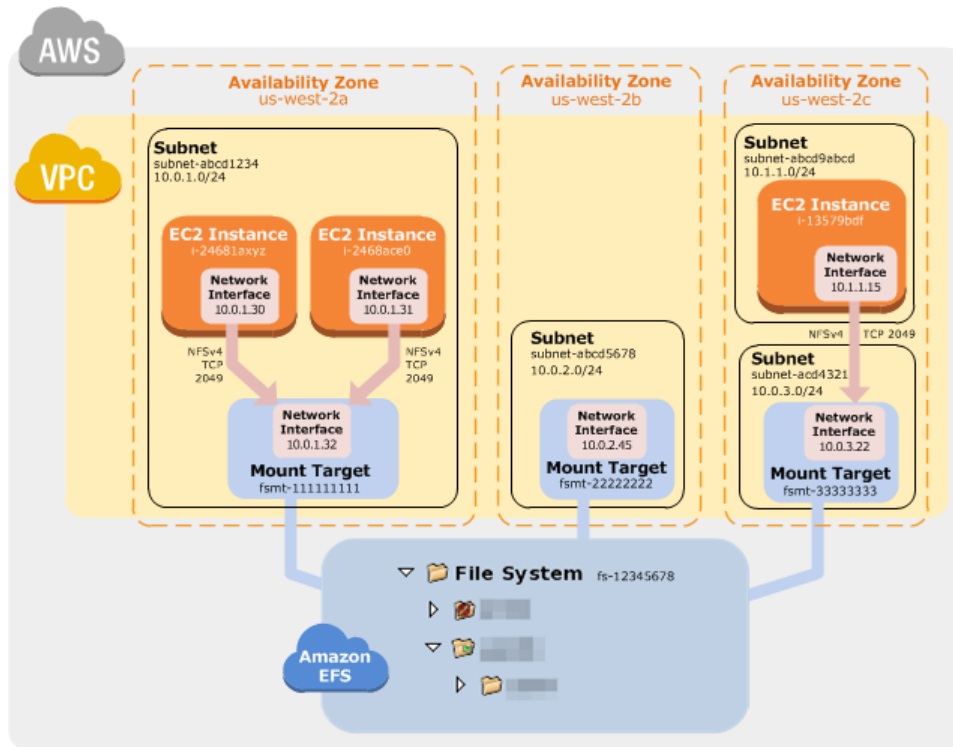
如果您是刚接触 Amazon EFS 的新用户，我们建议您尝试以下练习，先全面感受一下 Amazon EFS 文件系统的使用体验：

- [入门 \(p. 9\)](#) – 该练习提供基于控制台的端到端设置，您在该练习中创建一个文件系统，在 EC2 实例上挂载该文件系统，然后测试该设置。控制台可为您处理许多事务，从而帮助您快速设置端到端体验。
- [演练 1：使用 AWS CLI 创建 Amazon EFS 文件系统并将其挂载到 EC2 实例上 \(p. 110\)](#) – 该演练与入门练习类似，但它使用 AWS CLI 执行大部分任务。由于 CLI 命令紧密映射到 Amazon EFS API，因此该演练可以帮助您熟悉 Amazon EFS API。

## 管理文件系统网络可访问性

使用您为文件系统创建的挂载目标，将文件系统挂载到 VPC 中的 EC2 实例上。管理文件系统网络可访问性是指管理挂载目标。

下图显示了 VPC 中的 EC2 实例如何使用挂载目标访问 Amazon EFS 文件系统。



该图显示了在访问 Amazon 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。

管理挂载目标是指以下活动：

- 在 VPC 中创建和删除挂载目标 – 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

我们建议您在所有可用区中创建挂载目标，以便轻松将文件系统挂载到您可能在任何可用区中启动的 EC2 实例上。

如果删除挂载目标，则操作将通过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。

一次只能在一个 VPC 中使用一个文件系统。也就是说，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，则必须从当前 VPC 中删除挂载目标，然后在另一个 VPC 中创建新的挂载目标。

- 更新挂载目标配置 - 创建挂载目标时，会将安全组与挂载目标相关联。安全组充当虚拟防火墙，用于控制进出挂载目标的流量。您可以添加入站规则以控制对挂载目标的访问，从而控制对文件系统的访问。创建挂载目标后，您可能需要修改分配给它们的安全组。

每个挂载目标还有一个 IP 地址。创建挂载目标时，可以从放置挂载目标的子网中选择一个 IP 地址。如果省略值，Amazon EFS 会从该子网中选择未使用的 IP 地址。

创建挂载目标后，没有可用来更改 IP 地址的 Amazon EFS 操作，因此无法通过编程方式或使用 AWS CLI 更改 IP 地址。但是可使用控制台来更改 IP 地址。在幕后，控制台将删除挂载目标并再次创建挂载目标。



## Warning

如果更改挂载目标的 IP 地址，则会中断任何现有的文件系统挂载，而需要重新挂载文件系统。

对文件系统网络可访问性进行的任何配置更改不会影响文件系统本身。您的文件系统和数据保持不变。

以下几节提供了有关管理文件系统的网络可访问性的信息。

## 主题

- [在 VPC 中创建或删除挂载目标 \(p. 41\)](#)
- [在另一个 VPC 中创建挂载目标 \(p. 43\)](#)
- [更新挂载目标配置 \(p. 44\)](#)

# 在 VPC 中创建或删除挂载目标

要访问 VPC 中的 Amazon EFS 文件系统，您需要使用挂载目标。对于 Amazon EFS 文件系统，必须满足以下条件：

- 您可以在每个可用区中创建一个挂载目标。
- 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。

## Note

我们建议您在每个可用区中分别创建一个挂载目标。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](#)。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。

有关操作的更多信息，请参阅 [CreateMountTarget \(p. 164\)](#)。

您可以删除挂载目标。删除挂载目标将会强制中断通过该挂载目标的任何文件系统挂载，这可能会中断使用这些挂载的实例或应用程序。有关更多信息，请参阅 [DeleteMountTarget \(p. 176\)](#)。

## 使用控制台

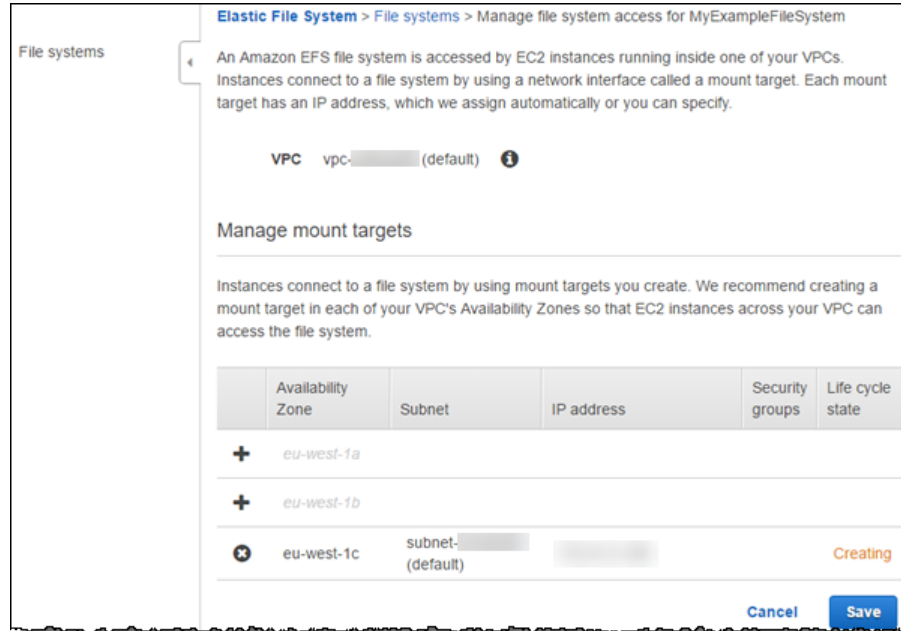
使用 AWS 管理控制台通过以下过程创建新挂载目标、删除或更新现有挂载目标。

1. 在 Amazon EFS 控制台中，选择文件系统，选择操作，然后选择管理文件系统访问。

控制台将显示管理文件系统访问页，其中列出您在选定的 VPC 中创建的文件系统挂载目标。还将显示可用区列表和挂载目标信息 (如果该可用区中有挂载目标)。

控制台显示文件系统在 eu-west-2c 可用区中具有一个挂载目标，如下所示：





## 2. 创建新挂载目标

- 单击特定的可用区行左侧。
- 如果可用区具有多个子网，请从子网列表中选择一个子网。
- Amazon EFS 会自动选择可用 IP 地址，或者您也可以显式提供另一个 IP 地址。
- 从列表中选择一个安全组。

有关安全组的更多信息，请参阅Amazon EC2 用户指南（适用于 Linux 实例）中的[Amazon EC2 安全组](#)。

- 要删除挂载目标，请选择要从中删除挂载目标的可用区旁边的 X。

## 使用 AWS CLI

要创建挂载目标，请使用 `create-mount-target` AWS CLI 命令（相应的操作是 [CreateMountTarget](#) (p. 164))，如下所示：

```
$ aws efs create-mount-target \
--file-system-id file-system-ID (for which to create the mount target) \
--subnet-id vpc-subnet-ID (in which to create mount target) \
--security-group security-group IDs (to associate with the mount target) \
--region aws-region (for example, us-west-2) \
--profile adminuser
```

AWS 区域（`region` 参数）必须是 VPC 区域。

您可以使用 `describe-mount-targets` AWS CLI 命令（相应的操作是 [DescribeMountTargets](#) (p. 185)) 获取为文件系统创建的挂载目标的列表，如下所示：

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

下面是示例响应：

```
{
  "MountTargets": [
    {
      "MountTargetId": "fsmt-52a643fb",
      "NetworkInterfaceId": "eni-f11e8395",
      "FileSystemId": "fs-6fa144c6",
      "LifeCycleState": "available",
      "SubnetId": "subnet-15d45170",
      "OwnerId": "23124example",
      "IpAddress": "10.0.2.99"
    },
    {
      "MountTargetId": "fsmt-55a643fc",
      "NetworkInterfaceId": "eni-14a6ae4d",
      "FileSystemId": "fs-6fa144c6",
      "LifeCycleState": "available",
      "SubnetId": "subnet-0b05fc52",
      "OwnerId": "23124example",
      "IpAddress": "10.0.19.174"
    }
  ]
}
```

要删除现有挂载目标，请使用 `delete-mount-target` AWS CLI 命令 (相应的操作是 [DeleteMountTarget \(p. 176\)](#))，如下所示：

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

## 在另一个 VPC 中创建挂载目标

一次只能在一个 VPC 中使用 Amazon 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从该 VPC 中的 EC2 实例提供对该文件系统的访问权限。要从另一个 VPC 中的 EC2 实例访问文件系统，则必须先从当前 VPC 中删除挂载目标，然后在另一个 VPC 中创建新的挂载目标。

## 在 Amazon EFS 中使用 VPC 对等

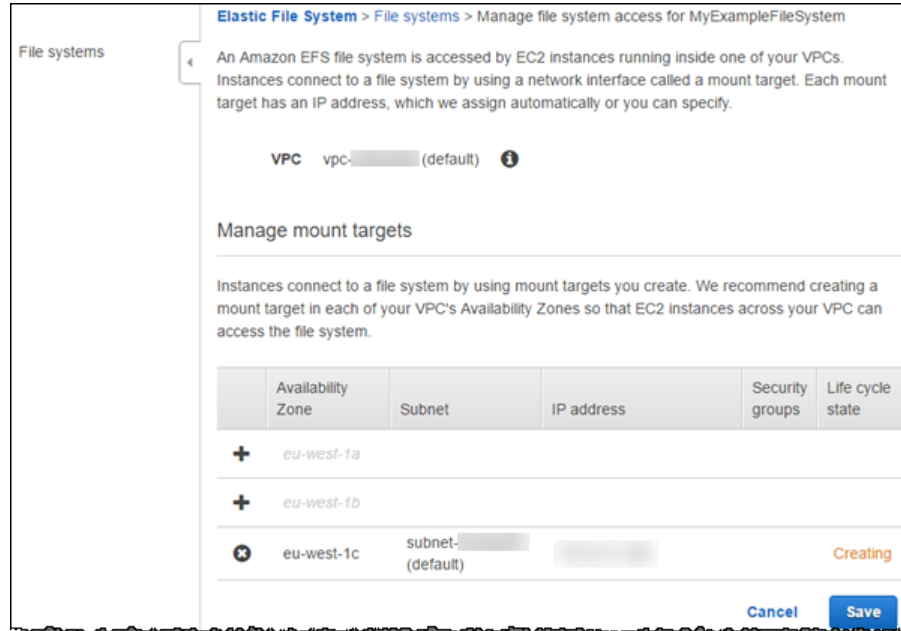
VPC 对等连接是两个 VPC 之间的网络连接，您可以通过该连接使用私有 Internet 协议版本 (IPv4) 或 Internet 协议版本 6 (IPv6) 地址在两个 VPC 之间路由流量。有关 VPC 对等的更多信息，请参阅 Amazon VPC Peering Guide 中的 [什么是 VPC 对等](#)。

对于 Amazon EFS，在使用 C5 或 M5 实例时，您可以在单个 AWS 区域中使用 VPC 对等。但是，不支持使用其他实例类型的其他 VPC 私有连接机制，如 VPN 连接、区域间 VPC 对等和区域内 VPC 对等。

## 使用控制台

1. 在 Amazon EFS 控制台中，选择文件系统，选择操作，然后选择管理文件系统访问。

控制台将显示管理文件系统访问页，其中列出您为 VPC 中的文件系统创建的挂载目标。下图显示了一个具有三个挂载目标的文件系统，每个可用区中一个。



2. 要更改 VPC，请从 VPC 列表中选择另一个 VPC。

控制台将清除所有挂载目标信息，并仅列出可用区。

3. 在一个或多个可用区中创建挂载目标，如下所示：

- a. 如果可用区具有多个子网，请从子网列表选择一个子网。
- b. Amazon EFS 会自动选择可用 IP 地址，或者您也可以显式提供另一个 IP 地址。
- c. 选择您想关联的安全组。

有关安全组的信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 [Amazon EC2 安全组](#)。

4. 选择 Save (保存)。

控制台首先从先前的 VPC 中删除挂载目标，然后在您选择的新 VPC 中创建新的挂载目标。

## 使用 CLI

要在另一个 VPC 中使用文件系统，您必须先删除先前在 VPC 中创建的任何挂载目标，然后在另一个 VPC 中创建新的挂载目标。有关示例 AWS CLI 命令，请参阅在 [VPC 中创建或删除挂载目标](#)。

## 更新挂载目标配置

为文件系统创建挂载目标后，可能需要更新生效的安全组。您不能更改现有挂载目标的 IP 地址。要更改 IP 地址，您必须删除挂载目标并使用新地址创建一个新目标。删除挂载目标将会中断任何现有的文件系统挂载。

## 修改安全组

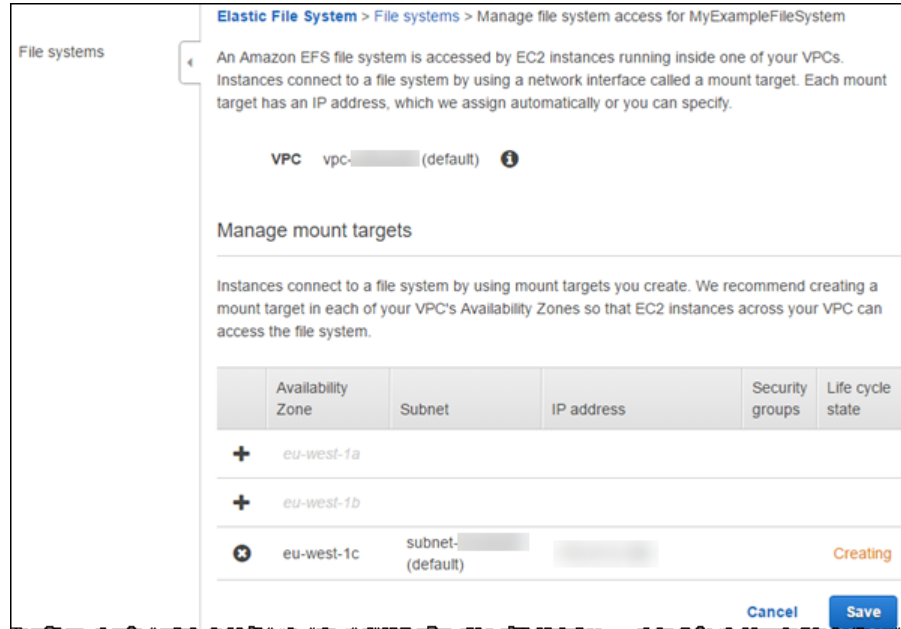
安全组定义入站/出站访问。当您更改与挂载目标相关联的安全组时，请确保您授权必要的入站/出站访问，以便 EC2 实例可以与文件系统通信。

有关安全组的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 [Amazon EC2 安全组](#)。

## 使用控制台

1. 在 Amazon EFS 控制台中，选择文件系统，选择操作，然后选择管理文件系统访问。

控制台将显示管理文件系统访问页，其中包含可用区列表和挂载目标信息（如果在可用区中具有挂载目标）。



2. 在安全组列中，您可以添加或删除安全组。可以选择 X 以删除现有的安全组。可以选择安全组框以从其他可用安全组中进行选择。

如果删除所有安全组，Amazon EFS 将分配 VPC 的默认安全组。

## 使用 CLI

要修改对挂载目标有效的安全组，请使用 `modify-mount-target-security-groups` AWS CLI 命令（相应的操作是 [ModifyMountTargetSecurityGroups \(p. 194\)](#)）替换任何现有安全组，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
```

# 管理文件系统标签

您可以创建新标签，更新现有标签的值或删除与文件系统关联的标签。

## 使用控制台

控制台会列出与文件系统关联的现有标签。您可以添加新标签，更改现有标签的值或删除现有标签。

1. 通过 <https://console.aws.amazon.com/efs/> 打开 Amazon Elastic File System 控制台。
2. 选择文件系统。

3. 选择操作，然后选择管理标签。
4. 在管理标签页中，添加或删除标签。对于每个新标签，请提供键及其值。
5. 选择 Save (保存)。

## 使用 AWS CLI

您可以使用 `create-tags` CLI 命令添加新标签，使用 `delete-tags` 命令删除现有标签，或使用 `describe-tags` 命令来检索与文件系统关联的标签。每个 CLI 命令分别对应于 [CreateTags \(p. 171\)](#)、[DeleteTags \(p. 179\)](#) 和 [DescribeTags \(p. 191\)](#) Amazon EFS 操作。

有关可用于添加和列出标签的 AWS CLI 命令的示例演示，请参阅 [步骤 2.1：创建 Amazon EFS 文件系统 \(p. 115\)](#)。

以下 `delete-tags` 命令将从指定文件系统的标签列表中删除标签键 `test1` 和 `test2`。

```
$ aws efs \
delete-tags \
--file-system-id fs-c5a1446c \
--tag-keys "test1" "test2" \
--region us-west-2 \
--profile adminuser
```

## 计量 – Amazon EFS 如何报告文件系统和对象大小

本节介绍 Amazon EFS 如何报告文件系统大小和文件系统内对象的大小。

### 计量 Amazon EFS 文件系统对象

Amazon EFS 系统中的客户可见对象可以是常规文件、目录、符号链接和特殊文件 (FIFO 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，如果使用的实际存储小于舍入到下一个 4 KiB 增量的逻辑大小，则 Amazon EFS 报告用作计量的数据大小的实际存储。
- 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` 属性报告用于对象的空间时，它包括对象的当前计量数据大小，但不包括其元数据大小。有两个实用程序可用于测量文件的磁盘使用情况，它们是 `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，然后用该值乘以数据块大小。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
```

## 计量 Amazon EFS 文件系统

整个 Amazon EFS 文件系统的计量大小是其所有当前对象大小的总和 (包括元数据)。每个对象的大小根据表示计量小时 (例如上午 8:00 到上午 9:00 这一小时) 内的对象大小的代表性取样计算得出。

例如，空文件对其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 Amazon EFS 文件系统的这一计量大小：

- DescribeFileSystems API – 在软件开发工具包、HTTP 和 AWS CLI 中使用。
- 文件系统表 – 对于 AWS 管理控制台中列出的每个文件系统。
- DF 命令 – 在 Linux 中，可在 EC2 实例的终端提示符处运行 `df` 命令。请使用 `df` 命令，而不是 `du` 命令。不要在文件系统的根目录中使用 `du` 命令以进行存储计量。这些结果不会提供完整数据。

### Note

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 [随突发模式扩展的吞吐量 \(p. 80\)](#)。

## 管理 Amazon EFS 文件同步

在本节中，您可以找到有关如何管理 Amazon EFS 文件同步的信息。

### 主题

- [删除同步代理 \(p. 47\)](#)
- [删除同步任务 \(p. 48\)](#)
- [了解同步代理状态 \(p. 48\)](#)
- [了解同步任务状态 \(p. 48\)](#)
- [在 EFS 文件同步虚拟机本地控制台上执行任务 \(p. 49\)](#)
- [在 Amazon EC2 EFS 文件同步本地控制台上执行任务 \(p. 54\)](#)

## 删除同步代理

如果不再需要使用同步代理，您可以从 Amazon EFS 管理控制台中删除该代理。

### 删除同步代理

1. 选择文件同步，选择代理，然后选择要删除的同步代理。

2. 对于 Actions，选择 Delete。
3. 在确认删除同步代理对话框中，选中确认删除复选框，然后选择确定。

## 删除同步任务

如果不再需要使用同步任务，您可以从 Amazon EFS 管理控制台中删除该任务。

### 删除同步任务

1. 选择文件同步，选择任务，然后选择要删除的同步任务。
2. 对于 Actions，选择 Delete。
3. 在确认删除同步任务对话框中，选中确认删除复选框，然后选择确定。

## 了解同步代理状态

下表描述了各种同步代理状态，以及您是否以及何时应根据状态采取措施。在使用同步代理时，它在所有或大部分时间内具有正在运行状态。

同步代理状态	意义
正在运行	已正确配置同步代理并且可供使用。“正在运行”状态是同步代理的正常运行状态。
离线	已关闭同步代理的虚拟机或 EC 实例，或者代理处于不正常运行状态。在解决导致不正常运行状态的问题后，代理将恢复为“正在运行”状态。

## 了解同步任务状态

下表描述了各种同步任务状态，以及您是否以及何时应根据状态采取措施。

同步任务状态	意义
Available	已正确配置同步任务，并且可以启动该任务。
Completed	任务创建过程已完成。
创建	EFS 文件同步正在创建同步任务。
正在启动	已启动任务创建过程。
正在准备	同步任务正在检查源和目标文件系统以确定要同步的文件。
正在同步	EFS 文件同步正在将文件从源文件系统同步到目标 Amazon EFS 文件系统。
正在验证	EFS 文件同步正在验证源和目标文件系统之间的一致性。



## 在 EFS 文件同步虚拟机本地控制台上执行任务

对于在本地部署的 EFS 文件同步，您可以使用虚拟机主机的本地控制台执行以下维护任务。

主题

- [使用默认凭证登录本地控制台 \(p. 49\)](#)
- [配置 EFS 文件同步网络 \(p. 50\)](#)
- [查看 EFS 文件同步系统资源状态 \(p. 52\)](#)
- [同步 EFS 文件同步虚拟机时间 \(p. 53\)](#)
- [在本地控制台上运行 EFS 文件同步命令 \(p. 53\)](#)

### 使用默认凭证登录本地控制台

在 VM 做好登录准备时，登录屏幕将显示。

登录到 EFS 文件同步的本地控制台

- 如果这是您首次登录到本地控制台，请使用用户名 admin 和密码 password 登录到虚拟机。否则，请使用您的凭证登录。

在登录后，您将看到 Amazon EFS 文件同步 Configuration (配置) 主菜单，如以下屏幕截图所示。

```
AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 192.168.145.137
#####

1: Network Configuration
2: View System Resource Check (0 Errors)
3: System Time Management
4: Command Prompt

Press "x" to exit session

Enter command: _
```

Note

我们建议您更改默认密码，为此，请从 EFS 文件同步Command Prompt (命令提示符) (主菜单上的第 5 项) 中运行 passwd 命令。有关如何运行该命令的信息，请参阅[在本地控制台上运行 EFS 文件同步命令 \(p. 53\)](#)。

收款人	请参阅
配置网络	<a href="#">配置 EFS 文件同步网络 (p. 50)</a> 。
查看系统资源检查	<a href="#">查看 EFS 文件同步系统资源状态 (p. 52)</a> 。
管理 VM 时间	<a href="#">同步 EFS 文件同步虚拟机时间 (p. 53)</a> 。
运行本地控制台命令	<a href="#">在本地控制台上运行 EFS 文件同步命令 (p. 53)</a> 。



要关闭 EFS 文件同步，请键入 0。

要退出配置会话，请键入 x 退出菜单。

## 配置 EFS 文件同步网络

EFS 文件同步的默认网络配置是动态主机配置协议 (DHCP)。在使用 DHCP 时，将为您 EFS 文件同步自动分配 IP 地址。在某些情况下，您可能需要手动将 EFS 文件同步的 IP 分配为静态 IP 地址，如下所述。

配置 EFS 文件同步以使用静态 IP 地址

1. 登录到 EFS 文件同步的本地控制台。
2. 在 Amazon EFS 文件同步 Configuration (配置) 主菜单中，键入选项 1 以开始配置静态 IP 地址。

```
AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 192.168.145.137
#####

1: Network Configuration
2: View System Resource Check (0 Errors)
3: System Time Management
4: Command Prompt

Press "x" to exit session

Enter command: _
```

3. 在 Amazon EFS 文件同步 Configuration (配置) 菜单中选择以下选项之一：

```
AWS Appliance Activation - Network Configuration

1: Describe Adapter
2: Configure DHCP
3: Configure Static IP
4: Reset all to DHCP
5: Set Default Adapter
6: View DNS Configuration
7: View Routes

Press "x" to exit

Enter command: _
```

目的	请执行此操作
描述网络适配器	<p>键入选项 1。</p> <p>将显示适配器名称的列表，并且系统会提示您输入一个适配器名称 — 例如 <b>eth0</b>。如果您指定的适配器正在使用中，有关该适配器的下列信息就会显示：</p> <ul style="list-style-type: none"><li>• 媒体访问控制 (MAC) 地址</li><li>• IP 地址</li></ul>

目的	请执行此操作
	<ul style="list-style-type: none"><li>• 网络掩码</li><li>• EFS 文件同步 IP 地址</li><li>• DHCP 启用状态</li></ul> <p>在配置静态 IP 地址（选项 3）和设置 EFS 文件同步的默认路由适配器（选项 5）时，您可以使用相同的适配器名称。</p>
配置 DHCP	<p>键入选项 2。</p> <p>系统将提示您将网络接口配置为使用 DHCP。</p>
为 EFS 文件同步配置静态 IP 地址	<p>键入选项 3。</p> <p>系统会提示您键入以下信息以配置静态 IP：</p> <ul style="list-style-type: none"><li>• 网络适配器名称</li><li>• IP 地址</li><li>• 网络掩码</li><li>• 默认 EFS 文件同步地址</li><li>• 主要域名服务 (DNS) 地址</li><li>• 备用 DNS 地址</li></ul> <p><b>Important</b></p> <p>如果已激活 EFS 文件同步，您必须从 EFS 文件同步控制台中将其关闭，然后重新启动以使设置生效。</p> <p>如果 EFS 文件同步使用多个网络接口，您必须将所有启用的接口设置为使用 DHCP 或静态 IP 地址。</p> <p>例如，假定您的 EFS 文件同步虚拟机使用两个配置为 DHCP 的接口。如果您稍后将一个接口设置为静态 IP，则会禁用另一个接口。在这种情况下，如需启用此接口，您必须将其设置为静态 IP。</p> <p>如果两个接口最初设置为使用静态 IP 地址，然后将 EFS 文件同步设置为使用 DHCP，两个接口将使用 DHCP。</p>
将 EFS 文件同步的所有网络配置重置为 DHCP	<p>键入选项 4。</p> <p>所有网络接口均设置为使用 DHCP。</p> <p><b>Important</b></p> <p>如果已激活 EFS 文件同步，您必须从 EFS 文件同步控制台中关闭 EFS 文件同步，然后重新启动以使设置生效。</p>

目的	请执行此操作
设置 EFS 文件同步的默认路由适配器	键入选项 5。  将显示 EFS 文件同步的可用适配器，并提示您选择其中的一个适配器，例如， <b>eth0</b> 。
查看 EFS 文件同步的 DNS 配置	键入选项 6。  主 DNS 和备用 DND 域名服务器的 IP 地址将会显示。
查看路由表	键入选项 7。  将显示 EFS 文件同步的默认路由。

## 查看 EFS 文件同步系统资源状态

在您的网关启动时，它检查其虚拟 CPU 核心数、根卷大小以及 RAM，并确定这些系统资源是否足以使 EFS 文件同步正常工作。您可以在 EFS 文件同步的本地控制台上查看该检查的结果。

查看系统资源检查的状态

1. 登录到 EFS 文件同步的本地控制台。
2. 在 EFS 文件同步 Configuration (配置) 主菜单中，键入 **2** 以查看系统资源检查结果。

```

AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 192.168.145.137
#####

1: Network Configuration
2: View System Resource Check (0 Errors)
3: System Time Management
4: Command Prompt

Press "x" to exit session

Enter command: _

```

控制台为每个资源显示 [OK]、[WARNING] 或 [FAIL] 消息，如下表中所述。

消息	描述
[确定]	该资源通过了系统资源检查。
[警告]	资源不满足建议的要求，但 EFS 文件同步将继续正常工作。EFS 文件同步显示一条消息以描述资源检查结果。
[FAIL]	资源不满足最低要求。EFS 文件同步可能无法正常工作。EFS 文件同步显示一条消息以描述资源检查结果。

控制台还会在资源检查菜单选项旁边显示错误和警告的数量。

以下屏幕截图显示 [FAIL] 消息和随附的错误消息。

## 同步 EFS 文件同步虚拟机时间

在部署并运行 EFS 文件同步后，在某些情况下，EFS 文件同步虚拟机的时间可能会出现偏差。例如，如果网络中断时间较长，并且管理程序主机和 EFS 文件同步没有更新时间，EFS 文件同步虚拟机的时间将与实际时间不同。当出现时间偏差时，操作（如快照）发生的预计时间和操作发生的实际时间之间会有差异。

对于在 VMware ESXi 上部署的 EFS 文件同步，设置管理程序主机时间并将虚拟机时间与主机同步就足以避免时间偏差。

## 在本地控制台上运行 EFS 文件同步命令

EFS 文件同步控制台帮助提供安全的环境以配置和诊断 EFS 文件同步问题。利用控制台命令，您可以执行维护任务，如保存路由表或连接到 AWS Support。

运行配置或诊断命令

1. 登录到 EFS 文件同步的本地控制台。
2. 在 EFS 文件同步 Configuration (配置) 主菜单上，为 Command Prompt (命令提示符) 键入选项 **4**。

```
AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 172.16.145.137
#####

1: Network Configuration
2: View System Resource Check (0 Errors)
3: System Time Management
4: Command Prompt

Press "x" to exit session

Enter command: _
```

3. 在 EFS 文件同步控制台上，键入 **h**，然后按 Return 键。

控制台将显示包含可用命令的 Available Commands (可用的命令) 菜单，并在该菜单后面显示一个命令提示符，如以下屏幕截图所示。

```
AVAILABLE COMMANDS
ip                Show / manipulate routing, devices, and tunnels
save-routing-table Save newly added routing table entry
ifconfig          View or configure network interfaces
iptables         Administration tool for IPv4 packet filtering and NAT
save-iptables     Persist IP tables
passwd            Update authentication tokens
open-support-channel Connect to AWS Support
h                Display available command list
exit             Return to Configuration menu

Command: _
```

### Note

将在 EFS 文件同步本地控制台中禁用 man 命令。

## 在 Amazon EC2 EFS 文件同步本地控制台上执行任务

在运行在 Amazon EC2 实例上部署的 EFS 文件同步时，某些维护任务要求您登录到本地控制台。在本节中，您可以在找到有关如何登录到本地控制台并执行维护任务的信息。

### 主题

- [登录到 Amazon EC2 EFS 文件同步本地控制台 \(p. 54\)](#)
- [查看 EFS 文件同步系统资源状态 \(p. 54\)](#)
- [在本地控制台上运行 EFS 文件同步命令 \(p. 55\)](#)

## 登录到 Amazon EC2 EFS 文件同步本地控制台

您可以使用安全外壳 (SSH) 客户端连接至 Amazon EC2 实例。有关详细信息，请参阅 Amazon EC2 用户指南中的[连接到您的实例](#)。要以这种方式连接，您需要在启动实例时指定的 SSH 密钥对。有关 Amazon EC2 密钥对的信息，请参阅 Amazon EC2 用户指南中的[Amazon EC2 密钥对](#)。

### 登录到 EFS 文件同步本地控制台

1. 登录到本地控制台。如果从 Windows 计算机连接到 EC2 实例，请使用用户名 admin 和密码 password 登录。否则，请使用您的凭证登录。
2. 在登录后，您将看到 Amazon EFS 文件同步 Configuration (配置) 主菜单，如下屏幕截图所示。

```
AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 172.31.29.255
#####

1: View System Resource Check (0 Errors)
2: Command Prompt

Press "x" to exit session

Enter command: █
```

目的	请参阅
查看系统资源检查	<a href="#">查看 EFS 文件同步系统资源状态 (p. 54)</a> 。
运行 EFS 文件同步控制台命令	<a href="#">在本地控制台上运行 EFS 文件同步命令 (p. 55)</a>

要关闭 EFS 文件同步，请键入 0。

要退出配置会话，请键入 x 退出菜单。

## 查看 EFS 文件同步系统资源状态

在 EFS 文件同步启动时，它检查其虚拟 CPU 核心数、根卷大小以及 RAM，并确定这些系统资源是否足以使 EFS 文件同步正常工作。您可以在 EFS 文件同步的本地控制台上查看该检查的结果。

## 查看系统资源检查的状态

1. 登录到 EFS 文件同步的本地控制台。有关说明，请参阅[登录到 Amazon EC2 EFS 文件同步本地控制台 \(p. 54\)](#)。
2. 在 Amazon EFS 文件同步 Configuration (配置) 主菜单中，键入 1 以查看系统资源检查结果。

```
AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 172.31.29.28
#####

1: View System Resource Check (0 Errors)
2: Command Prompt

Press "x" to exit session

Enter command: █
```

控制台为每个资源显示 [OK]、[WARNING] 或 [FAIL] 消息，如下表中所述。

消息	描述
[确定]	该资源通过了系统资源检查。
[警告]	资源不满足建议的要求，但 EFS 文件同步将继续正常工作。EFS 文件同步显示一条消息以描述资源检查结果。
[FAIL]	资源不满足最低要求。EFS 文件同步可能无法正常工作。EFS 文件同步显示一条消息以描述资源检查结果。

控制台还会在资源检查菜单选项旁边显示错误和警告的数量。

以下屏幕截图显示 [FAIL] 消息和随附的错误消息。

## 在本地控制台上运行 EFS 文件同步命令

EFS 文件同步本地控制台帮助提供安全的环境以配置和诊断 EFS 文件同步问题。通过使用本地控制台命令，您可以执行维护任务，例如，保存路由表或连接到 AWS Support。

### 运行配置或诊断命令

1. 登录到 EFS 文件同步的本地控制台。有关说明，请参阅[登录到 Amazon EC2 EFS 文件同步本地控制台 \(p. 54\)](#)。
2. 在 Amazon EFS 文件同步 Configuration (配置) 主菜单中，为 EFS 文件同步 Console (EFS 文件同步控制台) 键入 2。

```
AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 172.31.255.255
#####

1: View System Resource Check (0 Errors)
2: Command Prompt

Press "x" to exit session

Enter command: █
```

3. 在 EFS 文件同步控制台中，键入 **h**，然后按 Return 键。

控制台将显示包含可用命令的 Available Commands 菜单。将在该菜单后面显示 EFS 文件同步 Console (EFS 文件同步控制台) 提示，如以下屏幕截图所示。

```
AVAILABLE COMMANDS
ip                Show / manipulate routing, devices, and tunnels
save-routing-table Save newly added routing table entry
ifconfig          View or configure network interfaces
iptables         Administration tool for IPv4 packet filtering and NAT
save-iptables     Persist IP tables
open-support-channel Connect to AWS Support
h                Display available command list
exit             Return to Configuration menu

Command: █
```

#### Note

将在 EFS 文件同步本地控制台中禁用 man 命令。

## 删除 Amazon EFS 文件系统

文件系统删除是一种无法撤销的破坏性操作。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。

#### Important

应始终在删除之前卸载文件系统。

### 使用控制台

1. 通过 <https://console.aws.amazon.com/efs/> 打开 Amazon Elastic File System 控制台。
2. 选择要删除的文件系统。
3. 选择操作，然后选择删除文件系统。
4. 在永久删除文件系统确认框中，键入文件系统 ID，然后选择删除文件系统。

控制台简化了文件删除操作。首先，它将删除关联的挂载目标，然后删除文件系统。

## 使用 CLI

在使用 AWS CLI 命令删除文件系统之前，必须先删除为文件系统创建的所有挂载目标。

有关示例 AWS CLI 命令，请参阅 [步骤 4：清除](#) (p. 120)。

## 相关主题

[管理 Amazon EFS 文件系统](#) (p. 39)

# 管理对加密的文件系统的访问

通过使用 Amazon EFS，您可以创建加密的文件系统。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。您需要执行的任何密钥管理仅与静态加密相关。Amazon EFS 自动管理用于传输中加密的密钥。

如果创建使用静态加密的文件系统，则会静态加密数据和元数据。Amazon EFS 使用 AWS Key Management Service (AWS KMS) 进行密钥管理。在创建使用静态加密的文件系统时，您可以指定一个客户主密钥 (CMK)。CMK 可以是 `aws/elasticfilesystem` (适用于 Amazon EFS 的 AWS 托管 CMK)，也可以是您管理的 CMK。

文件数据 (文件内容) 是使用在创建文件系统时指定的 CMK 静态加密的。元数据 (文件名、目录名和目录内容) 是使用 Amazon EFS 管理的密钥加密的。

您的文件系统的 AWS 托管 CMK 用作文件系统元数据 (如文件名、目录名和目录内容) 的主密钥。您拥有用于静态加密文件数据 (文件内容) 的 CMK。

您管理哪些用户有权访问您的 CMK 以及您的加密文件系统的内容。该访问是由 AWS Identity and Access Management (IAM) 策略和 AWS KMS 控制的。IAM 策略控制用户对 Amazon EFS API 操作的访问。AWS KMS 密钥策略控制用户对在创建文件系统时指定的 CMK 的访问。有关更多信息，请参阅下列内容：

- IAM 用户指南 中的 [IAM 用户](#)
- AWS Key Management Service Developer Guide 中的 [在 AWS KMS 中使用密钥策略](#)
- AWS Key Management Service Developer Guide 中的 [使用授权](#)

作为密钥管理员，您可以导入外部密钥，以及启用、禁用或删除以修改密钥。您指定的 CMK 的状态 (在创建使用静态加密的文件系统时) 影响访问其内容。CMK 必须处于 `enabled` 状态，用户才能访问静态加密的文件系统的内容。

## 对 Amazon EFS 客户主密钥执行管理操作

您可以在下文了解如何启用、禁用或删除与您的 Amazon EFS 文件系统关联的 CMK。您还可以了解在执行这些操作时您的文件系统预计出现的行为。

## 禁用、删除或撤销文件系统 CMK 访问权限

您可以禁用或删除您的自定义 CMK，也可以撤销 Amazon EFS 访问您的 CMK 的权限。为 Amazon EFS 禁用和撤销访问您的密钥的权限是不可撤销的操作。在删除 CMK 时，应格外小心。删除 CMK 是不可撤销的操作。

如果您禁用或删除用于挂载的文件系统的 CMK，则满足以下条件：

- 该 CMK 不能用作新的静态加密文件系统的主密钥。



- 在经过一段时间后，使用该 CMK 的现有静态加密文件系统将停止工作。

如果您撤销为 Amazon EFS 授予的任何现有的挂载文件系统的访问权限，该行为与禁用或删除关联的 CMK 相同。换句话说，静态加密的文件系统继续正常工作，但在经过一段时间后停止工作。

要禁止访问具有已禁用、删除或撤销 Amazon EFS 访问权限的 CMK 的挂载静态加密文件系统，请卸载该文件系统并删除 Amazon EFS 挂载目标。

您无法立即删除 AWS KMS 密钥，但可以计划删除密钥。可以删除 CMK 的最早时间是计划删除密钥之后的 7 天。在计划删除密钥时，其行为与禁用相同。您也可以取消计划的密钥删除。有关在 AWS KMS 中删除主密钥的更多信息，请参阅 AWS Key Management Service Developer Guide 中的[删除客户主密钥](#)。

以下过程简要说明了如何禁用 CMK。

### 禁用 CMK

1. 打开 IAM 控制台 (<https://console.aws.amazon.com/iam/home#encryptionKeys>) 的加密密钥部分。
2. 对于区域，请选择相应的 AWS 区域。不要使用导航栏中的 AWS 区域选择器（右上角）。
3. 选中要禁用的每个 CMK 的别名旁边的复选框。

#### Note

您无法禁用以橙色 AWS 图标表示的 AWS 托管 CMK。

4. 要禁用 CMK，请依次选择 Key actions 和 Disable。

以下过程简要说明了如何启用 CMK。

### 启用 CMK

1. 打开 IAM 控制台 (<https://console.aws.amazon.com/iam/home#encryptionKeys>) 的加密密钥部分。
2. 对于区域，请选择相应的 AWS 区域。不要使用导航栏中的 AWS 区域选择器（右上角）。
3. 选中要启用的每个 CMK 的别名旁边的复选框。

#### Note

您无法启用以橙色 AWS 图标表示的 AWS 托管 CMK。

4. 要启用 CMK，请依次选择 Key actions 和 Enable。

## 相关主题

- 有关 Amazon EFS 中的静态加密的数据和元数据的更多信息，请参阅[在 EFS 中加密数据和元数据 \(p. 86\)](#)。
- 有关示例密钥策略，请参阅 [AWS KMS 的 Amazon EFS 密钥策略 \(p. 89\)](#)。
- 有关与加密的文件系统关联的 AWS CloudTrail 日志条目列表，请参阅 [静态加密的文件系统的 Amazon EFS 日志文件条目 \(p. 76\)](#)。
- 有关确定哪些账户和服务有权访问您的 CMK 的更多信息，请参阅 AWS Key Management Service Developer Guide 中的[确定 AWS KMS 客户主密钥的访问权限](#)。

# 挂载文件系统

在下一节中，您可以了解如何使用 Amazon EFS 挂载帮助程序在 Linux 实例上挂载 Amazon EFS 文件系统。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。

在具有 Amazon EFS 挂载帮助程序之前，我们建议您使用标准 Linux NFS 客户端挂载 Amazon EFS 文件系统。有关这些更改的更多信息，请参阅[在没有 EFS 挂载帮助程序的情况下挂载文件系统 \(p. 220\)](#)。

## Note

您必须创建、配置和启动相关的 AWS 资源，然后才可以挂载文件系统。有关详细说明，请参阅[Amazon Elastic File System 入门 \(p. 9\)](#)。

## 主题

- [AMI 和内核版本故障排除 \(p. 59\)](#)
- [安装 amazon-efs-utils 软件包 \(p. 59\)](#)
- [使用 EFS 挂载帮助程序进行挂载 \(p. 59\)](#)
- [自动挂载 Amazon EFS 文件系统 \(p. 61\)](#)
- [其他挂载注意事项 \(p. 63\)](#)

## AMI 和内核版本故障排除

要在从 Amazon EC2 实例中使用 Amazon EFS 时解决与某些 Amazon Machine Image (AMI) 或内核版本相关的问题，请参阅[解决 AMI 和内核问题 \(p. 97\)](#)。

## 安装 amazon-efs-utils 软件包

要在 Amazon EC2 实例上挂载 Amazon EFS 文件系统，我们建议您使用 amazon-efs-utils 软件包中的挂载帮助程序。amazon-efs-utils 软件包是一个开源 Amazon EFS 工具集。有关更多信息，请参阅[在 Amazon Linux 上安装 amazon-efs-utils 软件包 \(p. 35\)](#)。

## 使用 EFS 挂载帮助程序进行挂载

您可以在一些客户端上使用 Amazon EFS 挂载帮助程序挂载 Amazon EFS 文件系统。在以下几节中，您可以找到各种类型的客户端的挂载帮助程序过程。

## 主题

- [使用 EFS 挂载帮助程序在 Amazon EC2 上挂载 \(p. 59\)](#)
- [在本地 Linux 客户端上使用 EFS 挂载帮助程序通过 AWS Direct Connect 挂载 \(p. 60\)](#)

## 使用 EFS 挂载帮助程序在 Amazon EC2 上挂载

您可以使用 Amazon EFS 挂载帮助程序在 Amazon EC2 实例上挂载 Amazon EFS 文件系统。有关挂载帮助程序的更多信息，请参阅[EFS 挂载帮助程序 \(p. 37\)](#)。要使用挂载帮助程序，您需要具有：

- Amazon EFS 文件系统 ID – 在创建 Amazon EFS 文件系统后，您可以从控制台中获取该文件系统的 ID，或以编程方式通过 Amazon EFS API 获取该 ID。该 ID 采用以下格式：`fs-12345678`。
- Amazon EFS 挂载目标 – 您在 VPC 中创建挂载目标。如果您在控制台中创建文件系统，则会同时创建挂载目标。有关更多信息，请参阅[使用 Amazon EFS 控制台创建挂载目标 \(p. 21\)](#)。

- 运行支持的 Linux 发行版的 Amazon EC2 实例 – 支持使用挂载帮助程序挂载文件系统的 Linux 发行版是 Amazon Linux 2、Amazon Linux 2017.09 和更新版本、Red Hat Enterprise Linux ( 和衍生产品, 如 CentOS ) 7 和更新版本以及 Ubuntu 16.04 LTS 和更新版本。
- 安装的 Amazon EFS 挂载帮助程序 – 挂载帮助程序是 amazon-efs-utils 中的一个工具。有关如何安装 amazon-efs-utils 的信息, 请参阅在 [Amazon Linux 上安装 amazon-efs-utils 软件包](#) (p. 35)。

### 使用挂载帮助程序挂载 Amazon EFS 文件系统

1. 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。有关如何执行该操作的更多信息, 请参阅 Amazon EC2 用户指南 ( 适用于 Linux 实例 ) 中的 [使用 SSH 连接到您的 Linux 实例](#)。
2. 运行以下命令以挂载文件系统。

```
sudo mount -t efs fs-12345678:/ /mnt/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-12345678:/ /mnt/efs
```

您也可以选择在 `/etc/fstab` 文件中添加条目以自动进行挂载。在使用 `/etc/fstab` 进行自动挂载时, 您必须添加 `_netdev` 挂载选项。有关更多信息, 请参阅 [将现有 EC2 实例更新为自动挂载](#) (p. 61)。

#### Note

在使用挂载帮助程序进行挂载时, 将自动使用针对 Amazon EFS 优化的以下挂载选项:

- `nfsvers=4.1`
- `rsize=1048576`
- `wsiz=1048576`
- `hard`
- `timeo=600`
- `retrans=2`

要使用 `mount` 命令, 必须满足以下条件:

- 连接的 EC2 实例必须位于 VPC 中, 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 DHCP 选项集。
- 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](#) 中的查看您的 EC2 实例的 DNS 主机名。

#### 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。在该等待时间内, 将在文件系统所在的 AWS 区域中完全传播 DNS 记录。

## 在本地 Linux 客户端上使用 EFS 挂载帮助程序通过 AWS Direct Connect 挂载

在使用 AWS Direct Connect 连接到您的 Amazon VPC 时, 您可以在本地数据中心服务器上挂载 Amazon EFS 文件系统。使用 amazon-efs-utils 挂载 Amazon EFS 文件系统还可以简化使用挂载帮助程序进行挂载的过程, 并允许您启用传输中的数据加密。

要了解如何将 `amazon-efs-utils` 与 AWS Direct Connect 一起使用以将 Amazon EFS 文件系统挂载到本地 Linux 客户端上，请参阅[演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 \(p. 127\)](#)。

## 自动挂载 Amazon EFS 文件系统

在每次重启挂载了 Amazon EFS 文件系统的 Amazon EC2 实例时，您可以使用 `fstab` 通过挂载帮助程序自动挂载该文件系统。有关挂载帮助程序的更多信息，请参阅[EFS 挂载帮助程序 \(p. 37\)](#)。您可以通过两种方法设置自动挂载。您可在首次连接到 EC2 实例后更新该实例中的 `/etc/fstab` 文件，也可以在创建 EC2 实例时配置自动挂载 EFS 文件系统。

### 将现有 EC2 实例更新为自动挂载

要在 Amazon EC2 实例重启时自动重新挂载 Amazon EFS 文件系统目录，您可以使用 `fstab` 文件。`fstab` 文件包含有关文件系统和命令 `mount -a` 的信息，该命令在实例启动期间运行，并挂载 `fstab` 文件列出的文件系统。

#### Note

确保您已创建 Amazon EFS 文件系统，然后才能更新 EC2 实例的 `/etc/fstab` 文件。有关更多信息，请参阅 Amazon EFS 入门练习中的[第 2 步：创建您的 Amazon EFS 文件系统 \(p. 12\)](#)。

更新 EC2 实例中的 `/etc/fstab` 文件

1. 连接到您的 EC2 实例，然后在编辑器中打开 `/etc/fstab` 文件。
2. 将以下行添加到 `/etc/fstab` 文件中。

```
fs-12345678:/mnt/efs efs defaults,_netdev 0 0
```

如果不使用 `amazon-efs-utils` 进行挂载，请参阅[自动挂载 \(p. 223\)](#)。

#### 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 \(p. 99\)](#)。

3. 保存对文件所做的更改。

您的 EC2 实例现已配置为每次重启时都挂载 EFS 文件系统。

#### Note

如果您的 Amazon EC2 实例需要启动 (无论所挂载 Amazon EFS 文件系统的状态如何)，您需要将 `nofail` 选项添加到 `etc/fstab` 文件的文件系统条目中。

您添加到 `/etc/fstab` 文件的代码行将执行以下操作。

字段	描述
<code>fs-12345678:/</code>	您的 Amazon EFS 文件系统的 ID。您可以从控制台中获取该 ID，也可以从 CLI 或 AWS 开发工具包中以编程方式获取该 ID。
<code>/mnt/efs</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> 。

字段	描述
mount options	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： <ul style="list-style-type: none"><li>defaults – 该值指示操作系统使用默认挂载选项，您可以在挂载文件系统后查看 mount 命令输出以列出这些选项。</li><li>_netdev – 该值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。</li><li>您可以将此处的 defaults 替换为 tls 以启用传输中的数据加密。</li></ul>
0	非零值表示应由 dump 备份文件系统。对于 EFS，该值应为 0。
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。

## 将 EFS 文件系统配置为在 EC2 实例启动时自动挂载

您可以借助与 cloud-init 配合使用的脚本将 Amazon EC2 实例配置为在它首次启动时自动挂载 Amazon EFS 文件系统。您可以在 EC2 管理控制台的启动实例向导中添加该脚本。有关如何从控制台启动 EC2 实例的示例，请参阅 [入门 \(p. 9\)](#)。

该脚本会安装 NFS 客户端并在 /etc/fstab 文件中写入条目，该条目将标识挂载目标 DNS 名称以及要挂载 EFS 文件系统的 EC2 实例中的子目录。该脚本可确保在启动 EC2 实例时以及在每次系统重启后挂载该文件。

有关 Amazon Linux 所使用的 cloud-init 自定义版本的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 [cloud-init](#)。

### 将 EC2 实例配置为在启动时自动挂载 EFS 文件系统

- 在 Web 浏览器中打开 Amazon EC2 控制台，然后开始执行启动实例向导。
- 在到达步骤 3：配置实例详细信息时，配置实例详细信息，展开高级部分，然后执行以下操作：
  - 将以下脚本粘贴到用户数据中。您必须为 `fs-12345678` 和 `/mnt/efs` 提供相应的值以更新该脚本：

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-12345678
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults
```

#### 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 \(p. 99\)](#)。

如果您指定的是自定义挂载点路径 (如示例所示), 则可能需要使用 `mkdir -p`, 因为 `-p` 选项可根据需要创建中间父目录。上述示例中的 `- chown` 一行将挂载点目录的所有权从根用户更改为 Amazon Linux 默认的 Linux 系统用户账户, `ec2-user`。您可以使用该命令指定任何用户, 或从脚本中删除该行命令, 以将该目录的所有权保留为根用户。

有关用户数据脚本的更多信息, 请参阅 Amazon EC2 用户指南 (适用于 Linux 实例) 中的[添加用户数据](#)。

### 3. 完成启动实例向导。

#### Note

要验证您的 EC2 实例是否正常工作, 您可以将这些步骤集成到入门练习中。有关更多信息, 请参阅[入门 \(p. 9\)](#)。

您的 EC2 实例现已配置为在启动时挂载 EFS 文件系统。

## 其他挂载注意事项

在 Amazon EC2 实例上挂载 Amazon EFS 文件系统时, 请注意以下其他几项事项:

- 我们建议使用以下默认 Linux 挂载选项值:

```
rsize=1048576
wsize=1048576
hard
timeo=600
retrans=2
noresvport
```

- 如果您必须更改 IO 大小参数 (`rsize` 和 `wsize`), 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。
- 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。该 `timeo` 参数单位为分秒 (0.1 秒), 因此 15 秒等于 150 分秒。
- 建议您使用硬挂载选项。但是, 如果您使用软挂载, 则需要将 `timeo` 参数至少设置为 150 分秒。
- 如果使用 `noresvport` 选项, 在重新建立网络连接时, NFS 客户端将使用新的传输控制协议 (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。
- 避免设置不同于默认值的任何其他挂载选项。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 不支持任何 Kerberos 安全变体。例如, 以下命令将导致挂载失败:

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

- 建议您使用文件系统的 DNS 名称挂载文件系统, 该名称将解析为与 Amazon EC2 实例位于同一可用区中的 Amazon EFS 挂载目标的 IP 地址。如果您使用的挂载目标与 Amazon EC2 实例位于不同的可用区, 则需要为跨可用区发送数据支付标准 Amazon EC2 数据传输费, 并且可能会看到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 `man fstab` 和 `man nfs` 页面。

## 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。您可以通过在 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、AWS 管理控制

台或任何 AWS 开发工具包来卸载 Amazon EFS 文件系统。要卸载连接到运行 Linux 的 Amazon EC2 实例的 Amazon 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。

您可以通过运行 `df` 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情况统计信息，来验证您的 Amazon EFS 文件系统是否已卸载。如果在 `df` 命令输出中没有列出要卸载的 Amazon EFS 文件系统，这意味着已卸载该文件系统。

Example 示例：确定 Amazon 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% /
```



# 监控 Amazon EFS

监控是保持 Amazon EFS 和 AWS 解决方案的可靠性、可用性和性能的重要环节。您应从 AWS 解决方案的所有部分收集监控数据，以便更轻松地调试出现的多点故障。但是，在开始监控 Amazon EFS 之前，您应创建一个可以回答以下问题的监控计划：

- 您的监控目标是什么？
- 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？

接下来，通过在不同时间和不同负载条件下衡量性能，在您的环境中建立为正常的 Amazon EFS 性能建立基准。在监控 Amazon EFS 时，您应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，您可监控网络吞吐量、读写 I/O 和/或元数据操作、客户端连接以及文件系统的突增积分余额。如果性能低于您设定的基准，则您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。

## 监控工具

AWS 为您提供了各种可用于监控 Amazon EFS 的工具。您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

您可以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：

- Amazon CloudWatch 警报 – 按您指定的时间段观察单个指标，并根据相对于给定阈值的指标值在若干时间段内执行一项或多项操作。该操作是向 Amazon Simple Notification Service (Amazon SNS) 主题或 Amazon EC2 Auto Scaling 策略发送通知。CloudWatch 警报将不会调用操作，因为这些操作处于特定状态；该状态必须改变并在指定数量的时间段内一直保持。有关更多信息，请参阅 [使用 Amazon CloudWatch 进行监控 \(p. 66\)](#)。
- 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 Amazon CloudWatch 用户指南 中的 [监视日志文件](#)。
- Amazon CloudWatch Events – 匹配事件并将事件传送到一个或多个目标函数或流来进行更改、捕获状态信息和采取纠正措施。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的 [什么是 Amazon CloudWatch 事件](#)。
- AWS CloudTrail 日志监控 – 在账户间共享日志文件，通过将 CloudTrail 日志文件发送到 CloudWatch Logs 对它们进行实时监控，在 Java 中编写日志处理应用程序，以及验证您的日志文件在被 CloudTrail 交付后未发生更改。有关更多信息，请参阅 AWS CloudTrail User Guide 中的 [使用 CloudTrail 日志文件](#)。



## 手动监控工具

监控 Amazon EFS 的另一个重要环节是手动监控 Amazon CloudWatch 警报未涵盖的那些项。Amazon EFS、CloudWatch 和其他 AWS 控制台仪表板均提供 AWS 环境状态的概览视图。建议您还要查看 file system 上的日志文件。

- 您可以从 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：
  - 当前警报和状态
  - 警报和资源的图表
  - 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
- 搜索并浏览您所有的 AWS 资源指标
- 创建和编辑警报以接收有关问题的通知

## 使用 Amazon CloudWatch 进行监控

您可以使用 Amazon CloudWatch 监控文件系统，此工具可从 Amazon EFS 收集原始数据，并将数据处理为易读的近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。默认情况下，Amazon EFS 指标数据以 1 分钟为间隔自动发送到 CloudWatch。有关 CloudWatch 的更多信息，请参阅[什么是 Amazon CloudWatch](#)、[Amazon CloudWatch Events](#) 和 [Amazon CloudWatch Logs ?](#) (位于 Amazon CloudWatch 用户指南 中)。

## Amazon EFS 的 Amazon CloudWatch 指标

AWS/EFS 命名空间包括以下指标。

指标	描述
BurstCreditBalance	<p>文件系统具有的突增额度。</p> <p>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。有关更多信息，请参阅 <a href="#">Amazon EFS 中的吞吐量扩展</a>。</p> <p>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。</p> <p>单位：字节</p> <p>有效统计数据：Minimum、Maximum、Average</p>
ClientConnections	<p>文件系统的客户端连接数量。使用标准客户端时，每个装载的 Amazon EC2 实例使用一个连接。</p>

指标	描述
	<p><b>Note</b></p> <p>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。</p> <p>单位：客户端连接数量</p> <p>有效统计数据：Sum</p>
DataReadIOBytes	<p>每个文件系统读取操作的字节数。</p> <p>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。</p> <p>单位：</p> <ul style="list-style-type: none"><li>• 对于 Minimum、Maximum、Average 和 Sum，单位为字节。</li><li>• SampleCount 的数量。</li></ul> <p>有效统计数据：Minimum、Maximum、Average、Sum、SampleCount</p>
DataWriteIOBytes	<p>每个文件写入操作的字节数。</p> <p>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。</p> <p>单位：</p> <ul style="list-style-type: none"><li>• Minimum、Maximum、Average 和 Sum 统计数据的单位是字节。</li><li>• SampleCount 的数量。</li></ul> <p>有效统计数据：Minimum、Maximum、Average、Sum、SampleCount</p>
MetadataIOBytes	<p>每个元数据操作的字节数。</p> <p>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。</p> <p>单位：</p> <ul style="list-style-type: none"><li>• Minimum、Maximum、Average 和 Sum 统计数据的单位是字节。</li><li>• SampleCount 的数量。</li></ul> <p>有效统计数据：Minimum、Maximum、Average、Sum、SampleCount</p>

指标	描述
PercentIOLimit	<p>显示文件系统接近通用性能模式的 I/O 限制的情况。如果此指标经常达到 100%，请考虑将您的应用程序移到使用最高 I/O 性能模式的文件系统。</p> <p><b>Note</b></p> <p>只为使用通用性能模式的文件系统提交此指标。</p> <p>单位:</p> <ul style="list-style-type: none"><li>• 百分比</li></ul>
PermittedThroughput	<p>文件系统允许的最大吞吐量。对于预配置吞吐量模式下的文件系统，此值与预配置吞吐量相同。对于突发吞吐量模式下的文件系统，此值是文件系统大小与 BurstCreditBalance 的函数。有关更多信息，请参阅 <a href="#">Amazon EFS 性能</a>。</p> <p>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。</p> <p>单位：字节/秒</p> <p>有效统计数据：Minimum、Maximum、Average</p>
TotalIOBytes	<p>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。</p> <p>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。</p> <p><b>Note</b></p> <p>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。</p> <p>单位:</p> <ul style="list-style-type: none"><li>• Minimum、Maximum、Average 和 Sum 统计数据的单位是字节。</li><li>• SampleCount 的数量。</li></ul> <p>有效统计数据：Minimum、Maximum、Average、Sum、SampleCount</p>

## 在 CloudWatch 中报告的字节数

与 Amazon S3 和 Amazon EBS 一样，Amazon EFS CloudWatch 指标是作为原始字节数 报告的。字节数不会舍入到十进制或二进制单位倍数。在使用从指标中获取的数据计算突增速率时，请记住这一点。有关突增的更多信息，请参阅[随突发模式扩展的吞吐量 \(p. 80\)](#)。

## Amazon EFS 维度

Amazon EFS 指标使用 EFS 命名空间，并且为单个维度 FileSystemId 提供指标。可以在 Amazon EFS 管理控制台中找到文件系统 ID，该 ID 采用 fs-xxxxxxx 的格式。

## 如何使用 Amazon EFS 指标？

您可以通过多种方式分析 Amazon EFS 报告指标提供的信息。以下列表显示了这些指标的一些常见用途。下面列出的是能够带您入门的启发式问题，但并不全面。

如何？	相关指标
如何确定我的吞吐量？	您可以监控 <code>TotalIOBytes</code> 指标的每日 Sum 统计数据以查看您的吞吐量。
如何跟踪连接到文件系统的 Amazon EC2 实例数量？	您可以监控 <code>ClientConnections</code> 指标的 Sum 统计数据。要计算超过一分钟的时段的 <code>ClientConnections</code> 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 <code>BurstCreditBalance</code> 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 <a href="#">随突增模式扩展的吞吐量 (p. 80)</a> 。

## 使用 Amazon CloudWatch 监控 EFS 文件同步

您可以使用 Amazon CloudWatch 监控 EFS 文件同步，该工具从 Amazon EFS 中收集原始数据，并将其处理为近乎实时的可读指标。这些统计数据的记录期限为 15 个月，以便您可以访问历史信息，并更好地了解 EFS 文件同步的运行状况。默认情况下，每隔 5 分钟自动将 EFS 文件同步指标数据发送到 CloudWatch。有关 CloudWatch 的更多信息，请参阅[什么是 Amazon CloudWatch](#)、[Amazon CloudWatch Events](#) 和 [Amazon CloudWatch Logs](#)？(位于 Amazon CloudWatch 用户指南中)。

AWS/FileSync 命名空间包含以下指标。

指标	描述
<code>FilesTransferred</code>	从源文件系统传输到 Amazon EFS 文件系统的文件数。如果所需的同步的任何方面增加该指标，则将文件视为要进行传输。在这种情况下，将增加该指标。但是，如果仅更改了元数据，则不会传输任何实际数据。  单位：计数
<code>PhysicalBytesTransferred</code>	在同步代理将数据从源文件系统读取到 Amazon EFS 文件系统时通过网络传输的总字节数。  单位：字节
<code>LogicalBytesTransferred</code>	传输到 Amazon EFS 文件系统的文件或目录的总大小。在该指标中不包含元数据。  单位：字节

## Amazon EFS 文件同步维度

EFS 文件同步指标使用 AWS/FileSync 命名空间，并为以下维度提供指标：

- `HostId` – 主机服务器的唯一 ID。
- `HostName` – 主机服务器的名称或域。
- `SyncSetId` – 同步集的 ID。其形式为 `set-12345678912345678`。

## 访问 CloudWatch 指标

您可以通过很多方法查看 CloudWatch 的 Amazon EFS 指标。您可以通过 CloudWatch 控制台查看它们，也可以使用 CloudWatch CLI 或 CloudWatch API 访问它们。以下步骤向您介绍了如何使用这些不同工具访问指标。

### 使用 CloudWatch 控制台查看指标

1. 打开 CloudWatch 控制台 (<https://console.aws.amazon.com/cloudwatch/>)。
2. 在导航窗格中，选择 Metrics。
3. 选择 EFS 命名空间。
4. (可选) 要查看某个指标，请在搜索字段中键入其名称。
5. (可选) 要按维度筛选，请选择 FileSystemId。

### 从 AWS CLI 访问指标

- 带 `--namespace "AWS/EFS"` 命名空间使用 `list-metrics` 命令。有关更多信息，请参阅 [AWS CLI Command Reference](#)。

### 从 CloudWatch API 访问指标

- 调用 `GetMetricStatistics`。有关详细信息，请参见 [Amazon CloudWatch API Reference](#)。

## 创建 CloudWatch 警报以监控 Amazon EFS

您可以创建 CloudWatch 警报，以在警报改变状态时发送 Amazon SNS 消息。警报会每隔一段时间 (由您指定) 监控一个指标，并根据相对于给定阈值的指标值每隔若干个时间段执行一项或多项操作。操作是向 Amazon SNS 主题或 Auto Scaling 策略发送的通知。

警报只会调用操作进行持续的状态变更。CloudWatch 警报不会仅仅因为处于特定状态而调用操作；该状态必须已发生变化，并在指定数量的时间段内保持该状态。

Amazon EFS 的 CloudWatch 警报的一个重要用途是，为您的文件系统实施静态加密。您可以在创建 Amazon EFS 文件系统时启用静态加密。要为 Amazon EFS 文件系统实施静态数据加密策略，您可以使用 Amazon CloudWatch 和 AWS CloudTrail 检测创建的文件系统并验证是否启用了静态加密。有关更多信息，请参阅 [演练 6：在 Amazon EFS 文件系统上实施静态加密 \(p. 132\)](#)。

### 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 CloudWatch 控制台设置警报

1. 登录 AWS 管理控制台并通过以下网址打开 CloudWatch 控制台 <https://console.aws.amazon.com/cloudwatch/>。
2. 选择 Create Alarm。Create Alarm Wizard (创建警报向导) 将随即启动。
3. 选择 EFS Metrics 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 Next。
4. 填写指标的 Name、Description、Whenever 值。
5. 如果您希望 CloudWatch 在达到警报状态时向您发送一封电子邮件，请在 Whenever this alarm: (每当此警报:) 字段中，选择 State is ALARM (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 Create topic (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 Create Alarm。

#### 使用 AWS CLI 设置警报

- 调用 `put-metric-alarm`。有关更多信息，请参阅 [AWS CLI Command Reference](#)。

#### 使用 CloudWatch API 设置警报

- 调用 `PutMetricAlarm`。有关详细信息，请参见 [Amazon CloudWatch API Reference](#)

## 将指标数学与 Amazon EFS 一起使用

通过使用指标数学，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。您可以在 CloudWatch 控制台中直观显示生成的时间序列，并将其添加到控制面板中。例如，您可以使用 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 [使用指标数学](#)。

您可以在下文中找到一些有用的 Amazon EFS 指标数学表达式。

#### 主题

- [指标数学：吞吐量 \( MiB/秒 \) \(p. 71\)](#)
- [指标数学：百分比吞吐量 \(p. 72\)](#)
- [指标数学：吞吐量 IOPS \(p. 72\)](#)
- [指标数学：IOPS 百分比 \(p. 73\)](#)
- [指标数学：平均 I/O 大小 \(KiB\) \(p. 73\)](#)
- [通过 Amazon EFS 的 AWS CloudFormation 模板使用指标数学 \(p. 74\)](#)

## 指标数学：吞吐量 ( MiB/秒 )

要计算某个时间段的平均吞吐量 ( MiB/秒 )，请先选择总计统计数据 ( 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：(TotalIOBytes 总和 ÷ 1048576 (以转换为 MiB)) ÷ 该时间段的秒数

然后，您的 CloudWatch 指标信息如下所示。

ID	可用的指标	统计数据	Period
m1	<ul style="list-style-type: none"><li>• DataReadIOBytes</li><li>• DataWriteIOBytes</li><li>• MetadataIOBytes</li><li>• TotalIOBytes</li></ul>	sum	1 minute

您的指标数学 ID 和表达式如下所示。

ID	表达式
e1	$(m1/1048576)/PERIOD(m1)$

## 指标数学：百分比吞吐量

要计算某个时间段的各种 I/O 类型 ( `DataReadIOBytes`、`DataWriteIOBytes` 或 `MetadataIOBytes` ) 的百分比吞吐量，请先将相应的总计统计数据乘以 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： $(\text{DataReadIOBytes 总和} \times 100 \text{ (以转换为百分比)}) \div \text{TotalIOBytes 总和}$

然后，您的 CloudWatch 指标信息如下所示。

ID	可用的一个或多个指标	统计数据	Period
m1	<ul style="list-style-type: none"><li><code>TotalIOBytes</code></li></ul>	sum	1 minute
m2	<ul style="list-style-type: none"><li><code>DataReadIOBytes</code></li><li><code>DataWriteIOBytes</code></li><li><code>MetadataIOBytes</code></li></ul>	sum	1 minute

您的指标数学 ID 和表达式如下所示。

ID	表达式
e1	$(m2*100)/m1$

## 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： $\text{DataWriteIOBytes 样本数} \div \text{该时间段的秒数}$

然后，您的 CloudWatch 指标信息如下所示。

ID	可用的指标	统计数据	Period
m1	<ul style="list-style-type: none"><li><code>DataReadIOBytes</code></li><li><code>DataWriteIOBytes</code></li><li><code>MetadataIOBytes</code></li><li><code>TotalIOBytes</code></li></ul>	样本数	1 minute

您的指标数学 ID 和表达式如下所示。



ID	表达式
e1	m1/PERIOD(m1)

## 指标数学：IOPS 百分比

要计算某个时间段的各种 I/O 类型 ( `DataReadIOBytes`、`DataWriteIOBytes` 或 `MetadataIOBytes` ) 的每秒 IOPS 百分比，请先将相应的样本数统计数据乘以 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： $(\text{MetadataIOBytes 样本数} \times 100 \text{ (以转换为百分比)}) \div \text{TotalIOBytes 样本数}$

然后，您的 CloudWatch 指标信息如下所示。

ID	可用的指标	统计数据	Period
m1	<ul style="list-style-type: none"><li><code>TotalIOBytes</code></li></ul>	样本数	1 minute
m2	<ul style="list-style-type: none"><li><code>DataReadIOBytes</code></li><li><code>DataWriteIOBytes</code></li><li><code>MetadataIOBytes</code></li></ul>	样本数	1 minute

您的指标数学 ID 和表达式如下所示。

ID	表达式
e1	$(m2 \times 100) / m1$

## 指标数学：平均 I/O 大小 (KiB)

要计算某个时间段的平均 I/O 大小 (KiB)，请将 `DataReadIOBytes`、`DataWriteIOBytes` 或 `MetadataIOBytes` 指标的相应总计统计数据除以该指标的相同样本数统计数据。

假设您的示例逻辑是： $(\text{DataReadIOBytes 总和} \div 1024 \text{ (以转换为 KiB)}) \div \text{DataReadIOBytes 样本数}$

然后，您的 CloudWatch 指标信息如下所示。

ID	可用的指标	统计数据	Period
m1	<ul style="list-style-type: none"><li><code>DataReadIOBytes</code></li><li><code>DataWriteIOBytes</code></li><li><code>MetadataIOBytes</code></li></ul>	sum	1 minute
m2	<ul style="list-style-type: none"><li><code>DataReadIOBytes</code></li><li><code>DataWriteIOBytes</code></li><li><code>MetadataIOBytes</code></li></ul>	样本数	1 minute

您的指标数学 ID 和表达式如下所示。



ID	表达式
e1	(m1/1024)/m2

## 通过 Amazon EFS 的 AWS CloudFormation 模板使用指标数学

您还可以通过 AWS CloudFormation 模板创建指标数学表达式。您可以从 GitHub 上的 [Amazon EFS 教程](#) 中下载一个此类模板，并自定义以进行使用。有关使用 AWS CloudFormation 模板的更多信息，请参阅 AWS CloudFormation 用户指南 中的 [使用 AWS CloudFormation 模板](#)。

## 使用 AWS CloudTrail 记录 Amazon EFS API 调用

Amazon EFS 与 AWS CloudTrail 集成，后者是在 Amazon EFS 中记录用户、角色或 AWS 服务所执行操作的服务。CloudTrail 将对 Amazon EFS 的所有 API 调用作为事件捕获，包括来自 Amazon EFS 控制台（您的 Git 客户端）的调用、来自代码对 Amazon EFS API 操作的调用。

如果您创建了跟踪，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 Amazon EFS 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。通过使用 CloudTrail 收集的信息，您可以确定向 Amazon EFS 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，请参阅 [AWS CloudTrail User Guide](#)。

## CloudTrail 中的 Amazon EFS 信息

在您创建 AWS 账户时，将针对该账户启用 CloudTrail。Amazon EFS 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history（事件历史记录）中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅 [使用 CloudTrail 事件历史记录查看事件](#)。

要持续记录 AWS 账户中的事件（包括 Amazon EFS 的事件），请创建跟踪。通过跟踪，CloudTrail 可将日志文件传送至 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取操作。有关更多信息，请参阅 [AWS CloudTrail User Guide](#) 中的以下主题：

- [创建跟踪概述](#)
- [CloudTrail 支持的服务和集成](#)
- [为 CloudTrail 配置 Amazon SNS 通知](#)
- [接收多个区域中的 CloudTrail 日志文件和从多个账户中接收 CloudTrail 日志文件](#)。

CloudTrail 会记录所有 Amazon EFS [API 调用](#) (p. 154)。例如，对 CreateFilesystem、CreateMountTarget 和 CreateTags 操作的调用将在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员的信息。身份信息帮助您确定以下内容：

- 请求是使用根用户凭证还是 AWS Identity and Access Management (IAM) 用户凭证发出的。
- 请求是使用角色还是联合身份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 [AWS CloudTrail User Guide](#) 中的 [CloudTrail userIdentity 元素](#)。

## 了解 Amazon EFS 日志文件条目

跟踪 是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件 表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

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删除文件系统标签时的 `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"
}
```

## 静态加密的文件系统的 Amazon EFS 日志文件条目

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 [在 EFS 中加密数据和元数据 \(p. 86\)](#)。

如果使用静态加密的文件系统，Amazon EFS 代表您进行的调用在 AWS CloudTrail 日志中显示为来自 AWS 拥有的账户。如果在 CloudTrail 日志中看到以下账户 ID 之一（具体取决于在其中创建文件系统的 AWS 区域），则该 ID 是 Amazon EFS 服务拥有的 ID。

AWS 区域	账户 ID
美国东部（俄亥俄州）	771736226457
美国东部（弗吉尼亚北部）	055650462987
美国西部（加利福尼亚北部）	208867197265
美国西部（俄勒冈）	736298361104
亚太区域（首尔）	518632624599
亚太区域（东京）	620757817088
欧洲（法兰克福）	992038834663
欧洲（爱尔兰）	805538244694
亚太区域（悉尼）	288718191711

## 静态加密的 Amazon EFS 加密上下文

在发出 AWS KMS API 请求以生成数据密钥并解密 Amazon EFS 数据时，Amazon EFS 将发送 [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"
```

# Amazon EFS 性能

本章概述 Amazon EFS 性能，讨论可用的性能模式和吞吐量模式，并概述一些有用的性能提示。

## 性能概述

Amazon EFS 文件系统分布在数量不受约束的存储服务器上，使文件系统能够弹性扩展到 PB 级规模，并允许从 Amazon EC2 实例大规模并行访问您的数据。Amazon EFS 的分布式设计避免了传统文件服务器固有的瓶颈和限制。

这种分布式数据存储设计意味着多线程应用程序和同时从多个 Amazon EC2 实例访问数据的应用程序会带来巨大的聚合吞吐量和 IOPS。大数据和分析工作负载、媒体处理工作流、内容管理和 web 服务等都属于这类应用程序。

此外，Amazon EFS 数据分布在多个可用区(AZ)中，从而可提供高度持久性和可用性。下表对 Amazon 文件和数据块云存储服务的高性能和存储特性进行了比较。

性能比较，Amazon EFS 和 Amazon EBS

	Amazon EFS	Amazon EBS 预配置 IOPS
每次操作的延迟	低且一致的延迟。	最低且一致的延迟。
吞吐量规模	每秒 10+ GB。	每秒最多 2 GB。

存储特性比较，Amazon EFS 和 Amazon EBS

	Amazon EFS	Amazon EBS 预配置 IOPS
可用性与持久性	数据冗余存储在多个可用区中。	数据冗余存储在一个可用区中。
访问	多个可用区的多达数千个 Amazon EC2 实例可以同时连接到一个文件系统。	一个可用区的一个 Amazon EC2 实例可以连接到一个文件系统。
使用案例	大数据和分析、媒体处理工作流、内容管理、Web 服务和主目录。	引导卷、事务型数据库和 NoSQL 数据库、数据仓库和 ETL。

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。由于这种每次操作的延迟，总吞吐量通常会随着平均 I/O 大小增加而增加，因为开销在大量数据之间分摊。Amazon EFS 支持高度并行化的工作负载 (例如，从多个线程和多个 Amazon EC2 实例使用并行操作)，从而实现巨大的聚合吞吐量和每秒操作数。

## Amazon EFS 使用案例

Amazon EFS 旨在满足以下使用案例的性能需求。

### 大数据与分析

Amazon EFS 提供大数据应用程序所需的规模 and 性能，这些应用程序需要计算节点具有高吞吐量以及“写入后读取一致性”和低延迟文件操作。

## 媒体处理 workflows

媒体 workflows (如视频编辑、演播室制作、广播处理、声音设计和渲染等) 通常依赖于共享存储来操作大型文件。具有高吞吐量和共享文件访问的强数据一致性模型可以缩短执行这些作业所需的时间，并将多个本地文件存储库整合到一个位置以供所有用户使用。

## 内容管理和 Web 服务

Amazon EFS 为内容管理系统提供持久的高吞吐量文件系统，这些内容管理系统为各种应用 (如网站、在线出版物和存档) 存储和提供信息。

## 主目录

Amazon EFS 可以为拥有众多需要访问和共享公共数据集的组织提供存储。管理员可以使用 Amazon EFS 创建一个可供整个组织的人员访问的文件系统，并在文件或目录级别为用户和组建立权限。

## 将文件系统同步到 Amazon EFS

Amazon EFS 文件同步提供高效、高性能的并行数据同步，可以容忍不可靠和高延迟的网络。通过使用该数据同步，您可以轻松高效地将文件从现有的文件系统同步到 Amazon EFS。有关更多信息，请参阅 [Amazon EFS 文件同步 \(p. 28\)](#)。

## 性能模式

为了支持各种云存储工作负载，Amazon EFS 提供了两种性能模式。您应在创建文件系统时选择其性能模式。

两种性能模式没有额外成本，因此，无论您选择哪种性能模式，您的 Amazon EFS 文件系统的计量和计费方式都是一样的。有关文件系统限制的信息，请参阅 [Amazon EFS 文件系统的限制 \(p. 92\)](#)。

### Note

创建 Amazon EFS 文件系统后，其性能模式将无法再更改。

## 通用性能模式

我们建议对于绝大多数 Amazon EFS 文件系统采用通用性能模式。通用性能模式非常适合对延迟敏感的使用案例，如 Web 服务环境、内容管理系统、主目录和一般文件服务。如果您在创建文件系统时未选择性能模式，Amazon EFS 默认选择通用模式。

## 最大 I/O 性能模式

最大 I/O 模式下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，但代价是稍高的文件操作延迟。诸如大数据分析、媒体处理和基因组分析等高度并行化的应用程序和工作负载可以受益于这种模式。

## 使用合适的性能模式

具体应该使用哪种性能模式，我们的建议如下：

1. 使用默认通用性能模式 [创建新文件系统 \(p. 12\)](#)。
2. 运行您的应用程序 (或类似于您的应用程序的使用案例) 一段时间，测试它的性能。
3. 在性能测试期间监视 Amazon EFS 的 [PercentIOLimit \(p. 66\)](#) Amazon CloudWatch 指标。有关访问该指标以及其他指标的更多信息，请参阅 [Amazon CloudWatch 指标 \(p. 65\)](#)。

如果在测试期间的大部分时间返回的 `PercentIOLimit` 百分比达到或接近 100%，您的应用程序应使用最大 I/O 性能模式。否则，应使用默认的通用模式。

## 吞吐量模式

有两种吞吐量模式可供您的文件系统选择：突增吞吐量和预配置吞吐量。使用突增吞吐量模式，Amazon EFS 上的吞吐量随着文件系统的增长而扩展。使用预置吞吐量模式，您可以即时预置与存储的数据量无关的文件系统的吞吐量 (MiB/s)。

### Note

只要自上次降低以来超过 24 小时，您就可以在预配置吞吐量模式下降低文件系统吞吐量。此外，只要自上次吞吐量模式更改以来已超过 24 小时，您就可以在预配置吞吐量模式和默认突增吞吐量模式之间进行更改。

## 随突增模式扩展的吞吐量

使用突增吞吐量模式，Amazon EFS 上的吞吐量随着文件系统的增长而扩展。基于文件的工作负载通常会猛增，短时间内吞吐量较高，其余时间吞吐量较低。因此，Amazon EFS 被设计为可在一段时间内突增到高吞吐量。

所有文件系统，不管其大小如何，都能突增到 100 MiB/s 的吞吐量。那些超过 1 TiB 的大文件系统可以突增到每 TiB 文件系统存储数据 100 MiB/s。例如，一个 10 TiB 的文件系统可以突增到 1,000 MiB/s 吞吐量 (10 TiB x 100 MiB/s/TiB)。文件系统可能突增的时间部分由其大小决定。突增模型的设计使典型的文件系统工作负载几乎在任何需要的时候都可以突增。

Amazon EFS 使用积分系统来判断文件系统何时可以突增。随着时间推移，每个文件系统都以一定基准速率 (取决于文件系统大小) 获得积分，并在读写数据时使用积分。基准速率为每 TiB 存储 50 MiB/s (相当于每 GiB 存储 50 KiB/s)。

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。文件系统可以以其基准速率持续推动吞吐量，每当文件系统不活动或吞吐量低于其基准速率时，它就会累计突增积分。

例如，如果 100 GiB 文件系统在 95% 的时间内处于不活动状态，则可以在 5% 的时间内突增 (以 100 MiB/s 速率)。在一个 24 小时周期内，文件系统获得相当于 432000 MiB 的积分，可用于以 100 MiB/s 速率突增 72 分钟。

如果大于 1 TiB 的文件系统在 50% 的时间内处于不活动状态，则始终可以在其余 50% 的时间内突增。

下表提供了突增行为的示例。

文件系统大小	聚合读取/写入吞吐量
一个 100 GiB 文件系统可以...	<ul style="list-style-type: none"><li>每天以 100 MiB/s 速率突增长达 72 分钟，或者</li><li>持续维持在高达 5 MiB/s 的速率</li></ul>
一个 1 TiB 文件系统可以...	<ul style="list-style-type: none"><li>每天以 100 MiB/s 速率突增 12 小时，或者</li><li>持续维持在高达 50 MiB/s 的速率</li></ul>
一个 10 TiB 文件系统可以...	<ul style="list-style-type: none"><li>每天以 1 GiB/s 的速率突增 12 小时，或者</li><li>持续维持在高达 500 MiB/s 的速率</li></ul>
通常，一个更大的文件系统可以...	<ul style="list-style-type: none"><li>每天以每 TiB 存储 100MiB/s 的速率突增 12 小时，或者</li><li>持续维持在每 TiB 存储 50 MiB/s 的速率</li></ul>



## Note

计算基准速率所使用的最小文件系统大小为 1 GiB，因此，所有文件系统至少具有 50 KiB/s 基准速率。

确定基准速率和突增速率时所使用的文件系统大小与通过 `DescribeFileSystems` 操作得到的计量大小相同。

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。这种方法意味着文件系统可以累积足够的积分来持续突增长达 12 小时。

下表提供了不同大小文件系统更详细的突增行为示例。

文件系统大小 (GiB)	基准聚合吞吐量 (MiB/s)	突增聚合吞吐量 (MiB/s)	最大突增持续时间 (分钟/天)	文件系统突增时间百分比 (每天)
10	0.5	100	7.2	0.5%
256	12.5	100	180	12.5%
512	25.0	100	360	25.0%
1024	50.0	100	720	50.0%
1536	75.0	150	720	50.0%
2048	100.0	200	720	50.0%
3072	150.0	300	720	50.0%
4096	200.0	400	720	50.0%

## Note

如前所述，新文件系统最初具有 2.1 TB 突增积分余额。利用这一初始余额，您可以在不消耗从存储中获得的任何积分的情况下，以 100 MB/s 速率突增 6.12 小时。这个起始公式计算为  $2.1 \times 1024 \times (1024/100/3600)$ ，得到 6.116 小时，四舍五入为 6.12。

## 管理突增积分

当文件系统具有正突增积分余额时，就可以突增。您可以通过查看 Amazon EFS 的 `BurstCreditBalance` Amazon CloudWatch 指标了解文件系统的突增积分余额。有关访问该指标以及其他指标的更多信息，请参阅[监控 Amazon EFS \(p. 65\)](#)。

文件系统的突增能力 (时间长短和突增速率) 与其大小直接相关。越大的文件系统越能够以更大的速率突增越长的时间。在某些情况下，应用程序可能需要更多突增 (即，您可能会发现您的文件系统已耗尽了突增积分)。在这些情况下，您应增加文件系统大小，或者切换到预配置吞吐量模式。

使用您的历史吞吐量模式来计算维持期望的活动水平所需的文件大小。下面概括了具体步骤：

计算维持期望的活动水平所需的文件系统大小

1. 通过查看您的历史使用情况确定您的吞吐量需求。从 [Amazon CloudWatch 控制台](#) 中，将 `TotalIOBytes` 指标的 `sum` 统计与过去 14 天里每天的聚合进行核对。找出具有最大 `TotalIOBytes` 值的那一天。
2. 将该值除以 24 小时、60 分钟、60 秒和 1024 字节，得到您的应用程序在那一天所需的平均 KiB/s 值。
3. 通过将平均吞吐量值 (KiB/s) 除以 EFS 提供的基准吞吐量值 (50 KiB/s/GiB)，来计算维持此平均吞吐量所需的文件系统大小 (GiB)。



## 通过预配置模式指定吞吐量

预配置吞吐量模式适用于具有高吞吐量到存储 (MiB/s/TiB) 比率的应用程序，或具有比突增吞吐量模式允许的比率大的要求的应用程序。例如，假设您正在将 Amazon EFS 用于发工具、Web 服务或内容管理应用程序，而文件系统中的数据量相对于吞吐量需求来说是较低的。您的文件系统现在可以获得应用程序所需的高吞吐量，而无需填充文件系统。

使用预配置吞吐量模式会产生额外费用。使用预配置吞吐量模式，您需要为使用的存储和独立预置的吞吐量付费。有关更多信息，请参阅 <https://aws.amazon.com/efs/pricing>。

无论您选择何种吞吐量模式，吞吐量限制都保持不变。有关这些限制的更多信息，请参阅[您可以提高的 Amazon EFS 限制 \(p. 90\)](#)。

如果文件系统处于预配置吞吐量模式，您可以根据需要随时增加文件系统的预配置吞吐量。只要自上次降低以来超过 24 小时，您就可以在预配置吞吐量模式下降低文件系统吞吐量。此外，只要自上次吞吐量模式更改以来已超过 24 小时，您就可以在预配置吞吐量模式和默认突增吞吐量模式之间进行更改。

如果文件系统的计量大小提供的基准速率高于您预置的吞吐量，则文件系统将遵循默认 & EFS; 突增吞吐量模型。在突增吞吐量模式下，您不会在文件系统的权限下被收取预配置吞吐量费用。有关更多信息，请参阅[随突增模式扩展的吞吐量 \(p. 80\)](#)。

## 使用合适的吞吐量模式

默认情况下，我们建议您以突增吞吐量模式运行应用程序。如果遇到性能问题，请检查 BurstCreditBalance CloudWatch 指标。如果 BurstCreditBalance 指标的值为零或稳步下降，则预配置吞吐量适合于您的应用程序。

### Note

如前所述，新文件系统最初具有 2.1 TB 突增积分余额。利用这一初始余额，您可以在不消耗从存储中获得的任何积分的情况下，以 100 MB/s 速率突增 6.12 小时。这个起始公式计算为  $2.1 \times 1024 \times (1024/100/3600)$ ，得到 6.116 小时，四舍五入为 6.12。

## 本地性能注意事项

无论是从本地服务器还是从 Amazon EC2 实例访问，Amazon EFS 文件系统的突增吞吐量模型 Amazon EFS 都保持不变。然而，当从本地服务器访问 Amazon EFS 文件数据时，最大吞吐量还受限于 AWS Direct Connect 连接的带宽。

由于长距离数据传输会造成传播延迟，因此，本地数据中心和您的 Amazon VPC 之间的 AWS Direct Connect 连接的网络延迟可能会达到数十毫秒。如果您的文件操作按顺序执行，则 AWS Direct Connect 连接延迟将直接影响您的读取和写入吞吐量。实际上，您在一段时间内可以读取或写入的数据量受完成每次读写操作所需时间的束缚。若要最大限度地提高吞吐量，需并行处理文件操作，以便 Amazon EFS 可以同时处理多个读写操作。[GNU parallel](#) 等标准工具可以并行复制文件数据。

## 针对高可用性设计

为确保本地数据中心和 Amazon VPC 之间的连续可用性，我们建议您配置两个 AWS Direct Connect 连接。有关更多信息，请参阅 AWS Direct Connect 用户指南 中的[步骤 4：使用 AWS Direct Connect 配置冗余连接](#)。

为确保您的应用程序和 Amazon EFS 之间的连续可用性，我们建议将您的应用程序设计为能够从潜在的连接中断中恢复。一般而言，连接到 Amazon EFS 文件系统的本地应用程序存在两种情形：高度可用和非高度可用。

首先，您的应用程序具有高可用性 (HA)，并在其 HA 群集中使用多个本地服务器。在这种情况下，请确保 HA 群集中的每个本地服务器都连接到您的 Amazon VPC 中的不同可用区 (AZ) 中的安装目标。如果您的本地服务器由于挂载目标所在的可用区变得不可用而无法访问挂载目标，则您的应用程序应故障转移到有可用挂载目标的服务器。

第二，您的应用程序不是高度可用的，而且您的本地服务器由于挂载目标所在的可用区变得不可用而无法访问挂载目标。在这种情况下，您的应用程序应实现重新启动逻辑并连接到不同 AZ 中的安装目标。

## 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较高的可用性、持久性和可扩展性级别。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。由于这种每次操作的延迟，总吞吐量通常会随着平均 I/O 大小增加而增加，因为开销在大量数据之间分摊。
- 同时连接 – Amazon EFS 文件系统可以同时挂载到多达数千个 Amazon EC2 实例上。如果可以在更多实例中并行执行应用程序，您可以在文件系统上提高这些实例的总吞吐量级别。
- 请求模型 – 通过启用对文件系统的异步写入，待处理的写入操作先在 Amazon EC2 实例上缓冲，然后才异步写入 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过缓存选项 (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。

- NFS 客户端挂载设置 – 确认您使用的是[挂载文件系统 \(p. 59\)](#)和[其他挂载注意事项 \(p. 63\)](#)中推荐的挂载选项。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) 和 NFSv4.0 协议。NFSv4.1 提供更好的性能。

### Note

在挂载文件系统时，可能需要将 NFS 客户端的读取和写入缓冲区的大小提高到 1 MB。

- Amazon EC2 实例 – 执行大量读取和写入操作的应用程序可能比不执行这些操作的应用程序需要更多的内存或计算容量。在启动 Amazon EC2 实例时，应选择具有您的应用程序需要的这些资源量的实例类型。Amazon EFS 文件系统的性能特征不依赖于使用 EBS 优化的实例。
- 加密 – Amazon EFS 支持两种形式的加密：传输中加密和静态加密。该选项适用于静态加密。选择为文件系统启用一种或两种类型的加密对 I/O 延迟和吞吐量的影响很小。

有关 Amazon EFS 对通用性能模式中总文件系统吞吐量、每实例吞吐量和每秒操作数的限制的信息，请参阅[Amazon EFS 限制 \(p. 90\)](#)。

## 相关主题

- [本地性能注意事项 \(p. 82\)](#)
- [Amazon EFS 性能提示 \(p. 83\)](#)
- [计量 – Amazon EFS 如何报告文件系统和对象大小 \(p. 46\)](#)
- [Amazon EFS 故障排除 \(p. 94\)](#)

# 安全性

您可以在下文中找到有关使用 Amazon EFS 的安全注意事项说明。对于 Amazon EFS 文件系统，可以考虑四种级别的访问控制，每种级别使用不同的机制。

## 主题

- [API 调用的 AWS Identity and Access Management \(IAM\) 权限 \(p. 84\)](#)
- [Amazon EC2 实例和挂载目标的安全组 \(p. 84\)](#)
- [EFS 文件和目录的读取、写入和执行权限 \(p. 86\)](#)
- [使用 EFS 所需的源端口 \(p. 86\)](#)
- [在 EFS 中加密数据和元数据 \(p. 86\)](#)

## API 调用的 AWS Identity and Access Management (IAM) 权限

您可以调用 Amazon EFS API 以创建、管理和删除文件系统。如果调用方使用 AWS Identity and Access Management (IAM) 用户或担任的角色的凭证，则每个 API 调用要求调用方在其 IAM 策略中具有调用的操作的权限。有些 API 操作支持作为调用对象的文件系统特定的策略权限 (即资源级别权限)。使用账户的根凭证进行的 API 调用具有该账户拥有的文件系统上的所有 API 操作的权限。

IAM 权限的一个示例是，IAM 用户 Alice 可能有权检索其父 AWS 账户中的所有文件系统的说明。但是，可能仅允许她管理其中的一个文件系统的安全组，即，具有 ID `fs-12345678` 的文件系统。

有关 Amazon EFS API 的 IAM 权限的更多信息，请参阅[Amazon EFS 的身份验证和访问控制 \(p. 144\)](#)。

## Amazon EC2 实例和挂载目标的安全组

使用 Amazon EFS 时，需要为 EC2 实例指定 Amazon EC2 安全组，并为与文件系统关联的 EFS 挂载目标指定安全组。安全组充当防火墙，您添加的规则控制流量。在入门练习中，您在启动 EC2 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 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，以允许 EC2 实例使用网络文件系统 (NFS) 端口通过挂载目标连接到 Amazon EFS 文件系统。请执行以下步骤以创建和更新您的安全组。

### 为 EC2 实例和挂载目标创建安全组

1. 在 VPC 中创建两个安全组。

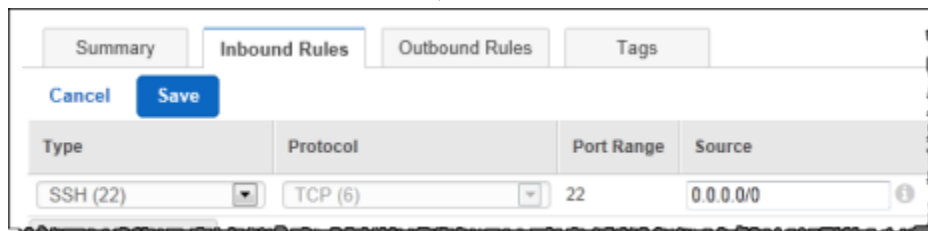
有关说明，请参阅 Amazon VPC 用户指南的[创建安全组](#)中的“创建安全组”过程。

2. 打开 Amazon VPC 管理控制台 (<https://console.aws.amazon.com/vpc/>)，并验证这些安全组的默认规则。两个安全组都应当只有一条允许出站流量的出站规则。

### 更新安全组必要的访问权限

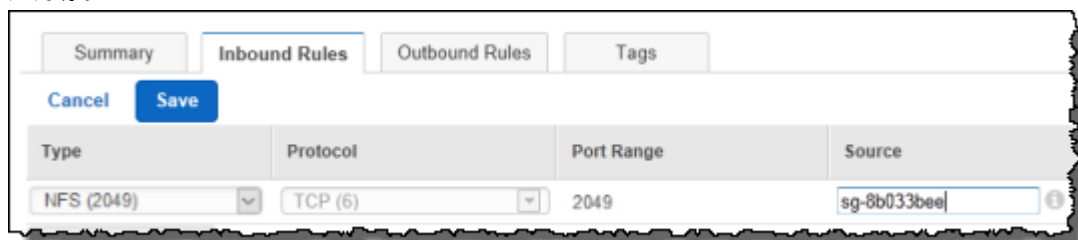
1. 打开 Amazon VPC 控制台 <https://console.aws.amazon.com/vpc/>。
2. 为 EC2 安全组添加一个规则，以允许从任何主机中使用安全 Shell (SSH) 进行入站访问。或者，限制源地址。

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。如果不是这种情况，您需要添加一个出站规则以在 NFS 端口上打开 TCP 连接，从而将挂载目标安全组指定为目标。



有关说明，请参阅 Amazon VPC 用户指南 中的 [添加和删除规则](#)。

3. 为挂载目标安全组添加一个规则，以允许从 EC2 安全组中进行入站访问，如下所示。EC2 安全组将指定为源。



4. 确认两个安全组现在授予了入站和出站访问权限。

有关安全组的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 [EC2-VPC 安全组](#)。

## 挂载 Amazon EFS 文件系统的安全注意事项

只有在可以建立到文件系统的某个挂载目标的 NFS 端口的网络连接时，NFSv4.1 客户端才能挂载该文件系统。同样，只有在可以建立该网络连接时，NFSv4.1 客户端才能在访问文件系统时声明用户和组 ID。

建立此网络连接的能力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。仅 Amazon VPC 中的 Amazon EC2 实例或使用 AWS Direct Connect 连接到 Amazon VPC 的本地服务器可以挂载 Amazon EFS 文件系统。

目前，您无法使用其他机制从 VPC 外部连接到 VPC 的私有 IP 地址以挂载 Amazon EFS 文件系统。例如，您无法使用 VPN 连接或 VPC 对等执行该操作。不要依靠此类其他方法进行文件系统访问控制。

- 客户端和挂载目标的 VPC 子网的网络访问控制列表 (ACL)，用于从挂载目标的子网外部进行访问 – 要挂载文件系统，客户端必须能够建立到挂载目标的 NFS 端口的 TCP 连接并接收返回的流量。
- 客户端和挂载目标的 VPC 安全组规则，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允许在 NFS 端口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，其安全组规则允许建立到文件系统的某个挂载目标上的 NFS 端口的出站连接。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 [创建挂载目标](#) (p. 20)。

## EFS 文件和目录的读取、写入和执行权限

EFS 文件系统上的文件和目录支持 Unix 风格的标准读取、写入和执行权限，这些权限基于通过挂载 NFSv4.1 客户端声明的用户和组 ID。有关更多信息，请参阅 [网络文件系统 \(NFS\) 级别用户、组和权限 \(p. 26\)](#)。

### Note

这个访问控制层取决于在用户和组 ID 声明中信任 NFSv4.1 客户端。在建立挂载连接时，不会对 NFSv4.1 客户端进行身份验证。因此，如果任何 NFSv4.1 客户端可以建立到文件系统挂载目标 IP 地址的 NFS 端口的网络连接，则可以作为根用户 ID 读取和写入文件系统。

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。

## 使用 EFS 所需的源端口

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。

```
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 (-m tcp -p tcp -dport 2049) 的连接。

## 在 EFS 中加密数据和元数据

Amazon EFS 支持两种形式的文件系统加密：传输中的数据加密和静态加密。您可以在创建 Amazon EFS 文件系统时启用静态数据加密。您可以在挂载文件系统时启用传输中的数据加密。

### 何时使用加密

如果您的组织的公司或监管策略要求静态加密数据和元数据，我们建议您创建加密的文件系统以挂载使用传输中的数据加密的文件系统。

### 加密传输中的数据

您可以使用 Amazon EFS 文件系统加密传输中的数据，而无需修改您的应用程序。

### 使用 TLS 加密传输中的数据

可以在使用 Amazon EFS 挂载帮助程序挂载文件系统时启用传输层安全性 (TLS)，以便为您的 Amazon EFS 文件系统启用传输中的数据加密。有关更多信息，请参阅 [使用 EFS 挂载帮助程序进行挂载 \(p. 59\)](#)。

在将传输中的数据加密声明为 Amazon EFS 文件系统的挂载选项时，挂载帮助程序初始化客户端 stunnel 进程。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。挂载帮助程序使用传输层安全性 (TLS) 1.2 版与您的文件系统进行通信。



## 使用挂载帮助程序挂载 Amazon EFS 文件系统并启用传输中的数据加密

1. 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。有关如何执行该操作的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[使用 SSH 连接到您的 Linux 实例](#)。
2. 运行以下命令以挂载文件系统。

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

## 传输中加密的工作方式

传输中的数据加密是通过使用 TLS 连接到 Amazon EFS 启用的。我们建议您使用挂载帮助程序，因为这是最简单的方法。

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。以下是完成该操作所需的简要步骤：

在不使用挂载帮助程序的情况下启用传输中的数据加密

1. 下载并安装 stunnel，并记下该应用程序侦听的端口。
2. 运行 stunnel 以使用 TLS 通过端口 2049 连接到您的 Amazon 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 进程侦听本地端口 20049 和 20449，并通过端口 2049 连接到 Amazon EFS。

### Note

默认情况下，在将 Amazon EFS 挂载帮助程序与 TLS 一起使用时，它强制使用在线证书状态协议 (OCSP) 和证书主机名检查。Amazon 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 版本之一时，使用 TLS 挂载 Amazon 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 \(p. 36\)](#)。有关加密问题，请参阅[排除加密故障 \(p. 102\)](#)。

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,rsize=1048576,wsiz=1048576,namlen=255,hard,proto=tcp,port=20127,timeo=600,retra
```

在使用 TLS 和 Amazon 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。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 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。请务必将 [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
```

在将挂载帮助程序用于传输中的数据加密时，它还会创建一个名为 amazon-efs-mount-watchdog 的进程。该进程确保每个挂载的 stunnel 进程正在运行，并在卸载 Amazon 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。

## 静态加密数据

与未加密的文件系统一样，您可以通过 AWS 管理控制台、AWS CLI 或以编程方式通过 Amazon EFS API 或某个 AWS 开发工具包创建加密的文件系统。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。

您可以使用 Amazon CloudWatch 和 AWS CloudTrail 检测创建的文件系统并验证是否启用了加密，以便为 Amazon EFS 文件系统实施数据加密策略。有关更多信息，请参阅 [演练 6：在 Amazon EFS 文件系统中实施静态加密 \(p. 132\)](#)。

### Note

AWS 密钥管理基础设施使用联邦信息处理标准 (FIPS) 140-2 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。

## 使用控制台静态加密文件系统

您可以选择在创建文件系统时为其启用静态加密。以下过程说明了从控制台中创建新的文件系统时如何为其启用加密。

在控制台上加新的文件系统

1. 通过 <https://console.aws.amazon.com/efs/> 打开 Amazon Elastic File System 控制台。
2. 选择创建文件系统以打开文件系统创建向导。
3. 对于步骤 1：配置文件系统访问，请选择您的 VPC，创建您的挂载目标，然后选择下一步。
4. 对于步骤 2：配置可选设置，请添加任何标签，选择您的性能模式，选中加密您的文件系统的复选框，然后选择下一步。
5. 对于步骤 3：审核和创建，请检查您的设置，然后选择创建文件系统。

现在，您具有新的静态加密的文件系统。

## 静态加密的工作方式

在静态加密的文件系统中，在将数据和元数据写入到文件系统之前，将自动对其进行加密。同样，在读取数据和元数据时，在将其提供给应用程序之前，将自动对其进行解密。这些过程是 Amazon EFS 透明处理的，因此，您不必修改您的应用程序。

Amazon EFS 使用行业标准 AES-256 加密算法静态加密 EFS 数据和元数据。有关更多信息，请参阅 AWS Key Management Service Developer Guide 中的 [加密基础知识](#)。

## Amazon EFS 如何使用 AWS KMS

Amazon EFS 与 AWS Key Management Service (AWS KMS) 集成在一起以进行密钥管理。Amazon EFS 使用客户主密钥 (CMK) 通过以下方法加密您的文件系统：

- 静态加密元数据 – 使用 EFS 托管密钥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 CMK。您可以启用、禁用或撤销对该 CMK 的授权。该 CMK 可以采用以下两种类型之一：
  - AWS 托管 CMK – 这是默认 CMK，并且可以免费使用。
  - 客户托管 CMK – 这是使用最灵活的主密钥，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 CMK 的更多信息，请参阅 AWS Key Management Service Developer Guide 中的 [创建密钥](#)。

如果将客户托管 CMK 作为主密钥以加密和解密文件数据，您可以启用密钥轮换。在启用密钥轮换时，AWS KMS 自动每年轮换一次您的密钥。此外，对于客户托管 CMK，您可以随时选择何时禁用、重

新启用、删除或撤销您的 CMK 的访问权限。有关更多信息，请参阅 [禁用、删除或撤销文件系统的 CMK 访问权限 \(p. 57\)](#)。

静态数据加密和解密是透明处理的。但是，Amazon EFS 特定的 AWS 账户 ID 显示在与 AWS KMS 操作相关的 AWS CloudTrail 日志中。有关更多信息，请参阅 [静态加密的文件系统的 Amazon EFS 日志文件条目 \(p. 76\)](#)。

## AWS KMS 的 Amazon EFS 密钥策略

密钥策略是控制对 CMK 访问的主要方法。有关密钥策略的更多信息，请参阅 AWS Key Management Service Developer Guide 中的[在 AWS KMS 中使用密钥策略](#)。以下列表描述了 Amazon EFS 为静态加密的文件系统支持的所有 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 ) 使用新的客户主密钥 (CMK) 加密服务器端的数据，而不公开客户端的数据明文。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 – ( 必需 ) 返回根据 CMK 加密的数据加密密钥。该权限包含在默认密钥策略中的 kms:GenerateDataKey\* 下面。
- kms:CreateGrant – ( 必需 ) 为密钥添加授权以指定哪些用户可以在什么条件下使用密钥。授权是密钥策略的替代权限机制。有关授权的更多信息，请参阅 AWS Key Management Service Developer Guide 中的[使用授权](#)。该权限包含在默认密钥策略中。
- kms:DescribeKey – ( 必需 ) 提供有关指定的客户主密钥的详细信息。该权限包含在默认密钥策略中。
- kms:ListAliases – ( 可选 ) 列出账户中的所有密钥别名。在使用控制台创建加密的文件系统时，该权限将填充选择 KMS 主密钥列表。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。

## 相关主题

有关使用 Amazon EFS 进行加密的更多信息，请参阅以下相关主题：

- [为 Amazon EFS 创建资源 \(p. 16\)](#)
- [管理对加密的文件系统的访问 \(p. 57\)](#)
- [Amazon EFS 性能提示 \(p. 83\)](#)
- [Amazon EFS API 权限：操作、资源和条件参考 \(p. 151\)](#)
- [静态加密的文件系统的 Amazon EFS 日志文件条目 \(p. 76\)](#)
- [排除加密故障 \(p. 102\)](#)



# Amazon EFS 限制

您可以在下文中找到在使用 Amazon EFS 时的限制。

## 主题

- [您可以提高的 Amazon EFS 限制 \(p. 90\)](#)
- [资源限制 \(p. 91\)](#)
- [客户端 EC2 实例的限制 \(p. 91\)](#)
- [Amazon EFS 文件系统的限制 \(p. 92\)](#)
- [EFS 文件同步限制 \(p. 92\)](#)
- [不支持的 NFSv4 功能 \(p. 92\)](#)
- [其他注意事项 \(p. 93\)](#)

## 您可以提高的 Amazon EFS 限制

以下是可通过联系 AWS Support 提高的 Amazon EFS 限制。这些是吞吐量限制，它们由您为文件系统选择的吞吐量模式定义，即突发或预配置。有关这些模式的更多信息，请参阅[Amazon EFS 性能 \(p. 78\)](#)。

突发吞吐量模式限制如下。

资源	默认限制
所有连接客户端的总吞吐量	美国东部（俄亥俄）区域 – 3 GB/s 美国东部（弗吉尼亚北部）地区 – 3 GB/s 美国西部（加利福尼亚北部）区域 – 1 GB/s 美国西部（俄勒冈）区域 – 3 GB/s 亚太区域（首尔）– 1 GB/s 亚太区域（东京）– 1 GB/s 欧洲（法兰克福）区域 – 1 GB/s 欧洲（爱尔兰）区域 – 3 GB/s 亚太区域（悉尼）– 3 GB/s

预配置吞吐量模式限制如下。

资源	默认限制
所有客户端的总吞吐量	所有区域 – 1 GB/s

您可以通过执行以下步骤来请求提高这些限制。我们不会立即同意提高这些限制，因此，您的提高请求可能需要几天才能生效。

#### 申请提高限制

1. 打开 [AWS Support Center](#) 页面，登录（如有必要），然后选择 Create Case。
2. 在 Regarding 下，选择 Service Limit Increase。
3. 在 Limit Type 下，选择要提高的 limits 的类型，填写表单中的必填字段，然后选择您的首选联系方式。

## 资源限制

以下是 AWS 区域中的每个客户账户的 Amazon EFS 资源限制。

资源	限制
文件系统数	美国东部（俄亥俄）区域 – 125 美国东部（弗吉尼亚北部）地区 – 70 美国西部（加利福尼亚北部）区域 – 125 美国西部（俄勒冈）区域 – 125 亚太区域（首尔）125 亚太区域（东京）125 欧洲（法兰克福）区域 – 125 欧洲（爱尔兰）区域 – 125 亚太区域（悉尼）– 125
可用区中的每个文件系统的挂载目标数	1
每个挂载目标的安全组数	5
每个文件系统的标签数	50
每个文件系统的 VPC 数	1

## 客户端 EC2 实例的限制

客户端 EC2 实例的以下限制适用，假定是 Linux NFSv4.1 客户端：

- 每个 Amazon EC2 实例可实现的最大吞吐量为 250 MB/s。
- 每个实例的最多 128 个活动用户账户可能同时打开文件。每个用户账户表示一个登录到实例的本地用户。
- 实例上可同时打开最多 32,768 个文件。
- 实例上的每个唯一挂载可以在最多 256 个唯一文件/进程对中最多获取总共 8192 个锁。例如，单个进程可以在 256 个单独的文件上获取一个或多个锁，或者说 8 个进程中的每个进程均可以在 32 个文件上获取一个或多个锁。
- 不支持将 Amazon EFS 与 Microsoft Windows Amazon EC2 实例结合使用。

## Amazon EFS 文件系统的限制

以下是特定于 Amazon EFS 文件系统的限制：

- 最大名称长度：255 字节。
- 最大符号链接 (symlink) 长度：4080 字节。
- 文件的最大硬链接数：177。
- 单个文件的最大大小：52,673,613,135,872 字节 (47.9 TiB)。
- 最大目录深度：1000 级。
- 任何一个特定文件可以在文件系统的所有用户中具有最多 87 个锁。您可以在一个或多个 Amazon EC2 实例上挂载文件系统，但文件的最多 87 个锁限制适用。
- 在通用模式中，每秒的文件系统操作数为 7000 个。将针对连接到单个文件系统的所有客户端计算该操作限制。

## EFS 文件同步限制

以下是 AWS 区域中的每个客户账户的 EFS 文件同步资源限制。

资源	限制
最大同步任务数	10
每个同步任务的最大文件数	35,000,000
每个同步任务的最大文件接收速率	每秒 500 个文件
每个同步任务的最大数据吞吐量	1Gbps

## 不支持的 NFSv4 功能

虽然 Amazon Elastic File System 不支持 NFSv2 或 NFSv3，但 Amazon EFS 支持 NFSv4.1 和 NFSv4.0，以下功能除外：

- pNFS
- 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 CLAIM\_DELEGATE\_CUR 和 CLAIM\_DELEGATE\_PREV 声明类型返回 NFSERR\_NOTSUPP。
- 强制锁定

Amazon EFS 中的所有锁定都是建议性锁定，这意味着 READ 和 WRITE 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，它主要由用户的 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 (ACL)
- 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 (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 CREATE 操作时不支持的文件类型：块储存设备 (NF4BLK)、字符设备 (NF4CHR)、属性目录 (NF4ATTRDIR) 和命名属性 (NF4NAMEDATTR)。
- 不支持的属性：  
FATTR4\_ARCHIVE、FATTR4\_FILES\_AVAIL、FATTR4\_FILES\_FREE、FATTR4\_FILES\_TOTAL、FATTR4\_FS\_LOCA 和 FATTR4\_ACL。

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：

- 有关您可以创建 Amazon EFS 文件系统的 AWS 区域的列表，请参阅 [AWS General Reference](#)。
- 2012 年之前创建的某些 AWS 账户或许能够访问 us-east-1 中不支持创建挂载目标的可用区。如果无法在某个 AWS 区域中创建挂载目标，请尝试使用该 AWS 区域中的其他可用区。不过，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。
- 您可以使用在 VPC 中创建的挂载目标，从 VPC 的 EC2 实例中挂载文件系统。您还可以在 EC2-Classical 实例（不在 VPC 中）上挂载文件系统，但必须先使用 ClassicLink 将其链接到您的 VPC。有关使用 ClassicLink 的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 [ClassicLink](#)。
- 您可以通过 AWS Direct Connect 从本地数据中心服务器中挂载 Amazon EFS 文件系统。
- 在使用 C5 或 M5 实例时，支持单个 AWS 区域中的 VPC 对等。但是，不支持使用其他实例类型的其他 VPC 私有连接机制，如 VPN 连接、区域间 VPC 对等和区域内 VPC 对等。

# Amazon EFS 故障排除

您可以在下文中查找有关如何解决 Amazon Elastic File System (Amazon EFS) 问题的信息。

## 主题

- [Amazon EFS 故障排除：一般问题 \(p. 94\)](#)
- [解决文件操作错误 \(p. 96\)](#)
- [解决 AMI 和内核问题 \(p. 97\)](#)
- [解决挂载问题 \(p. 99\)](#)
- [排除加密故障 \(p. 102\)](#)

## Amazon EFS 故障排除：一般问题

您可以在下文中查找有关对 Amazon EFS 相关的一般故障排除问题的信息。有关性能的信息，请参阅 [Amazon EFS 性能 \(p. 78\)](#)。

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：

- Amazon Linux 2015.09 或更高版本
- RHEL 7.3 或更高版本
- 具有内核 2.6.32-696 或更高版本的 RHEL 6.9
- 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。

## 主题

- [Amazon EC2 实例挂起 \(p. 94\)](#)
- [写入大量数据的应用程序挂起 \(p. 94\)](#)
- [打开和关闭操作被序列化 \(p. 95\)](#)
- [自定义 NFS 设置导致写入延迟 \(p. 95\)](#)
- [使用 Oracle Recovery Manager 创建备份的速度很慢 \(p. 96\)](#)

## 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。

## 措施

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 \(p. 63\)](#)。

## 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。

#### 措施

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
```

## 打开和关闭操作被序列化

在单个 Amazon EC2 实例上由用户在文件系统中执行的打开和关闭操作被序列化。

#### 措施

要解决该问题，请使用 NFS 协议 4.1 版和建议的 Linux 内核之一。通过在挂载文件系统时使用 NFSv4.1，可以对文件启用并行打开和关闭操作。我们建议您将 Amazon Linux AMI 2016.03.0 作为将文件系统挂载到的 Amazon EC2 实例的 AMI。

如果无法使用 NFSv4.1，请注意，Linux NFSv4.0 客户端按用户 ID 和组 ID 序列化打开和关闭请求。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。仅当所有 ID 均匹配时，客户端才一次向 NFS 服务器发送一个打开或关闭操作。

此外，您还可以执行以下任意操作来解决该问题：

- 您可以在同一 Amazon EC2 实例上通过不同用户 ID 运行每个进程。
- 您可以对所有打开请求使用相同的用户 ID，并修改组 ID 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。

#### 措施

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，已启用 NFSv4.1 中的属性缓存。

#### 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。

### 措施

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。

## 解决文件操作错误

当您访问 Amazon EFS 文件系统时，对文件系统中的文件的某些限制可能适用。超出这些限制会导致文件操作错误。有关 Amazon EFS 中基于客户端和文件的限制的更多信息，请参阅[客户端 EC2 实例的限制 \(p. 91\)](#)。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。

## 命令失败，并显示“Disk quota exceeded”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：

- 一个实例同一时刻最多可以有 128 个活动用户账户打开文件。
- 一个实例同一时刻最多可打开 32,768 个文件。
- 实例上的每个唯一挂载可以在 256 个唯一文件/进程对中最多获取总共 8192 个锁。例如，单个进程可以在 256 个单独的文件上获取一个或多个锁，或者说 8 个进程中的每个进程均可以在 32 个文件上获取一个或多个锁。

### 措施

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。

## 命令失败，并显示“I/O error”

遇到下列问题之一时会发生此错误：

- 每个实例同一时刻最多有 128 个活动用户账户打开文件。

### 措施

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。

- 已删除加密您的文件系统的 AWS KMS 密钥。

### 措施

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“File name is too long”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。

### 措施

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“Too many links”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。

### 措施

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“File too large”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)。

### 措施

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。

## 命令失败，并显示“Try again”错误

如果尝试访问单个文件的用户或应用程序太多，则会出现该错误。当应用程序或用户访问某个文件时，会对该文件实施锁定。在文件系统的所有用户和应用程序中，任何一个特定文件最多可以有 87 个锁。您可以将一个文件系统挂载到一个或多个 Amazon EC2 实例上，但一个文件 87 个锁的限制仍然适用。

### 措施

如果遇到该问题，可通过减少访问该文件的应用程序或用户的数量，直到该数量满足允许的锁数量或更少，加以解决。

# 解决 AMI 和内核问题

下文介绍了如何解决在从 Amazon EC2 实例使用 Amazon EFS 时遇到的与特定 Amazon 系统映像 (AMI) 或内核版本相关的问题。

### 主题

- [无法更改所有权 \(p. 98\)](#)
- [由于客户端错误，文件系统重复执行操作 \(p. 98\)](#)
- [客户端发生死锁 \(p. 98\)](#)
- [列出大型目录中的文件需要很长时间 \(p. 98\)](#)



## 无法更改所有权

当使用 Linux `chown` 命令时，无法更改文件/目录的所有权。

出现该错误的内核版本

2.6.32

措施

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。

措施

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

措施

执行以下任一操作：

- 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 2.6.32-696.1.1.el6 的 CentOS 内核版本

措施

要解决该问题，请升级到较新的内核版本。

## 解决挂载问题

您可以在下文中找到有关解决 Amazon 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 (p. 99)
- 自动挂载失败，并且实例没有响应 (p. 99)
- 在 /etc/fstab 中挂载多个 Amazon EFS 文件系统失败 (p. 99)
- 挂载命令失败，并显示“wrong fs type”错误消息 (p. 100)
- 挂载命令失败，并显示“incorrect mount option”错误消息 (p. 100)
- 在创建文件系统后文件系统挂载立即失败 (p. 100)
- 文件系统挂载挂起，然后失败，并显示超时错误 (p. 101)
- 使用 DNS 名称的文件系统挂载失败 (p. 101)
- 挂载目标生命周期状态停滞 (p. 101)
- 挂载没有响应 (p. 101)
- 针对新挂载的文件系统的操作返回“bad file handle”错误 (p. 102)
- 卸载文件系统失败 (p. 102)

### 在 Windows 实例上挂载文件系统失败

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

#### 措施

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

### 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。

#### 措施

如果出现该问题，请与 AWS Support 联系。

### 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
```

#### 措施

在这种情况下，我们建议您使用以下内容在 `/etc/systemd/system/mount-nfs-sequentially.service` 中创建新的 `systemd` 服务文件。

```
[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
```

在执行该操作后，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“wrong fs type”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.
```

### 措施

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](#) (p. 221)。

## 挂载命令失败，并显示“incorrect mount option”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
```

### 措施

该错误消息很可能意味着您的 Linux 发行版不支持 4.0 和 4.1 版网络文件系统 (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。有关支持 NFSv4.1 的 Amazon Elastic Compute Cloud (Amazon EC2) 的 Amazon 系统映像 (AMI) 列表，请参阅 [NFS 支持](#) (p. 220)。

## 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，可能最多需要 90 秒的时间才能在 AWS 区域中完全传播。

### 措施

如果以编程方式创建和挂载文件系统（例如，使用 AWS CloudFormation 模板），我们建议您实施等待条件。

## 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
$
```

### 措施

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。有关更多信息，请参阅 [创建安全组 \(p. 24\)](#)。

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 DNS 名称的文件系统挂载失败

使用 DNS 名称的文件系统挂载失败。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 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.

$
```

### 措施

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 [在您的 VPC 中使用 DNS](#)。

要在 mount 命令中指定一个 DNS 名称，您必须执行以下操作：

- 确保 Amazon EC2 实例所在的同一可用区中有一个 Amazon EFS 挂载目标。
- 在配置为使用由 Amazon 提供的 DNS 服务器的 Amazon VPC 内连接到您的 Amazon EC2 实例。有关更多信息，请参阅 Amazon VPC 用户指南中的 [DHCP 选项集](#)。
- 确保连接的 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 [更新 VPC 的 DNS 支持](#)。

## 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。

### 措施

重试 CreateMountTarget 或 DeleteMountTarget 调用。

## 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。

#### 措施

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。

## 针对新挂载的文件系统的操作返回“bad file handle”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。

#### 措施

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](#) (p. 63)。

## 卸载文件系统失败

如果文件系统繁忙，则无法将其卸载。

#### 措施

您可以通过以下方法解决该问题：

- 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 `-f` 选项强制完成 `umount` 命令。

#### 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。

## 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。

- [具有传输中的数据加密的挂载失败](#) (p. 103)
- [具有传输中的数据加密的挂载中断](#) (p. 103)
- [无法创建静态加密的文件系统](#) (p. 103)
- [无法使用的加密文件系统](#) (p. 103)

## 具有传输中的数据加密的挂载失败

默认情况下，在将 Amazon EFS 挂载帮助程序与 TLS 一起使用时，它强制使用在线证书状态协议 (OCSP) 和证书主机名检查。如果您的系统不支持任一功能（例如，在使用 Red Hat Enterprise Linux 或 CentOS 时），使用 TLS 挂载 EFS 文件系统将失败。

### 措施

我们建议您升级客户端上的 stunnel 版本以支持这些功能。有关更多信息，请参阅 [升级 stunnel \(p. 36\)](#)。

## 具有传输中的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您的 Amazon EFS 文件系统的加密连接挂起或中断。

### 措施

如果到使用传输中的数据加密的 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 \(p. 38\)](#)。
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/amazon-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。

如果仍然中断，请与 AWS Support 联系。

## 无法创建静态加密的文件系统

您已尝试创建新的静态加密的文件系统。不过，您会收到一条错误消息，指出 AWS KMS 不可用。

### 措施

在极少数情况下，AWS KMS 可能在您的 AWS 区域中暂时不可用，从而出现该错误。如果发生这种情况，请等到 AWS KMS 恢复完全可用，然后重试以创建文件系统。

## 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。如果由于以下原因之一 EFS 无法从 AWS KMS 中检索主密钥，则可能会出现这些错误：

- 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。

### 措施

首先，确认已启用 AWS KMS 密钥。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 Developer Guide 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 Developer Guide 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 Developer Guide 中的[计划和取消密钥删除](#)。

如果已启用密钥并且仍遇到问题，或者在重新启用密钥时遇到问题，请与 AWS Support 联系。



# EFS 文件同步故障排除

您可以在下文中找到有关如何解决 EFS 文件同步问题的信息。

## 主题

- [本地源文件系统停滞在“正在挂载”状态 \(p. 105\)](#)
- [Amazon EC2 源文件系统停滞在“正在挂载”状态 \(p. 105\)](#)
- [同步任务停滞在“正在启动”状态 \(p. 106\)](#)
- [同步任务失败，并显示“权限被拒绝”错误消息 \(p. 106\)](#)
- [完成同步任务的“正在准备”状态需要多长时间？ \(p. 106\)](#)
- [完成同步任务的“正在验证”状态需要多长时间？ \(p. 106\)](#)
- [允许 AWS Support 帮助解决在本地运行的 EFS 文件同步问题 \(p. 107\)](#)
- [允许 AWS Support 帮助解决在 Amazon EC2 上运行的 EFS 文件同步问题 \(p. 108\)](#)

## 本地源文件系统停滞在“正在挂载”状态

在所选的同步代理无法挂载在配置期间指定的位置时，本地源文件系统可能会停滞在正在挂载状态。

### 措施

首先，确保您指定的 NFS 服务器和导出均有效。如果它们无效，请删除同步集，使用正确的 NFS 服务器创建新的同步集，然后导出。

如果 NFS 服务器和导出均有效，则通常表明出现以下两种问题之一。防火墙禁止同步代理挂载 NFS 服务器，或者 NFS 服务器未配置为允许同步代理挂载该服务器。

确保在同步代理和 NFS 服务器之间没有防火墙。然后，确保 NFS 服务器配置为允许同步代理挂载在同步集中指定的导出。

如果执行这些操作并且同步代理仍无法挂载 NFS 服务器和导出，请打开支持渠道并与 AWS 客户支持部门联系。有关如何打开支持渠道的信息，请参阅[允许 AWS Support 帮助解决在本地运行的 EFS 文件同步问题 \(p. 107\)](#)或[允许 AWS Support 帮助解决在 Amazon EC2 上运行的 EFS 文件同步问题 \(p. 108\)](#)。

## Amazon EC2 源文件系统停滞在“正在挂载”状态

在所选的同步代理无法挂载在配置期间指定的位置时，Amazon EC2 源文件系统可能会停滞在正在挂载状态。

### 措施

首先，确保您指定的 NFS 服务器和导出均有效。如果它们无效，请删除同步集，使用正确的 NFS 服务器创建新的同步集，然后导出。

如果 NFS 服务器和导出均有效，则通常表明出现以下两种问题之一。防火墙禁止同步代理挂载 NFS 服务器，或者 NFS 服务器未配置为允许同步代理挂载该服务器。

确保 NFS 服务器所在的 VPC 具有一个安全组入站规则，该规则允许传输到为源文件系统创建的同步代理的所有流量。然后，确保在其中运行同步代理的 VPC 具有一个安全组出站规则，该规则允许从同步代理传输的所有流量。

如果执行这些操作并且同步代理仍无法挂载 NFS 服务器和导出，请打开支持渠道并与 AWS 客户支持部门联系。有关如何打开支持渠道的信息，请参阅[允许 AWS Support 帮助解决在本地运行的 EFS 文件同步问题 \(p. 107\)](#)或[允许 AWS Support 帮助解决在 Amazon EC2 上运行的 EFS 文件同步问题 \(p. 108\)](#)。



## 同步任务停滞在“正在启动”状态

在 EFS 文件同步无法指示指定的源同步代理开始执行同步任务时，同步任务可能会停滞在正在启动状态。出现该问题通常是因为，同步代理已关闭电源或断开网络连接。

### 措施

确保已连接源同步代理，并且状态为正在运行。如果状态为脱机，则未连接代理。

接下来，确保同步代理已打开电源。否则，请打开同步代理电源。

如果同步代理已打开电源并且同步任务仍停滞在正在启动状态，则很可能在同步代理和 EFS 文件同步之间出现网络连接问题。检查网络和防火墙设置，以确保同步代理可以连接到 EFS 文件同步。

如果执行这些操作并且未解决该问题，请打开支持渠道并与 AWS 客户支持部门联系。有关如何打开支持渠道的信息，请参阅[允许 AWS Support 帮助解决在本地运行的 EFS 文件同步问题 \(p. 107\)](#)或[允许 AWS Support 帮助解决在 Amazon EC2 上运行的 EFS 文件同步问题 \(p. 108\)](#)。

## 同步任务失败，并显示“权限被拒绝”错误消息

如果配置 NFS 服务器并启用 `root_squash` 或 `all_squash`，并且您的文件没有所有读取访问权限，则可能会显示“权限被拒绝”错误消息。

### 措施

要解决该问题，请为 NFS 导出配置 `no_root_squash`，或者确保要同步的所有文件的权限允许所有用户进行读取访问。通过执行上述任一操作，将允许同步代理读取这些文件。要使同步代理能够访问目录，您还必须启用所有执行访问。有关 NFS 导出配置的信息，请参阅 Centos 文档中的[18.7. /etc/exports 配置文件](#)。

如果执行这些操作并且未解决该问题，请与 AWS 客户支持部门联系。

## 完成同步任务的“正在准备”状态需要多长时间？

EFS 文件同步处于正在准备状态的时间取决于源和目标文件系统中的文件数以及这些文件系统的性能。在同步任务启动时，EFS 文件同步执行递归目录列表操作以查找源和目标文件系统中的所有文件和文件元数据。这些列表用于查找差异和确定要复制的内容。

### 要采取的操作

您不需要采取任何措施。等待正在准备状态完成，并且状态变为正在同步。如果状态未变为正在同步，请与 AWS 客户支持部门联系。

## 完成同步任务的“正在验证”状态需要多长时间？

EFS 文件同步处于正在验证状态的时间取决于源和目标文件系统中的文件数、所有文件的总大小以及这些文件系统的性能。默认情况下，将在同步设置中启用验证模式。EFS 文件同步执行的验证包括所有文件内容的 SHA256 校验和以及所有文件元数据的精确比较。

### 要采取的操作

您不需要采取任何措施。等待正在验证状态完成。如果正在验证状态未完成，请与 AWS 客户支持部门联系。

## 允许 AWS Support 帮助解决在本地运行的 EFS 文件同步问题

EFS 文件同步提供了一个本地控制台，可用于执行很多维护任务，包括允许 AWS Support 访问您的 EFS 文件同步以帮助解决 EFS 文件同步问题。默认情况下，禁止 AWS Support 访问您的 EFS 文件同步。您可通过主机的本地控制台启用此访问。要允许 AWS Support 访问您的 EFS 文件同步，请先登录到主机的本地控制台，然后连接到支持服务器。

### 允许 AWS Support 访问 EFS 文件同步

1. 登录到主机的本地控制台。请使用用户名 `admin` 和密码 `password`。

本地控制台类似如下所示。

```
AWS Appliance Activation - Configuration

#####
## Currently connected network adapters:
##
## eth0: 172.16.0.1/16
#####

1: Network Configuration
2: View System Resource Check (0 Errors)
3: System Time Management
4: Command Prompt

Press "x" to exit session

Enter command: _
```

2. 在提示符下，键入 `4` 以打开帮助菜单。
3. 键入 `h` 以打开 AVAILABLE COMMANDS (可用命令) 窗口。
4. 在 AVAILABLE COMMANDS (可用的命令) 窗口中，键入 `open-support-channel` 以连接到客户支持。必须允许 TCP 端口 22 以启动针对 AWS 的支持渠道。在连接到客户支持时，EFS 文件同步将为您分配一个支持编号。请记住您的支持编号。

```
AVAILABLE COMMANDS
ip                Show / manipulate routing, devices, and tunnels
save-routing-table Save newly added routing table entry
ifconfig          View or configure network interfaces
iptables         Administration tool for IPv4 packet filtering and NAT
save-iptables     Persist IP tables
passwd            Update authentication tokens
open-support-channel Connect to AWS Support
h                Display available command list
exit             Return to Configuration menu

Command: _
```

### Note

渠道号不是传输控制协议/用户数据报协议 (TCP/UDP) 端口号。相反，它建立到服务器的安全 Shell (SSH) (TCP 22) 连接，并为该连接提供支持渠道。

5. 建立支持通道后，为 AWS Support 提供您的支持服务编号，以便 AWS Support 提供故障排除帮助。
6. 在支持会话完成后，键入 `q` 以将其结束。
7. 键入 `exit` 以注销 EFS 文件同步本地控制台。
8. 按照提示操作退出本地控制台。

## 允许 AWS Support 帮助解决在 Amazon EC2 上运行的 EFS 文件同步问题

EFS 文件同步提供了一个本地控制台，可用于执行很多维护任务，包括允许 AWS Support 访问您的 EFS 文件同步以帮助解决 EFS 文件同步问题。默认情况下，禁止 AWS Support 访问您的 EFS 文件同步。可通过 Amazon EC2 本地控制台启用此访问。可通过安全 Shell (SSH) 登录到 Amazon EC2 本地控制台。要通过 SSH 成功登录，您的实例的安全组必须具有开放 TCP 端口 22 的规则。

### Note

如果将新规则添加到现有安全组，则新规则适用于使用该安全组的所有实例。有关安全组以及如何添加安全组规则的更多信息，请参阅 Amazon EC2 用户指南 中的 [Amazon EC2 安全组](#)。

要允许 AWS Support 连接到您的 EFS 文件同步，请先登录到 Amazon EC2 实例的本地控制台，导航到 EFS 文件同步的控制台，然后提供访问权限。

允许 AWS Support 访问在 Amazon EC2 实例上部署的 EFS 文件同步

1. 登录到 Amazon EC2 实例的本地控制台。有关说明，请转到《Amazon EC2 用户指南》<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html> 中的连接到您的实例。

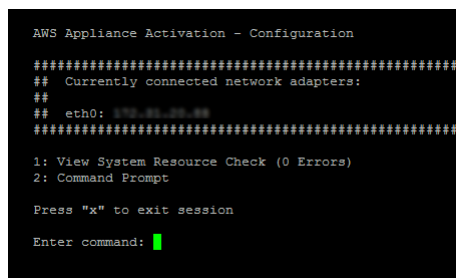
您可使用以下命令登录到 EC2 实例的本地控制台。用户名为 **admin**。

```
ssh -i PRIVATE-KEY admin@INSTANCE-PUBLIC-DNS-NAME
```

### Note

**PRIVATE-KEY** 是 .pem 文件，其中包含用来启动 Amazon EC2 实例的 EC2 密钥对的私有证书。有关更多信息，请参阅 Amazon EC2 用户指南 中的 [检索密钥对的公有密钥](#)。  
**INSTANCE-PUBLIC-DNS-NAME** 是运行 EFS 文件同步的 Amazon EC2 实例的公有域名系统 (DNS) 名称。可通过选择 EC2 控制台中的 Amazon EC2 实例并单击 Description (说明) 选项卡来获取此公共 DNS 名称。

本地控制台类似如下所示。



```
AWS Appliance Activation - Configuration
#####
## Currently connected network adapters:
##
## eth0: 172.31.0.254
#####

1: View System Resource Check (0 Errors)
2: Command Prompt

Press "x" to exit session

Enter command: █
```

2. 在提示符下，键入 **2** 以打开帮助菜单。
3. 键入 **h** 以打开 AVAILABLE COMMANDS (可用命令) 窗口。
4. 在 AVAILABLE COMMANDS (可用的命令) 窗口中，键入 **open-support-channel** 以连接到 EFS 文件同步的客户支持。必须允许 TCP 端口 22 以启动针对 AWS 的支持渠道。在连接到客户支持时，EFS 文件同步将为您分配一个支持编号。请记住您的支持编号。

```
AVAILABLE COMMANDS
ip                Show / manipulate routing, devices, and tunnels
save-routing-table Save newly added routing table entry
ifconfig          View or configure network interfaces
iptables         Administration tool for IPv4 packet filtering and NAT
save-iptables     Persist IP tables
open-support-channel Connect to AWS Support
h                Display available command list
exit             Return to Configuration menu

Command: █
```

#### Note

渠道号不是传输控制协议/用户数据报协议 (TCP/UDP) 端口号。相反，EFS 文件同步建立到 EFS 文件同步服务器的安全 Shell (SSH) (TCP 22) 连接，并为连接提供支持渠道。

5. 建立支持通道后，为 AWS Support 提供您的支持服务编号，以便 AWS Support 提供故障排除帮助。
6. 在支持会话完成后，键入 **q** 以将其结束。
7. 键入 **exit** 以退出 EFS 文件同步控制台。
8. 按照控制台菜单注销 EFS 文件同步实例。

# Amazon Elastic File System 演练

本节提供您可以用来探索 Amazon EFS 并测试端到端设置的演练。

## 主题

- [演练 1：使用 AWS CLI 创建 Amazon EFS 文件系统并将其挂载到 EC2 实例上 \(p. 110\)](#)
- [演练 2：设置 Apache Web 服务器并提供 Amazon EFS 文件服务 \(p. 121\)](#)
- [演练 3：创建可写的每用户子目录以及配置在重启时自动重新挂载 \(p. 126\)](#)
- [演练 4：Amazon EFS 文件系统的备份解决方案 \(p. 127\)](#)
- [演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 \(p. 127\)](#)
- [演练 6：在 Amazon EFS 文件系统上实施静态加密 \(p. 132\)](#)
- [演练 7：使用 EFS 文件同步将文件从本地文件系统同步到 Amazon EFS \(p. 134\)](#)
- [演练 8：使用 EFS 文件同步将文件系统从 Amazon EC2 同步到 Amazon EFS \(p. 138\)](#)

## 演练 1：使用 AWS CLI 创建 Amazon EFS 文件系统并将其挂载到 EC2 实例上

本演练使用 AWS CLI 来探索 Amazon EFS API。在本演练中，您将创建一个 Amazon EFS 文件系统，将其挂载到 VPC 中的 EC2 实例上，然后测试设置。

### Note

本演练类似于入门练习。在[入门 \(p. 9\)](#)练习中，您使用控制台创建 EC2 和 Amazon EFS 资源。在本演练中，您将使用 AWS CLI 来创建它们，主要目的是熟悉 Amazon EFS API。

在本演练中，您将在您的账户中创建以下 AWS 资源：

- Amazon EC2 资源：
  - 两个安全组（一个用于 EC2 实例，一个用于 Amazon EFS 文件系统）。  
您将规则添加到这些安全组中以授权适当的入站/出站访问，从而允许您的 EC2 实例使用标准 NFSv4.1 TCP 端口通过挂载目标连接文件系统。
  - 您的 VPC 中的一个 Amazon 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：工作原理 \(p. 3\)](#)。

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

本演练在美国西部（俄勒冈）区域（us-west-2）创建所有这些资源。不论您使用哪个 AWS 区域，请确保使用方式一致。您的所有资源（VPC、EC2 资源和 Amazon EFS 资源）必须位于同一个 AWS 区域。

## 开始前的准备工作

- 您可以使用 AWS 账户的根凭证登录到控制台并尝试入门练习。但是，AWS Identity and Access Management (IAM) 不建议使用 AWS 账户的根凭证。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[设置 \(p. 7\)](#)。

- 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 [更新 VPC 的 DNS 支持](#)。
  - 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 Internet 网关。
  - 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 VPC 中的 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 adminuser 配置文件。

## 设置 AWS CLI

按照以下说明来设置 AWS CLI 和用户配置文件。

### 设置 AWS CLI

1. 下载并配置 AWS CLI。有关说明，请参阅 AWS Command Line Interface 用户指南 中的以下主题。

[使用 AWS 命令行界面进行设置](#)

[安装 AWS 命令行接口](#)

[配置 AWS 命令行界面](#)

2. 设置配置文件。

您将用户凭证存储在 AWS CLI config 文件中。本演练中的示例 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
```

上述配置文件还设置默认的 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
```

下一步

[步骤 1：创建 Amazon EC2 资源 \(p. 112\)](#)

## 步骤 1：创建 Amazon 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 Amazon EFS 文件系统并挂载到该实例上。

主题

- [步骤 1.1：创建两个安全组 \(p. 112\)](#)
- [步骤 1.2：在安全组中添加规则以授权入站/出站访问 \(p. 113\)](#)
- [步骤 1.3：启动 EC2 实例 \(p. 114\)](#)

### 步骤 1.1：创建两个安全组

在本节中，您将在 VPC 中为 EC2 实例和 Amazon EFS 挂载目标创建安全组。在演练的稍后部分，您要将这些安全组分配给 EC2 实例和 Amazon EFS 挂载目标。有关安全组的信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[EC2-VPC 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`)。您需要提供 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 Amazon 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 Amazon EFS 挂载目标之间的流量 (在本演练的稍后部分，您会将这些安全组与它们关联)。

## 步骤 1.2：在安全组中添加规则以授权入站/出站访问

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. 授权到 EC2 实例安全组 (efs-walkthrough1-ec2-sg) 的传入 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 Amazon 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 adminuser 配置文件运行下面的 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：

a. 密钥对名称。

- 有关介绍性信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[使用 Amazon EC2 进行设置](#)。
- 有关创建 .pem 文件的说明，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[创建密钥对](#)。

b. 要启动的 AMI 的 ID。

您用来启动 EC2 实例的 AWS CLI 命令需要一个 AMI ID (您想要部署的 AMI) 作为参数。本练习使用 Amazon Linux HVM AMI。

#### Note

您可以使用大部分通用的基于 Linux 的 AMI。如果您使用另一个 Linux API，请记住，您将使用 yum 在实例上安装 NFS 客户端，并且您可能需要根据情况添加软件包。

对于 Amazon Linux HVM AMI，您可以在 [Amazon Linux AMI](#) 找到最新的 ID。您从 Amazon Linux AMI ID 表中选择 ID 值，如下所示：

- 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）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）。
- c. 您为 EC2 实例创建的安全组的 ID。
- d. AWS 区域。本演练使用 us-west-2 区域。
- e. 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该值。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 名称时需要该值，用于将文件系统挂载到 EC2 实例上。

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

```
$ aws ec2 run-instances \  
--image-id AMI ID \  
--count 1 \  
--instance-type t2.micro \  
--associate-public-ip-address \  
--profile adminuser
```

```
--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
```

- 记下 `run-instances` 命令返回的实例 ID。
- 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 [开始前的准备工作](#) (p. 110)。

- 可为创建的 EC2 实例分配名称，方法是添加一个标签，将键名和值设置为要分配给该实例的名称。运行以下 AWS CLI `create-tags` 命令。

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

下一步

[步骤 2：创建 Amazon EFS 资源](#) (p. 115)

## 步骤 2：创建 Amazon EFS 资源

在此步骤中，您将执行以下操作：

- 创建 Amazon EFS 文件系统。
- 在启动了 EC2 实例的可用区创建挂载目标。

主题

- [步骤 2.1：创建 Amazon EFS 文件系统](#) (p. 115)
- [步骤 2.2：创建挂载目标](#) (p. 117)

### 步骤 2.1：创建 Amazon EFS 文件系统

在该步骤中，您将创建一个 Amazon 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 `Name` 标签。

- a. 在命令提示符处，运行以下命令 AWS CLI `create-file-system` 命令。

```
$ aws efs create-file-system \
--creation-token FileSystemForWalkthrough1 \
--region us-west-2 \
--profile adminuser
```

- b. 通过调用 `describe-file-systems` CLI 命令验证文件系统是否已创建。

```
$ aws efs describe-file-systems \
--region us-west-2 \
--profile adminuser
```

以下是示例响应：

```
{
  "FileSystems": [
    {
      "SizeInBytes": {
        "Timestamp": 1418062014.0,
        "Value": 1024
      },
      "CreationToken": "FileSystemForWalkthrough1",
      "CreationTime": 1418062014.0,
      "FileSystemId": "fs-cda54064",
      "PerformanceMode": "generalPurpose",
      "NumberOfMountTargets": 0,
      "LifeCycleState": "available",
      "OwnerId": "account-id"
    }
  ]
}
```

- c. 记下 `FileSystemId` 的值。在下一步中为该文件系统创建挂载目标时需要该值。
2. (可选) 使用 `create-tag` CLI 命令向您创建的文件系统添加一个标签。

虽然不需要为文件系统创建标签也可完成本演练，但既然您在探索 Amazon EFS API，我们可以测试用于创建和管理标签的 Amazon EFS API。有关更多信息，请参阅 [CreateTags \(p. 171\)](#)。

- a. 添加标签。

```
$ aws efs create-tags \
--file-system-id File-System-ID \
--tags Key=Name,Value=SomeExampleNameValue \
--region us-west-2 \
--profile adminuser
```

- b. 使用 `describe-tags` CLI 命令检索添加到文件系统的标签列表。

```
$ aws efs describe-tags \
--file-system-id File-System-ID \
--region us-west-2 \
--profile adminuser
```

Amazon EFS 在响应正文中返回标签列表。

```
{
  "Tags": [
    {
```

```
        "Value": "SomeExampleNameValue",  
        "Key": "Name"  
      }  
    ]  
  }  
}
```

## 步骤 2.2：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](#) 中在其中启动了 EC2 实例的 VPC 子网 ID。

在本演练中，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
```

您将获得此响应：

```
{  
  "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
```

下一步

[步骤 3：将 Amazon EFS 文件系统挂载到 EC2 实例上并测试 \(p. 117\)](#)

## 步骤 3：将 Amazon EFS 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：

#### 主题

- [步骤 3.1：收集信息](#) (p. 118)
  - [步骤 3.2：在 EC2 实例上安装 NFS 客户端](#) (p. 118)
  - [步骤 3.3：将文件系统挂载到 EC2 实例上并测试](#) (p. 119)
  - [下一步](#) (p. 120)
- 
- 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设置。

## 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 Amazon 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、Mac OS 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 EC2 实例：
  - 要从运行 Mac OS 或 Linux 的计算机连接到您的实例，需要使用 `-i` 选项和私有密钥的路径，将 `.pem` 文件指定给 `ssh` 命令。
  - 要从运行 Windows 的计算机连接到您的实例，可以使用 MindTerm 或 PuTTY。如果您计划使用 PuTTY，则需要安装它并按以下过程将 `.pem` 文件转换为 `.ppk` 文件。

有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的以下主题：

- [使用 PuTTY 从 Windows 连接到 Linux 实例](#)
  - [使用 SSH 连接到 Linux 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 Amazon 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 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. 创建一个目录 ("efs-mount-point")。

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

2. 挂载 Amazon EFS 文件系统。

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

3. 您已经将 Amazon 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. 创建一个文本文件。

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

- e. 列出目录的内容。

```
$ ls -l
```

现在，您已成功创建一个 Amazon 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。有关更多信息，请参阅[重启时自动重新挂载 \(p. 127\)](#)。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。有关示例，请参阅[演练 2：设置 Apache Web 服务器并提供 Amazon EFS 文件服务 \(p. 121\)](#)。

## 下一步

[步骤 4：清除 \(p. 120\)](#)

## 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。

- 删除 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 Amazon EFS 资源 (文件系统、挂载目标)。

### 删除本演练中创建的 AWS 资源

1. 终止为本演练创建的 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 Amazon EC2 用户指南（适用于 Linux 实例）中的[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 用户指南（适用于 Linux 实例）中的[删除安全组](#)。

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
```

## 演练 2：设置 Apache Web 服务器并提供 Amazon EFS 文件服务

您可能运行 Apache Web 服务器的 EC2 实例，为 Amazon EFS 文件系统上存储的文件提供服务。它可以是一个 EC2 实例，如果您的应用程序需要，您也可以有多个 EC2 实例为您的 Amazon EFS 文件系统中的文件提供服务。下面的过程描述了具体操作步骤。

- 在 EC2 实例上设置 Apache Web 服务器 (p. 121)。
- 通过创建 Auto Scaling 组，在多个 EC2 实例上设置 Apache Web 服务器 (p. 123)。您可以使用 Amazon EC2 Auto Scaling 创建多个 EC2 实例，它是一种 AWS 服务，可以根据您应用程序的需要增加或减少组中的 EC2 实例数。当您有多个 Web 服务器时，还需要一个负载均衡器在它们之间分布请求流量。

### Note

对于这两个过程，您将在 美国西部（俄勒冈）区域 (us-west-2) 中创建所有资源。

## 提供文件的单个 EC2 实例

按照以下步骤在一个 EC2 实例上设置 Apache Web 服务器，以便为您在 Amazon EFS 文件系统中创建的文件提供服务。

1. 按照入门练习中的步骤操作，以便您具有包含以下内容的可正常工作的配置：

- Amazon EFS 文件系统
- EC2 实例
- 文件系统挂载在 EC2 实例上

有关说明，请参阅 [Amazon Elastic File System 入门](#) (p. 9)。在执行这些步骤时，请记住以下内容：

- EC2 实例的公有 DNS 名称。

- 在启动 EC2 实例的同一可用区中创建的挂载目标的公有 DNS 名称。
2. (可选) 您可以选择从入门练习中创建的挂载点卸载文件系统。

```
$ sudo umount ~/efs-mount-point
```

在本演练中，您将为文件系统创建另一个挂载点。

3. 在您的 EC2 实例上，安装 Apache Web 服务器并进行如下配置：
  - a. 连接到 EC2 实例并安装 Apache Web 服务器。

```
$ sudo yum -y install httpd
```

- b. 启动服务。

```
$ sudo service httpd start
```

- c. 创建挂载点。

首先请注意，`/etc/httpd/conf/httpd.conf` 文件中的 `DocumentRoot` 指向 `/var/www/html` (`DocumentRoot "/var/www/html"`)。

您将 Amazon EFS 文件系统挂载在文档根目录下的子目录中。

- i. 在 `/var/www/html` 下创建子目录 `efs-mount-point`。

```
$ sudo mkdir /var/www/html/efs-mount-point
```

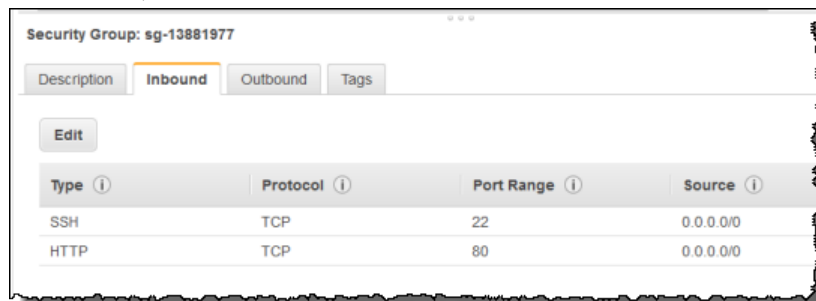
- ii. 挂载您的 Amazon EFS 文件系统。您需要通过提供您的文件系统 ID 和 AWS 区域来更新以下命令 (如果您按照入门练习创建文件系统，则入门练习假定为 `us-west-2` AWS 区域)。

```
$ 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:/ /var/www/html/efs-mount-point
```

在这里，您可以从您所在的 EC2 实例动态构建挂载目标的 DNS 名称。有关更多信息，请参阅[使用 DNS 名称在 Amazon EC2 上挂载 \(p. 222\)](#)。

4. 测试设置。
  - a. 在入门练习中创建的 EC2 实例安全组中添加规则，以允许 TCP 端口 80 上来自任何位置的 HTTP 流量。

添加规则后，EC2 实例安全组将具有以下入站规则。



有关说明，请参阅[使用 AWS 管理控制台创建安全组 \(p. 25\)](#)。

- b. 创建示例 `html` 文件。

- i. 更改目录。

```
$ cd /var/www/html/efs-mount-point
```

- ii. 为 `sampledir` 创建一个子目录并更改所有权。然后更改目录，以便可以在 `sampledir` 子目录中创建文件。

```
$ sudo mkdir sampledir
$ sudo chown ec2-user sampledir
$ sudo chmod -R o+r sampledir
$ cd sampledir
```

- iii. 创建示例 `hello.html` 文件。

```
$ echo "<html><h1>Hello from Amazon EFS</h1></html>" > hello.html
```

- c. 打开浏览器窗口并输入访问该文件的 URL (它是 EC2 实例的公有 DNS 名称，后跟文件名)。例如：

```
http://EC2-instance-public-DNS/efs-mount-point/sampledir/hello.html
```

现在，您正在为存储在 Amazon EFS 文件系统上的网页提供服务。

#### Note

此设置不会将 EC2 实例配置为在引导时自动启动 `httpd` (Web 服务器)，也不会引导时挂载文件系统。在下一个演练中，您将创建一个启动配置来进行这种设置。

## 提供文件服务的多个 EC2 实例

按照以下步骤从多个 EC2 实例中的 Amazon EFS 文件系统提供相同的内容，以提高可扩展性或可用性。

1. 按照[入门 \(p. 9\)](#)练习中的步骤操作，以便创建并测试 Amazon EFS 文件系统。

#### Important

对于本演练，您不使用在入门练习中创建的 EC2 实例，而是启动新的 EC2 实例。

2. 使用以下步骤在 VPC 中创建负载均衡器。

1. 定义负载均衡器

在基本配置部分中，选择您的 VPC，您还会在其中创建 EC2 实例以挂载文件系统。

在选择子网部分中，您可以选择所有可用的子网，或选择 。有关详细信息，请参阅下一节中的 `cloud-config` 脚本。

2. 分配安全组

为负载均衡器创建一个新安全组，以允许从任何位置通过端口 80 进行 HTTP 访问权限，如下所示：

- Type : HTTP
- Protocol : TCP
- Port Range : 80
- Source : Anywhere (0.0.0.0/0)

## Note

一切就绪后，您还可以更新 EC2 实例安全组入站规则访问，以便仅允许来自负载均衡器的 HTTP 流量。

### 3. 配置运行状况检查

将 Ping 路径值设置为 `/efs-mount-point/test.html`。`efs-mount-point` 是您在其中挂载文件系统的子目录。在此过程的稍后步骤中您要向其中添加 `test.html` 页面。

## Note

请勿添加任何 EC2 实例。稍后，您将会创建 Auto Scaling 组，并在其中启动 EC2 实例和指定该负载均衡器。

有关创建负载均衡器的说明，请参阅 Elastic Load Balancing 用户指南 中的 [Elastic Load Balancing 入门](#)。

3. 创建包含两个 EC2 实例的 Auto Scaling 组。首先，您将创建一个描述实例的启动配置。然后，您可以通过指定启动配置来创建 Auto Scaling 组。以下步骤提供了您从 Amazon EC2 控制台中指定的用来创建 Auto Scaling 组的配置信息。
  - a. 从左侧导航窗格的 Auto Scaling 下面选择启动配置。
  - b. 选择创建 Auto Scaling 组以启动向导。
  - c. 选择 Create launch configuration。
  - d. 从快速启动中，选择最新版本的 Amazon Linux (HVM) AMI。这是您在入门练习的 [第 1 步：创建您的 EC2 资源并启动您的 EC2 实例](#) (p. 9) 中使用的同一 AMI。
  - e. 在高级部分中，执行以下操作：
    - 对于 IP 地址类型，请选择向每个实例分配公有 IP 地址。
    - 在用户数据框中，复制/粘贴以下脚本。

您必须通过提供 `file-system-id` 和 `aws-region` 的值来更新脚本 (如果您已按照入门练习操作，则已经在 `us-west-2` 区域创建了文件系统)。

在脚本中，注意以下方面：

- 该脚本会安装 NFS 客户端和 Apache Web 服务器。
- `echo` 命令在 `/etc/fstab` 文件中写入以下条目，以指定文件系统的 DNS 名称及其挂载子目录。此条目确保在每次系统重启后都挂载该文件。请注意，文件系统的 DNS 名称是动态构建的。有关更多信息，请参阅 [使用 DNS 名称在 Amazon EC2 上挂载](#) (p. 222)。

```
file-system-ID.efs.aws-region.amazonaws.com:/ /var/www/html/efs-mount-point  
nfs4 defaults
```

- 创建 `efs-mount-point` 子目录并在其中挂载文件系统。
- 创建 `test.html` 页面，以便 ELB 运行状况检查可以找到该文件 (在创建负载均衡器时，您将该文件指定为 Ping 点)。

有关用户数据脚本的更多信息，请参阅 Amazon EC2 用户指南 (适用于 Linux 实例) 中的 [添加用户数据](#)。

```
#cloud-config  
package_upgrade: true  
packages:  
- nfs-utils  
- httpd
```

```
runcmd:
- echo "$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-
zone).file-system-id.efs.aws-region.amazonaws.com:/ /var/www/html/efs-mount-
point nfs4 defaults" >> /etc/fstab
- mkdir /var/www/html/efs-mount-point
- mount -a
- touch /var/www/html/efs-mount-point/test.html
- service httpd start
- chkconfig httpd on
```

- f. 对于分配安全组，请选择选择一个现有的安全组，然后选择您为 EC2 实例创建的安全组。

在配置 Auto Scaling 组详细信息时，请使用以下信息：

1. 对于组大小，请选择# **2** #####。您将创建两个 EC2 实例。
2. 从 Network (网络) 列表中选择您的 VPC。
3. 选择在上一步中创建启动配置时，在用户数据脚本中指定挂载目标 ID 时使用的同一可用区中的子网。
4. 在“Advanced Details”部分
  - a. 对于负载均衡，请选择从弹性负载均衡器接收流量，然后选择您为本练习创建的负载均衡器。
  - b. 对于运行状况检查类型，请选择 ELB。

按照 Amazon EC2 Auto Scaling 用户指南 的[设置具有扩展和负载均衡功能的应用程序](#)中的说明创建 Auto Scaling 组。使用上述表格中的信息 (如果适用)。

4. 成功创建 Auto Scaling 组后，您将具有两个安装了 `nfs-utils` 和 Apache Web 服务器的 EC2 实例。在每个实例上，确认您具有已挂载 Amazon EFS 文件系统的 `/var/www/html/efs-mount-point` 子目录。有关如何连接到 EC2 实例的说明，请参阅[第 3 步：连接到您的 Amazon EC2 实例并挂载 Amazon EFS 文件系统](#) (p. 13)。

#### Note

如果在启动 Amazon 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 AMI 中。

5. 创建示例页面 (`index.html`)。
- a. 更改目录。

```
$ cd /var/www/html/efs-mount-point
```

- b. 为 `sampledir` 创建一个子目录并更改所有权。然后更改目录，以便可以在 `sampledir` 子目录中创建文件。如果您已按照前面的[提供文件的单个 EC2 实例](#) (p. 121) 操作，则您已经创建了 `sampledir` 子目录，因此，可以跳过该步骤。

```
$ sudo mkdir sampledir
$ sudo chown ec2-user sampledir
$ sudo chmod -R o+r sampledir
$ cd sampledir
```

- c. 创建示例 `index.html` 文件。

```
$ echo "<html><h1>Hello from Amazon EFS</h1></html>" > index.html
```

6. 现在，您可以测试设置。使用负载均衡器的公有 DNS 名称访问 `index.html` 页面。

```
http://load balancer public DNS Name/efs-mount-point/sampledir/index.html
```

负载均衡器向某个运行 Apache Web 服务器的 EC2 实例发送请求。然后，Web 服务器将为存储在您的 Amazon EFS 文件系统中的文件提供服务。

## 演练 3：创建可写的每用户子目录以及配置在重启时自动重新挂载

创建 Amazon EFS 文件系统并将它本地挂载到您的 EC2 实例上后，它会公开一个称为 `#####` 的空目录。一个常用案例是，在这个“文件系统根目录”下为您在 EC2 实例上创建的每个用户创建一个“可写”子目录，并将它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 Amazon EFS 文件系统中创建。

在本演练中，您将首先在您的 EC2 实例上创建用户“mike”。然后，将一个 Amazon EFS 子目录挂载到用户 mike 的主目录上。本演练还将阐释如何配置在系统重启时自动重新挂载子目录。

假设您创建了一个 Amazon EFS 文件系统并挂载到 EC2 实例上的一个本地目录中。姑且称之为 `EFSroot`。

### Note

您可以按照[入门 \(p. 9\)](#)练习创建一个 Amazon EFS 文件系统并将其挂载到 EC2 实例上。

在以下步骤中，您将创建用户 mike，为他创建一个子目录 (`EFSroot/mike`)，将他设为该子目录的所有者，授予他完全权限，最后在他的主目录 (`/home/mike`) 上挂载 Amazon EFS 子目录。

#### 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 mike 并分配密码。

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

这也将为该用户创建一个主目录 `/home/mike`。

#### 2. 在 `EFSroot` 下为用户 mike 创建一个子目录：

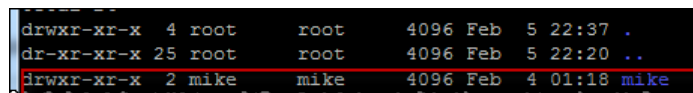
- 在 `EFSroot` 下创建子目录 mike。

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

您需要将 `EFSroot` 替换为您的本地目录名称。

- 根用户和根组都是 `/mike` 子目录的所有者 (可以使用 `ls -l` 命令来验证)。要为用户 mike 授予对子目录的完全权限，可授予 mike 对该目录的所有权。

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



```
drwxr-xr-x 4 root root 4096 Feb 5 22:37 .
drwxr-xr-x 25 root root 4096 Feb 5 22:20 ..
drwxr-xr-x 2 mike mike 4096 Feb 4 01:18 mike
```

#### 3. 使用 `mount` 命令将 `EFSroot/mike` 子目录挂载到 mike 的主目录中。

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

`mount-target-DNS` 地址标识远程 Amazon EFS 文件系统根目录。

现在，用户 mike 的主目录就是 Amazon EFS 文件系统内 mike 可以写入的一个子目录。如果卸载此挂载目标，用户将无法访问其 EFS 目录，除非重新挂载，而这需要根权限。

## 重启时自动重新挂载

您可以使用 `fstab` 文件实现在每次系统重启后都自动重新挂载您的文件系统。有关更多信息，请参阅 [自动挂载 Amazon EFS 文件系统 \(p. 61\)](#)。

## 演练 4：Amazon EFS 文件系统的备份解决方案

如果您需要能够从 Amazon EFS 文件系统的意外更改或删除中恢复，我们建议您使用 [EFS 到 EFS 备份解决方案](#)。

EFS 到 EFS 备份解决方案适用于所有 AWS 区域中的所有 Amazon EFS 文件系统。它包括一个 AWS CloudFormation 模板以启动、配置和运行部署该解决方案所需的 AWS 服务。该解决方案遵循 AWS 的安全和可用性最佳实践。

有关 EFS 到 EFS 备份解决方案的更多信息，请参阅 AWS Answers 中的 [EFS 到 EFS 备份解决方案](#)。

### Note

在具有 EFS 到 EFS 备份解决方案之前，我们建议您使用仅适用于具有 AWS Data Pipeline 支持的 AWS 区域的替代备份解决方案。有关以前的解决方案的更多信息，请参阅 [使用 AWS Data Pipeline 备份 Amazon EFS 文件系统 \(p. 209\)](#)。

## 演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统

本演练使用 AWS 管理控制台通过 AWS Direct Connect 连接在本地客户端上创建和挂载文件系统。

### Note

不支持将 Amazon EFS 与基于 Microsoft Windows 的客户端一起使用。

在本演练中，我们假定您已具有 AWS Direct Connect 连接。如果没有该连接，您可以立即开始建立连接，并在建立连接后返回到本演练。有关更多信息，请参阅 [AWS Direct Connect 产品详细信息](#)。

在具有 AWS Direct Connect 连接时，您创建一个 Amazon EFS 文件系统，并在 Amazon VPC 中创建一个挂载目标。然后，您下载并安装 `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。

本演练在美国西部（俄勒冈）区域（`us-west-2`）创建所有这些资源。不论您使用哪个 AWS 区域，请确保使用方式一致。您的所有资源（VPC、挂载目标和 Amazon EFS 文件系统）必须位于同一个 AWS 区域。

### Note

如果本地应用程序需要知道 EFS 文件系统是否可用，在第一个挂载点暂时不可用时，您的应用程序应该能够指向不同的挂载点 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（AZ）连接到您的文件系统，以提供更高的可用性。



## 开始前的准备工作

您可以使用 AWS 账户的根凭证登录到控制台并尝试此练习。但是，AWS Identity and Access Management (IAM) 最佳实践建议您不要使用您的 AWS 账户的根凭证，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 [设置 \(p. 7\)](#)。

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 Internet 网关。
- 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 Amazon Elastic File System 资源

在该步骤中，您将创建 Amazon EFS 文件系统和挂载目标。

### 创建 Amazon EFS 文件系统

1. 在 <https://console.aws.amazon.com/efs/> 处打开 Amazon EFS 控制台。
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 [创建挂载目标 \(p. 20\)](#)。
5. 选择 Next Step。
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。

您刚创建的文件系统包含步骤 1.4 中创建的挂载目标。每个挂载目标都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 则将 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。

接下来，您在挂载目标的安全组中添加一个规则，以允许将入站流量传输到 NFS 端口 (2049)。您可以使用 AWS 管理控制台将该规则添加到您的挂载目标在 VPC 中的安全组。

### 允许入站流量进入 NFS 端口

1. 登录 AWS 管理控制台并通过以下网址打开 Amazon EC2 控制台 <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 Amazon Elastic File System 资源 \(p. 128\)](#) 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 Edit。
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 端口上打开 TCP 连接，从而将挂载目标安全组指定为目标。

## 步骤 2：下载并安装 amazon-efs-utils

amazon-efs-utils 软件包是一个开源 Amazon EFS 工具集并附带提供了挂载帮助程序和一些工具，从而为 Amazon EFS 轻松加密传输中的数据。有关更多信息，请参阅 [使用 amazon-efs-utils 工具 \(p. 34\)](#)。可以从 GitHub 中免费下载该软件包，您可以克隆该软件包的存储库从获取该软件包。

从 GitHub 中克隆 amazon-efs-utils

1. 访问本地客户端的终端。
2. 从终端中，使用以下命令将 amazon-efs-utils 工具从 GitHub 克隆到所选的目录中。

```
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
- 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

作为 RPM 软件包构建并安装 amazon-efs-utils

1. 在客户端上打开一个终端，然后导航到具有从 GitHub 克隆的 amazon-efs-utils 软件包的目录。
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，您需要使用以下命令进行安装：

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

3. 使用以下命令安装该软件包：

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

作为 deb 软件包构建并安装 amazon-efs-utils

1. 在客户端上打开一个终端，然后导航到具有从 GitHub 克隆的 amazon-efs-utils 软件包的目录。
2. 使用以下命令构建该软件包：

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

3. 使用以下命令安装该软件包：

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

在安装该软件包后，请配置 amazon-efs-utils 以在具有 AWS Direct Connect 的 AWS 区域中使用。

配置 amazon-efs-utils 以在您的 AWS 区域中使用

1. 使用所选的文本编辑器打开 `/etc/amazon/efs/amazon-efs-utils.conf` 以进行编辑。
2. 查找 `"dns_name_format = {fs_id}.efs.{region}.amazonaws.com"` 行。
3. 使用您的 AWS 区域的 ID 更改 `{region}`，例如，`us-west-2`。

现已安装并配置 amazon-efs-utils 软件包，您可以将文件系统挂载到本地客户端中。

## 步骤 3：在本地客户端上挂载 Amazon EFS 文件系统

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、挂载目标 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：

1. 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。
2. 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
```

3. 创建一个将文件系统挂载到的本地目录。

Example

```
mkdir ~/efs
```

4. 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
```

现已挂载 Amazon EFS 文件系统，您可以使用以下过程对其进行测试。

## 测试 Amazon 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
```

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 [自动挂载 Amazon EFS 文件系统 \(p. 61\)](#)。

### 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 \(p. 99\)](#)。

## 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 Amazon EFS 文件系统。

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

2. 在 <https://console.aws.amazon.com/efs/> 处打开 Amazon EFS 控制台。
3. 选择要从文件系统列表中删除的 Amazon EFS 文件系统。
4. 对于 Actions，选择 Delete file system。
5. 在永久删除文件系统对话框中，键入要删除的 Amazon EFS 文件系统的文件系统 ID，然后选择删除文件系统。
6. 打开 Amazon EC2 控制台 <https://console.aws.amazon.com/ec2/>。
7. 在导航窗格中，选择 Security Groups。
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。

## 演练 6：在 Amazon EFS 文件系统中实施静态加密

您可以在下文中找到有关如何使用 Amazon CloudWatch 和 AWS CloudTrail 实施静态加密的详细信息。本演练基于 AWS 白皮书[使用 Amazon EFS 加密文件系统静态加密数据](#)。

### Note

目前，您无法实施传输中加密。

## 实施静态加密

您的组织可能要求静态加密符合特定分类条件的所有数据，或者静态加密与特定应用程序、工作负载或环境关联的所有数据。您可以使用检测性控制为 Amazon EFS 文件系统实施静态数据加密策略。这些控制检测创建的文件系统，并验证是否启用了静态加密。

如果检测到没有静态加密的文件系统，您可以通过多种方法进行响应。这些方法包括删除文件系统和挂载目标以及通知管理员。

如果要删除未静态加密的文件系统，但希望保留数据，请先创建新的静态加密的文件系统。然后，将数据复制到新的静态加密的文件系统。在复制数据后，您可以删除未静态加密的文件系统。

## 检测未静态加密的文件系统

您可以创建 CloudWatch 警报以监控 CloudTrail 日志中的 `CreateFileSystem` 事件。然后，您可以触发警报，以便在创建未静态加密的文件系统时通知管理员。

## 创建指标筛选条件

要创建一个在创建未加密的 Amazon EFS 文件系统时触发的 CloudWatch 警报，请使用以下过程。

在开始之前，您必须创建了一个跟踪以将 CloudTrail 日志发送到 CloudWatch Logs 日志组。有关更多信息，请参阅 AWS CloudTrail User Guide 中的[将事件发送到 CloudWatch Logs](#)。

### 创建指标筛选条件

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Logs。
3. 在日志组列表中，选择为 CloudTrail 日志事件创建的日志组。
4. 选择 Create Metric Filter。
5. 在定义日志指标筛选条件页上，选择筛选模式，然后键入以下内容：

```
{ ($.eventName = CreateFileSystem) && ($.responseElements.encrypted IS FALSE) }
```

6. 选择 Assign Metric。
7. 对于筛选条件名称，请键入 **UnencryptedFileSystemCreated**。

8. 对于 Metric Namespace，请键入 **CloudTrailMetrics**。
9. 对于指标名称，请键入 **UnencryptedFileSystemCreatedEventCount**。
10. 选择 Show advanced metric settings。
11. 对于 Metric Value，请键入 **1**。
12. 选择 Create Filter。

## 创建警报

在创建指标筛选条件后，请使用以下过程创建一个警报。

### 创建警报

1. 在 Log\_Group\_Name 页面的筛选条件上，在 UnencryptedFileSystemCreated 筛选条件名称旁边选择创建警报。
2. 在创建警报页上，设置以下参数：
  - 对于名称，请键入 **Unencrypted File System Created**。
  - 对于每当，请执行以下操作：
    - 将是设置为 **> = 1**。
    - 将对于: 设置为 **1** 个连续时间段。
  - 对于将缺失的数据作为以下内容处理，请选择好 (未超出阈值)。
  - 对于操作，请执行以下操作：
    - 对于每当此警报，请选择状态为“警报”。
    - 对于发送通知到，选择 NotifyMe，选择新建列表，然后为该列表键入唯一的主题名称。
    - 对于电子邮件列表，请键入要将通知发送到的电子邮件地址。将会通过该地址接收一封电子邮件，以确认创建了该警报。
  - 对于警报预览，请执行以下操作：
    - 对于周期，请选择 1 分钟。
    - 对于统计数据，请选择标准和总计。
3. 选择 Create Alarm。

## 测试创建未加密的文件系统的警报

您可以创建未静态加密的文件系统以测试警报，如下所示。

### 创建未静态加密的文件系统以测试警报

1. 打开 Amazon EFS 控制台 (<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. 选择所有可用区。确保选择了默认子网、自动 IP 地址和默认安全组。这些是您的挂载目标。
5. 选择 Next Step。
6. 命名您的文件系统，并取消选中启用加密以创建未加密的文件系统。
7. 选择 Next Step。
8. 选择创建文件系统。

您的跟踪记录 CreateFileSystem 操作，并将事件传送到您的 CloudWatch Logs 日志组。该事件会触发您的指标警报，而 CloudWatch Logs 会向您发送有关相应更改的通知。

## 演练 7：使用 EFS 文件同步将文件从本地文件系统同步到 Amazon EFS

本演练说明了如何使用 EFS 文件同步将文件从本地文件系统同步到 Amazon EFS。

### 主题

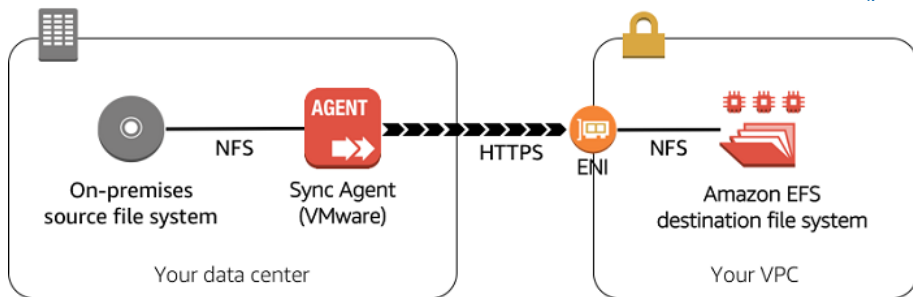
- [开始前的准备工作](#) (p. 134)
- [步骤 1：创建同步代理](#) (p. 134)
- [步骤 2：创建同步任务](#) (p. 135)
- [步骤 3：将源文件系统同步到 Amazon EFS](#) (p. 137)
- [步骤 4：访问文件](#) (p. 138)
- [第 5 步：清除](#) (p. 138)

## 开始前的准备工作

在本演练中，我们假定：

- 在本地数据中心具有一个网络文件系统 (NFS) 文件服务器。
- 在本地数据中心具有一个 VMware ESXi 管理程序主机。
- 您已创建一个 Amazon EFS 文件系统。如果没有 Amazon EFS 文件系统，请立即创建一个文件系统，并在完成后返回到本演练。有关如何创建 Amazon EFS 文件系统的更多信息，请参阅[Amazon Elastic File System 入门](#) (p. 9)。

您可以通过 Internet 安全高效地复制文件或使用 AWS Direct Connect。有关如何使用 AWS Direct Connect 的信息，请参阅[演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统](#) (p. 127)。



## 步骤 1：创建同步代理

要创建同步代理，请下载一个虚拟机 (VM) 映像并将其部署到本地环境中，以便它可以挂载您的源文件系统。在部署后，您激活代理以安全地将其与您的 AWS 账户相关联。

为本地数据创建同步代理

1. 打开 Amazon EFS 管理控制台 (<https://console.aws.amazon.com/efs/>)。
2. 选择文件同步。如果尚未在该 AWS 区域中使用 EFS 文件同步，将会看到一个介绍页面。选择开始使用以打开选择主机平台页。

如果以前在该 AWS 区域中使用了 EFS 文件同步，请从左侧导航窗格中选择代理，然后选择创建同步代理以打开选择主机平台页。

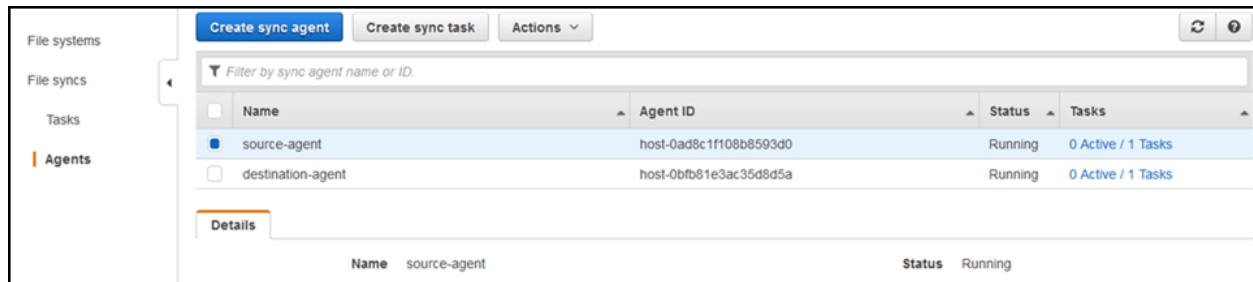


3. 从选择主机平台页中，选择 VMware ESXi，然后选择下载映像。将开始下载虚拟机 (VM) 映像。
4. 在下载完成后，将虚拟机部署到 VMware ESXi 管理程序中，然后使用 VMware 客户端配置虚拟机。我们建议您使用具有 4 个 vCPU、32 GB 内存、10 Gb 网络以及 80 GB 根卷的虚拟机。
5. 启动虚拟机，然后记下虚拟机 IP 地址。该虚拟机必须能够使用 NFS 挂载源文件系统。

#### Note

虽然不是必需的，但我们建议您将半虚拟化网络控制器用于 VMware ESXi 虚拟机。  
您不需要在虚拟机中添加额外的磁盘。EFS 文件同步仅使用根磁盘。

6. 在 Amazon EFS 控制台上，选择下一步：连接到代理。
7. 对于 IP 地址，请键入虚拟机的 IP 地址，然后选择下一步：激活代理。您的浏览器将连接到该 IP 地址，以便从同步代理中获取唯一的激活密钥。该密钥安全地将同步代理与您的 AWS 账户相关联。不需要能够从您的网络外部访问该 IP 地址，但必须能够从浏览器中访问该 IP 地址。
8. 在激活代理页上，键入同步代理的名称，然后选择激活代理。



此时，将会在 Amazon EFS 控制台上看到激活的同步代理。

## 步骤 2：创建同步任务

创建一个同步任务，并配置源和目标文件系统。

### 创建同步任务

1. 选择创建同步任务。将显示配置源位置页。

**Create sync task**

**Configure source**

Configure destination

Configure settings

Review

**Configure source location**

Specify the location that you want to copy from.

NFS server: 192.0.2.0

Mount Path: /source-path

Agent: Syn-agent-op (host-08ca202c534bbadee)

Cancel Next: Configure destination

2. 为源文件系统提供以下信息：
  - 对于 NFS 服务器，请键入源 NFS 服务器的域名或 IP 地址。
  - 对于挂载路径，请键入源文件系统的挂载路径。
  - 对于代理，请选择您以前创建的同步代理。
3. 选择下一步：配置目标。将显示配置目标位置页。

The screenshot shows the 'Create sync task' console page. On the left, a sidebar contains links for 'Configure source', 'Configure destination' (which is highlighted with an orange bar), 'Configure settings', and 'Review'. The main area is titled 'Configure destination location' with the instruction 'Specify the location that you want to copy into.' Below this, there are three fields: 'Amazon EFS file systems' with a dropdown menu showing 'fs-01bf5478', 'File system path' with a text input field containing '/', and 'Security group' with a dropdown menu showing 'sg-2... (default)'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next: Configure settings'.

4. 为目标文件系统提供以下信息：

- 对于 Amazon EFS 文件系统，请选择要同步到的 EFS 文件系统。如果没有 Amazon EFS 文件系统，请立即创建一个文件系统，并在完成后重新启动本演练。有关如何创建 Amazon EFS 文件系统的更多信息，请参阅[Amazon Elastic File System 入门 \(p. 9\)](#)。
- 对于文件系统路径，请键入要将数据写入到的文件系统路径。该路径必须位于目标文件系统中。
- 对于安全组，请选择一个允许访问选定的目标 Amazon EFS 文件系统的安全组。

5. 选择下一步：配置设置。将显示配置同步设置页。

The screenshot shows the 'Create sync task' console page, specifically the 'Configure sync settings' step. The sidebar on the left now highlights 'Configure settings'. The main area is titled 'Configure sync settings' with the instruction 'Settings to use when synchronizing your files'. It contains three sections of settings, each with a checkbox and a label: 'Copy file metadata' with checkboxes for 'Ownership', 'Permissions', and 'Timestamps'; 'File deletion' with a checkbox for 'Keep files in destination even if not found in source'; and 'Verification mode' with a checkbox for 'Full consistency between source and destination file systems'. All checkboxes are checked. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

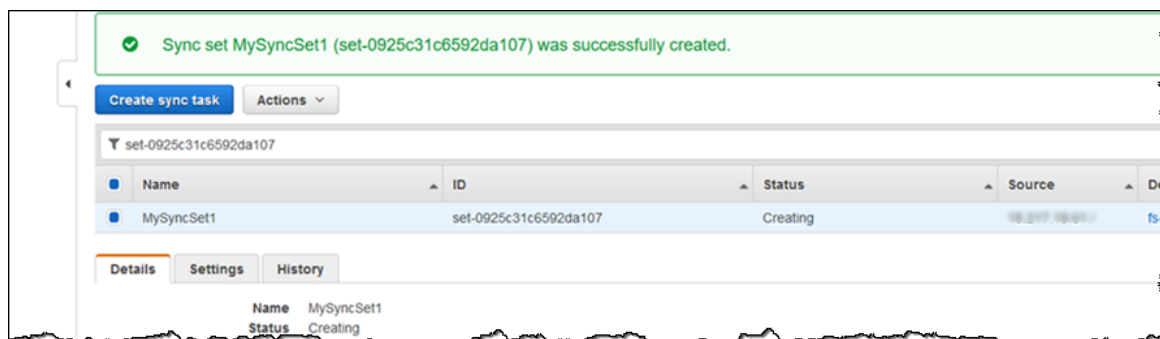
6. 配置您希望该同步任务在同步文件时使用的默认设置：

#### Note

您可以在以后启动同步任务时覆盖这些设置。

- 选择所有权 (用户/组 ID) 以从源文件中复制用户和组 ID。
- 选择权限以复制源文件权限。
- 选择时间戳以从源文件中复制时间戳。
- 选择文件删除以在目标中保留在源文件系统中未找到的所有文件。如果清除该框，则在目标中删除在源文件系统中未找到的所有文件。
- 选择验证模式以在同步任务完成后检查目标文件系统是否为源文件系统的精确副本。如果未选择该选项，则仅验证传输的数据。不会查找在主动传输文件时对其进行的更改以及对未主动传输的文件进行的更改。我们建议您选择完整验证。

7. 选择下一步：审核和创建，然后查看您的同步任务设置。在准备就绪时，选择创建同步任务。



将创建您的同步任务。如果已挂载源和目标文件系统，任务状态将显示为可用。

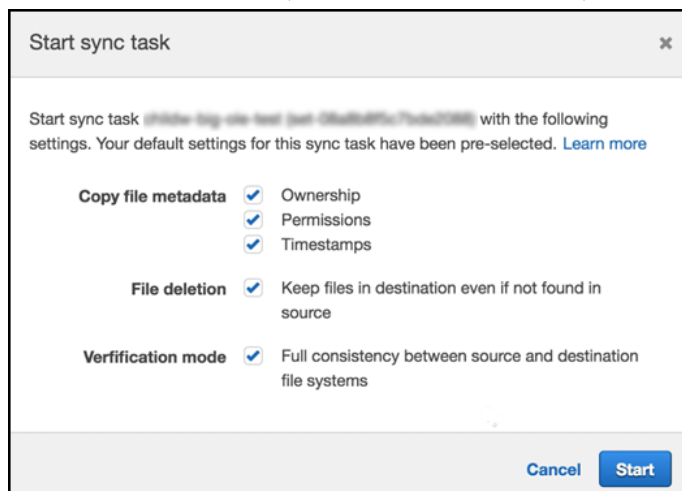
详细信息选项卡显示源和目标文件系统的状态和设置。

## 步骤 3：将源文件系统同步到 Amazon EFS

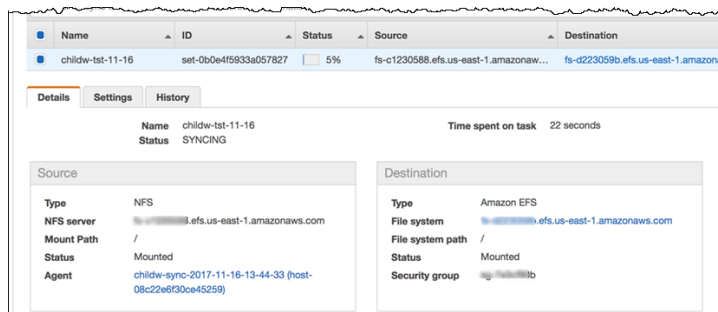
现在，您具有一个同步任务，您可以启动同步任务以开始将文件从源文件系统同步到目标 Amazon EFS 文件系统。

### 同步源文件系统

1. 在“任务”页中，选择刚创建的同步任务。“详细信息”选项卡将显示同步任务的状态。
2. 在操作菜单中，选择启动。
3. 在启动同步任务对话框中，您可以修改同步任务设置，然后选择启动。



4. 选择启动以开始同步文件。



5. 在启动同步任务时，状态列将显示同步任务进度。在同步任务开始准备时，状态将从正在启动变为正在准备。在任务开始同步文件时，状态将从正在准备变为正在同步。在启动文件一致性验证时，状态将变为正在验证。在同步任务完成后，状态将变为成功。

## 步骤 4：访问文件

要访问您的文件，请从 Amazon EC2 实例连接到 Amazon EFS 文件系统或使用 AWS Direct Connect。

有关如何使用 Amazon EC2 进行连接的信息，请参阅[连接到 Amazon EC2 实例和挂载 Amazon EFS 文件系统](#)。

有关如何使用 AWS Direct Connect 进行连接的信息，请参阅[演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 \(p. 127\)](#)。

## 第 5 步：清除

如果不再需要使用创建的资源，应将其删除：

- 删除您创建的任务。有关更多信息，请参阅[删除同步任务 \(p. 48\)](#)。
- 删除您创建的同步代理。这不会删除部署到本地管理程序的虚拟机。
- 清理您创建的 Amazon EFS 资源。有关更多信息，请参阅[步骤 5：清理资源并保护您的 AWS 账户 \(p. 15\)](#)。

# 演练 8：使用 EFS 文件同步将文件系统从 Amazon EC2 同步到 Amazon EFS

本演练说明了如何使用 EFS 文件同步将文件从 AWS 中的文件系统同步到 Amazon EFS 的步骤。

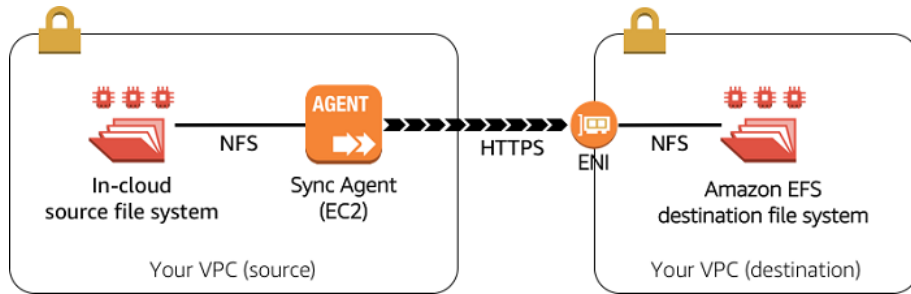
### 主题

- [开始前的准备工作 \(p. 138\)](#)
- [步骤 1：创建同步代理 \(p. 139\)](#)
- [步骤 2：创建同步任务 \(p. 140\)](#)
- [步骤 3：将源文件系统同步到 Amazon EFS \(p. 142\)](#)
- [步骤 4：访问文件 \(p. 143\)](#)
- [步骤 4：清除 \(p. 143\)](#)

## 开始前的准备工作

在本演练中，我们假定：

- 在 Amazon EC2 实例上具有一个网络文件系统 (NFS) 文件服务器。
- 您已创建一个 Amazon EFS 文件系统。如果没有 Amazon EFS 文件系统，请立即创建一个文件系统，并在完成后返回到本演练。有关如何创建 Amazon EFS 文件系统的更多信息，请参阅[Amazon Elastic File System 入门 \(p. 9\)](#)。



## 步骤 1：创建同步代理

要在 Amazon EC2 中创建同步代理，您可以使用提供的 AMI 创建一个 Amazon EC2 实例，该实例可以在您的 AWS 环境中挂载源文件系统。该 Amazon EC2 实例将在与源文件系统相同的 AWS 区域中运行。在部署后，您激活代理以安全地将其与 AWS 账户相关联。

为 AWS 中的数据创建同步代理

1. 打开 Amazon EFS 管理控制台 (<https://console.aws.amazon.com/efs/>)，然后选择创建了源文件系统的 AWS 区域。
2. 选择文件同步。如果尚未在 AWS 区域中使用 EFS 文件同步，将会看到一个介绍页面。选择开始使用以打开选择主机平台页。

如果以前在该 AWS 区域中使用了 EFS 文件同步，请从左侧导航窗格中选择代理，然后选择创建同步代理以打开选择主机平台页。

3. 从选择主机平台页中，选择 Amazon EC2，选择源文件系统所在的 AWS 区域，然后选择启动实例。在该 AWS 区域的 Amazon EC2 管理控制台中，将重定向到选择一个实例类型页，您可以在其中选择一种实例类型。

### Note

同步代理将文件同步到激活了同步代理的 AWS 区域中的 EFS 文件系统。将为该实例应用标准 Amazon EC2 费率。

4. 在 Choose an Instance Type 页面上，选择实例的硬件配置。在 Amazon EC2 上部署同步代理时，我们建议为您的同步代理选择内存优化实例类型之一。选择的实例大小必须至少为 xlarge。
5. 选择 Next: Configure Instance Details。
6. 在 Configure Instance Details 页面上，选择 Auto-assign Public IP 的值。如果希望能够从公共 Internet 中访问您的实例，请将自动分配公有 IP 设置为启用。否则，将自动分配公有 IP 设置为禁用。

Step 3: Configure Instance Details

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: ☐ Request Spot instances

Network: vpc-02824b63 (default) [Create new VPC](#)

Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP: **Use subnet setting (Enable)**

IAM role: None [Create new IAM role](#)

Shutdown behavior: Stop

Enable termination protection: ☐ Protect against accidental termination

Monitoring: ☐ Enable CloudWatch detailed monitoring  
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance  
Additional charges will apply for dedicated tenancy.

7. 选择下一步：添加存储，然后选择下一步：添加标签。EFS 文件同步代理使用根卷，而不需要使用额外的存储。
8. 在添加标签页上，您可以选择为实例添加标签。然后选择 Next: Configure Security Group。

- 在配置安全组页上，为传输到您的实例的特定流量添加防火墙规则。您可以创建新安全组或者选择现有安全组。

#### Important

至少，您的安全组必须允许从您的 Web 浏览器到 HTTP 端口 80 的入站访问以激活您的同步代理。

- 选择审核和启动以检查您的配置，然后选择启动以启动您的实例。我们建议您为您的实例选择现有的密钥对或创建新的密钥对。EFS 文件同步不需要使用该密钥对即可正常运行，但在与 AWS 联系以获得支持时可能需要使用该密钥对。

将显示一个确认页面，以指出您的实例正在启动。

- 选择 View Instances 以关闭确认页面并返回控制台。在 Instances (实例) 屏幕上，您可以查看您实例的状态。启动实例只需很短的时间。启动实例时，其初始状态为 pending (待处理)。在实例启动后，其状态将变为正在运行，并为其分配一个公有 DNS 名称和 IP 地址。
- 选择您的实例，并记下描述选项卡中的公有 IP 地址。您将使用该 IP 地址连接到您的同步代理。

#### Note

不需要能够从您的网络外部访问该 IP 地址。

#### Important

如果源文件系统和目标 Amazon EFS 文件系统位于不同的 AWS 区域中，您可以在目标 Amazon EFS 文件系统所在的 AWS 区域中打开 Amazon EFS 控制台以进行连接。

- 选择文件同步，选择创建同步代理，然后在选择主机平台页上选择下一步：连接到代理。
- 对于 IP 地址，请键入 Amazon EC2 实例 IP 地址，然后选择下一步：激活代理。您的浏览器将连接到该 IP 地址，以便从同步代理中获取唯一的激活密钥。该密钥安全地将同步代理与您的 AWS 账户相关联。不需要能够从您的网络外部访问该 IP 地址，但必须能够从浏览器中访问该 IP 地址。
- 在激活代理页上，键入您的同步代理的名称，然后选择激活代理。

此时，将会在 Amazon EFS 控制台上看到激活的同步代理。

## 步骤 2：创建同步任务

创建一个同步任务，并配置源和目标文件系统。

### 创建同步任务

- 选择创建同步任务。将显示配置源位置页。

The screenshot shows the 'Create sync task' interface in the Amazon EFS console. On the left, there is a sidebar with four steps: 'Configure source' (highlighted with an orange bar), 'Configure destination', 'Configure settings', and 'Review'. The main area is titled 'Configure source location' and contains the instruction 'Specify the location that you want to copy from.' Below this, there are three input fields: 'NFS server' with the value '192.0.2.0', 'Mount Path' with the value '/source-path', and 'Agent' with a dropdown menu showing 'Syn-agent-op (host-08ca202c534bbadee)'. At the bottom right, there are two buttons: 'Cancel' and 'Next: Configure destination'.

- 为源文件系统提供以下信息：

- 对于 NFS 服务器，请键入源 NFS 服务器的域名或 IP 地址。
  - 对于挂载路径，请键入源文件系统的挂载路径。
  - 对于代理，请选择您以前创建的同步代理。
3. 选择下一步：配置目标。将显示配置目标位置页。

The screenshot shows the 'Create sync task' console page. On the left, a sidebar contains links for 'Configure source', 'Configure destination' (which is highlighted with an orange bar), 'Configure settings', and 'Review'. The main content area is titled 'Configure destination location' and includes the instruction 'Specify the location that you want to copy into.' Below this, there are three configuration fields: 'Amazon EFS file systems' with a dropdown menu showing 'fs-01bf5478', 'File system path' with a text input field containing '/', and 'Security group' with a dropdown menu showing 'sg-xxxxxx1 (default)'. At the bottom right of the main area, there are three buttons: 'Cancel', 'Previous', and 'Next: Configure settings'.

4. 为目标文件系统提供以下信息：
- 对于 Amazon EFS 文件系统，请选择要同步到的 EFS 文件系统。如果没有 Amazon EFS 文件系统，请立即创建一个文件系统，并在完成后重新启动本演练。有关如何创建 Amazon EFS 文件系统的更多信息，请参阅[Amazon Elastic File System 入门 \(p. 9\)](#)。
  - 对于文件系统路径，请键入要将数据写入到的文件系统路径。该路径必须位于目标文件系统中。
  - 对于安全组，请选择一个允许访问选定的目标 Amazon EFS 文件系统的安全组。
5. 选择下一步：配置设置。将显示配置同步设置页。

The screenshot shows the 'Create sync task' console page, specifically the 'Configure sync settings' step. The left sidebar is the same as in the previous screenshot, but 'Configure settings' is now highlighted with an orange bar. The main content area is titled 'Configure sync settings' and includes the instruction 'Settings to use when synchronizing your files'. It contains three sections of settings, each with a label and a checkbox: 'Copy file metadata' with checkboxes for 'Ownership', 'Permissions', and 'Timestamps'; 'File deletion' with a checkbox for 'Keep files in destination even if not found in source'; and 'Verification mode' with a checkbox for 'Full consistency between source and destination file systems'. All three checkboxes are checked. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

6. 配置您希望该同步任务在同步文件时使用的默认设置：

#### Note

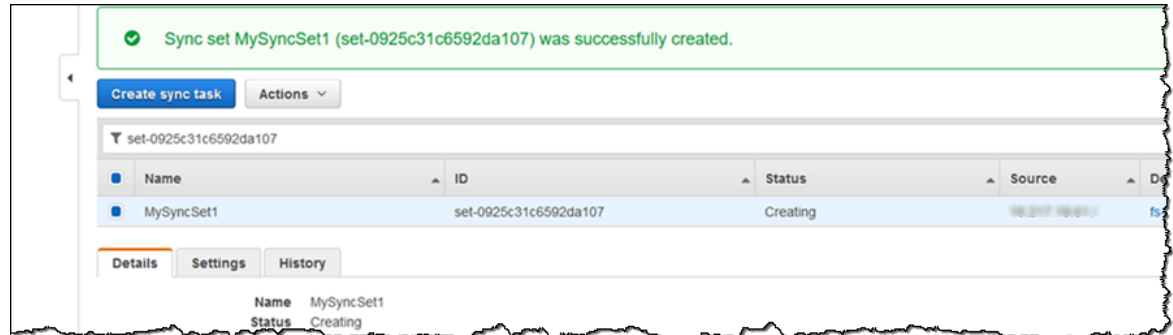
您可以在以后启动同步任务时覆盖这些设置。

- 选择所有权 (用户/组 ID) 以从源文件中复制用户和组 ID。
- 选择权限以复制源文件权限。
- 选择时间戳以从源文件中复制时间戳。



- 选择文件删除以在目标中保留在源文件系统中未找到的所有文件。如果清除该框，则在目标中删除在源文件系统中未找到的所有文件。
- 选择验证模式以在同步任务完成后检查目标文件系统是否为源文件系统的精确副本。如果未选择该选项，则仅验证传输的数据。不会查找在主动传输文件时对其进行的更改以及对未主动传输的文件进行的更改。我们建议您选择完整验证。

7. 选择下一步：审核和创建，然后查看您的同步任务设置。在准备就绪时，选择创建同步任务。



将创建您的同步任务。如果已挂载源和目标文件系统，任务状态将显示为可用。

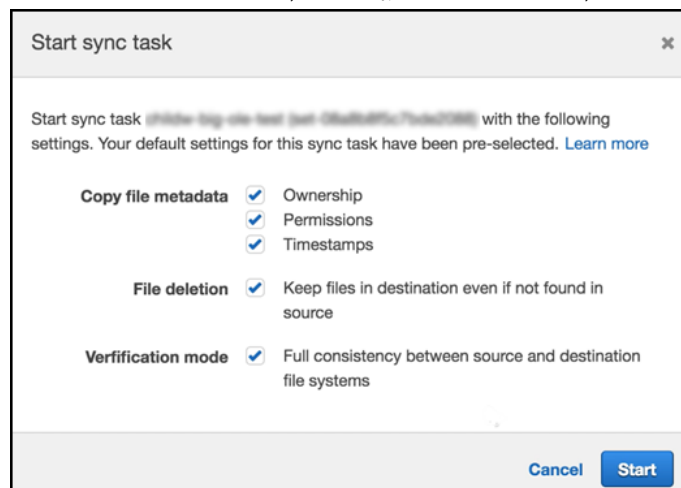
详细信息选项卡显示源和目标文件系统的状态和设置。

## 步骤 3：将源文件系统同步到 Amazon EFS

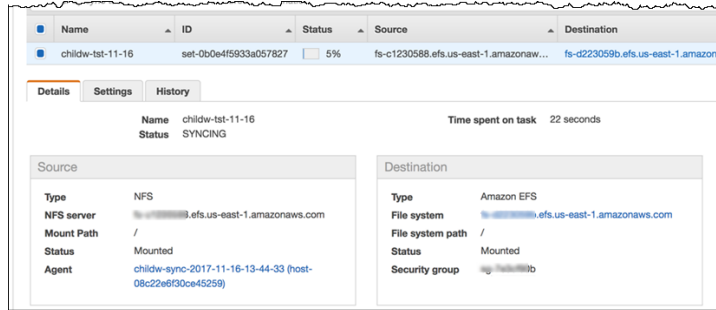
现在，您具有一个同步任务，您可以启动同步任务以开始将文件从源文件系统同步到目标 EFS 文件同步文件系统。

### 同步源文件系统

1. 在“任务”页中，选择刚创建的同步任务。“详细信息”选项卡将显示同步任务的状态。
2. 在操作菜单中，选择启动。
3. 在启动同步任务对话框中，您可以修改同步任务设置，然后选择启动。



4. 选择启动以开始同步文件。



5. 在启动同步任务时，状态列将显示同步任务进度。在同步任务开始准备时，状态将从正在启动变为正在准备。在任务开始同步文件时，状态将从正在准备变为正在同步。在启动文件一致性验证时，状态将变为正在验证。在同步任务完成后，状态将变为成功。

## 步骤 4：访问文件

要访问您的文件，请从 Amazon EC2 实例连接到 Amazon EFS 文件系统或使用 AWS Direct Connect。

有关如何使用 Amazon EC2 进行连接的信息，请参阅[连接到 Amazon EC2 实例和挂载 Amazon EFS 文件系统](#)。

有关如何使用 AWS Direct Connect 进行连接的信息，请参阅[演练 5：使用 AWS Direct Connect 在本地创建和挂载文件系统 \(p. 127\)](#)。

## 步骤 4：清除

如果不再需要使用创建的资源，应将其删除以保护您的账户：

如果不再需要使用创建的资源，应将其删除：

- 删除您创建的任务。有关更多信息，请参阅[删除同步任务 \(p. 48\)](#)。
- 删除您创建的同步代理。这不会删除您启动的 Amazon EC2 实例。
- 清理您创建的 Amazon EFS 资源。有关更多信息，请参阅[步骤 5：清理资源并保护您的 AWS 账户 \(p. 15\)](#)。
- 如果在 Amazon EC2 上创建了 EFS 文件同步，请清除您的实例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的[步骤 3：清除您的实例](#)。

# Amazon EFS 的身份验证和访问控制

要访问 Amazon EFS 或 Amazon EFS 文件同步，需要使用 AWS 可用于验证您的请求的凭证。这些凭证必须有权访问 AWS 资源，如 Amazon EFS 文件系统或 Amazon Elastic Compute Cloud (Amazon EC2) 实例。下面几节详细说明如何使用 [AWS Identity and Access Management \(IAM\)](#) 和 Amazon EFS 控制有权访问资源的角色，从而对这些资源进行保护。

- [身份验证 \(p. 144\)](#)
- [访问控制 \(p. 145\)](#)

## 身份验证

您可以以下面任一类型的身份访问 AWS：

- **AWS 账户根用户** – 注册 AWS 时，您需要提供与您的 AWS 账户关联的电子邮件地址和密码。这就是您的 AWS 账户根用户。其凭证可为您提供访问您所有 AWS 资源的完整权限。

### Important

出于安全考虑，我们建议您仅使用根用户创建管理员用户，此类用户是对您的 AWS 账户具有完全访问权限的 IAM 用户。然后，您可以使用此管理员用户来创建权限有限的其他 IAM 用户和角色。有关更多信息，请参阅 IAM 用户指南 中的 [IAM 最佳实践](#) 和 [创建管理员用户和组](#)。

- **IAM 用户** – [IAM 用户](#) 就是您的 AWS 账户中的一种身份，它具有特定的自定义权限（例如，在 Amazon EFS 中创建 a file system 的权限）。您可以使用 IAM 用户名和密码来登录以保护 AWS 网页，如 [AWS 管理控制台](#)、[AWS 开发论坛](#) 或 [AWS Support Center](#)。

除了用户名和密码之外，您还可以为每个用户生成 [访问密钥](#)。在通过 [多个软件开发工具包](#) 之一或使用 [AWS Command Line Interface \(CLI\)](#) 以编程方式访问 AWS 服务时，可以使用这些密钥。SDK 和 CLI 工具使用访问密钥对您的请求进行加密签名。如果您不使用 AWS 工具，则必须自行对请求签名。Amazon EFS supports 签名版本 4，后者是一种用于对入站 API 请求进行身份验证的协议。有关验证请求的更多信息，请参阅 AWS General Reference 中的 [签名版本 4 签名流程](#)。

- **IAM 角色** – [IAM 角色](#) 是可在账户中创建的另一种具有特定权限的 IAM 身份。它类似于 IAM 用户，但未与特定人员关联。利用 IAM 角色，您可以获得可用于访问 AWS 服务和资源的临时访问密钥。具有临时凭证的 IAM 角色在以下情况下很有用：
- **联合身份用户访问** – 您可以不创建 IAM 用户，而是使用来自 AWS Directory Service、您的企业用户目录或 Web 身份提供商的既有用户身份。他们被称为联合身份用户。在通过 [身份提供商](#) 请求访问权限时，AWS 将为联合用户分配角色。有关联合身份用户的更多信息，请参阅 IAM 用户指南 中的 [联合身份用户和角色](#)。
- **跨账户管理** – 可以使用您的账户中的 IAM 角色为另一个 AWS 账户授予权限以管理您的账户的 Amazon EFS 资源。有关示例，请参阅 IAM 用户指南 中的 [教程：使用 IAM 角色委派跨 AWS 账户的访问权限](#)。请注意，您无法跨 VPC 或账户挂载 Amazon EFS 文件系统。有关更多信息，请参阅 [管理文件系统网络可访问性 \(p. 39\)](#)。

- AWS 服务访问 - 可以使用您账户中的 IAM 角色向 AWS 服务授予对您账户的资源的访问权。例如，您可以创建一个角色，此角色允许 Amazon Redshift 代表您访问 Amazon S3 存储桶，然后将该存储桶提供的数据库加载到 Amazon Redshift 群集中。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](#)。
- 在 Amazon EC2 上运行的应用程序 – 您可以使用 IAM 角色管理在 EC2 实例上运行的应用程序的临时凭证并发出 AWS API 请求。这优先于在 EC2 实例中存储访问密钥。要将 AWS 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅 IAM 用户指南 中的 [对 Amazon EC2 上的应用程序使用角色](#)。

## 访问控制

您可以拥有有效的凭证来对自己的请求进行身份验证，但您还必须拥有权限才能创建或访问 Amazon Elastic File System 资源。例如，您必须拥有权限才能创建 Amazon EFS 文件系统。

以下几节介绍如何管理 Amazon Elastic File System 的权限。我们建议您先阅读概述。

- [管理您的 Amazon EFS 资源的访问权限概述 \(p. 145\)](#)
- [为 Amazon Elastic File System 使用基于身份的策略 \(IAM 策略\) \(p. 148\)](#)

## 管理您的 Amazon EFS 资源的访问权限概述

每个 AWS 资源都归某个 AWS 账户所有，创建和访问资源的权限由权限策略进行管理。账户管理员可以向 IAM 身份 (即：用户、组和角色) 附加权限策略，某些服务 (如 AWS Lambda) 也支持向资源附加权限策略。

### Note

账户管理员 (或管理员用户) 是具有管理员权限的用户。有关更多信息，请参阅 IAM 用户指南 中的 [IAM 最佳实践](#)。

在授予权限时，您要决定谁获得权限，获得对哪些资源的权限，以及您允许对这些资源执行的具体操作。

### 主题

- [Amazon Elastic File System 资源和操作 \(p. 145\)](#)
- [了解资源所有权 \(p. 146\)](#)
- [管理对资源的访问 \(p. 146\)](#)
- [指定策略元素：操作、效果和委托人 \(p. 147\)](#)
- [在策略中指定条件 \(p. 148\)](#)

## Amazon Elastic File System 资源和操作

在 Amazon Elastic File System 中，主要资源是文件系统。Amazon Elastic File System 还支持其他资源类型，例如挂载目标和标签。不过，对于 Amazon EFS，您只能在现有文件系统范围内创建挂载目标和标签。挂载目标和标签称为子资源。

这些资源和子资源具有与其关联的唯一 Amazon 资源名称 (ARN)，如下表所示。

Amazon EFS 提供一组操作用来处理 Amazon EFS 资源。有关可用操作的列表，请参阅 Amazon Elastic File System [Actions \(p. 155\)](#)。

## 了解资源所有权

AWS 账户对在该账户下创建的资源具有所有权，而无论创建资源的人员是谁。具体而言，资源所有者是对资源创建请求进行身份验证的[委托人实体](#) (即根账户、IAM 用户或 IAM 角色) 的 AWS 账户。以下示例说明了它的工作原理：

- 如果使用您的 AWS 账户的根账户凭证来创建文件系统，则您的 AWS 账户就是该资源的所有者 (在 Amazon EFS 中，资源就是文件系统)。
- 如果您在您的 AWS 账户中创建 IAM 用户并向该用户授予创建文件系统的权限，则该用户可以创建文件系统。但是，您的 AWS 账户 (即该用户所属的账户) 拥有该文件系统资源。
- 如果您在您的 AWS 账户中创建具有文件系统创建权限的 IAM 角色，则能够担任该角色的任何人都可以创建文件系统。角色所属的 AWS 账户拥有该文件系统资源。

## 管理对资源的访问

权限策略 规定谁可以访问哪些内容。下一节介绍创建权限策略时的可用选项。

### Note

本节讨论如何在 Amazon Elastic File System 范围内使用 IAM。这里不提供有关 IAM 服务的详细信息。有关完整的 IAM 文档，请参阅[什么是 IAM？](#) (在 IAM 用户指南 中)。有关 IAM 策略语法和说明的信息，请参阅 IAM 用户指南 中的[AWS IAM 策略参考](#)。

挂载到 IAM 身份的策略称作基于身份的策略 (IAM 策略)，而挂载到资源的策略称作基于资源的策略。Amazon Elastic File System 只支持基于身份的策略 (IAM 策略)。

### 主题

- [基于身份的策略 \(IAM 策略\) \(p. 146\)](#)
- [基于资源的策略 \(p. 147\)](#)

## 基于身份的策略 (IAM 策略)

您可以向 IAM 身份挂载策略。例如，您可以执行以下操作：

- 将权限策略附加到您的账户中的用户或组 – 要向用户授予创建 Amazon EFS 资源 (例如文件系统) 的权限，您可以将权限策略附加到用户或用户所属的组。
- 向角色附加权限策略 (授予跨账户权限) – 您可以向 IAM 角色附加基于身份的权限策略，以授予跨账户的权限。例如，账户 A 中的管理员可以创建一个角色，以向其他 AWS 账户 (如账户 B) 或某项 AWS 服务授予跨账户权限，如下所述：
  1. 账户 A 管理员创建一个 IAM 角色，向该角色挂载授权其访问账户 A 中资源的权限策略。
  2. 账户 A 管理员可以向将账户 B 标识为能够担任该角色的委托人的角色附加信任策略。
  3. 之后，账户 B 管理员可以委派权限，指派账户 B 中的任何用户担任该角色。这样，账户 B 中的用户就可以创建或访问账户 A 中的资源了。如果您需要授予 AWS 服务权限来担任该角色，则信任策略中的委托人也可以是 AWS 服务委托人。

有关使用 IAM 委派权限的更多信息，请参阅 IAM 用户指南 中的[访问权限管理](#)。

以下是允许用户对您的 AWS 账户执行 CreateFileSystem 操作的示例策略。

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

```
{
  "Sid" : "Stmt1EFSpermissions",
  "Effect": "Allow",
  "Action": [
    "elasticfilesystem:CreateFileSystem",
    "elasticfilesystem:CreateMountTarget"
  ],
  "Resource": "arn:aws:elasticfilesystem:us-west-2:account-id:file-system/*"
},
{
  "Sid" : "Stmt2EC2permissions",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeSubnets",
    "ec2:CreateNetworkInterface",
    "ec2:DescribeNetworkInterfaces"
  ],
  "Resource": "*"
}
]
```

有关将基于身份的策略用于 Amazon EFS 的更多信息，请参阅[Amazon Elastic File System 使用基于身份的策略 \(IAM 策略\) \(p. 148\)](#)。有关用户、组、角色和权限的更多信息，请参阅 IAM 用户指南 中的[身份 \(用户、组和角色\)](#)。

## 基于资源的策略

其他服务 (如 Amazon S3) 也支持基于资源的权限策略。例如，您可以将策略附加到 S3 存储桶以管理对该存储桶的访问权限。Amazon Elastic File System 不支持基于资源的策略。

## 指定策略元素：操作、效果和委托人

对于每个 Amazon Elastic File System 资源 (请参阅[Amazon Elastic File System 资源和操作 \(p. 145\)](#))，该服务定义了一组 API 操作 (请参阅[Actions \(p. 155\)](#))。为授予这些 API 操作的权限，Amazon EFS 定义了一组您可以在策略中指定的操作。例如，对于 Amazon EFS 文件系统资源，定义了以下操作：CreateFileSystem、DeleteFileSystem 和 DescribeFileSystems。请注意，执行某项 API 操作可能需要执行多个操作的权限。

以下是最基本的策略元素：

- Resource – 在策略中，您可以使用 Amazon 资源名称 (ARN) 标识策略应用到的资源。有关更多信息，请参阅[Amazon Elastic File System 资源和操作 \(p. 145\)](#)。
- Action – 操作关键字用于标识要允许或拒绝的资源操作。例如，根据指定的 Effect，elasticfilesystem:CreateFileSystem 允许或拒绝执行 Amazon Elastic File System CreateFileSystem 操作的用户权限。
- Effect – 用于指定当用户请求特定操作 (可以是允许或拒绝) 时的效果。如果没有显式授予 (允许) 对资源的访问权限，则隐式拒绝访问。您也可显式拒绝对资源的访问，这样可确保用户无法访问该资源，即使有其他策略授予了访问权限的情况下也是如此。
- Principal – 在基于身份的策略 (IAM 策略) 中，附加了策略的用户是隐式委托人。对于基于资源的策略，您可以指定要接收权限的用户、账户、服务或其他实体 (仅适用于基于资源的策略)。Amazon EFS 不支持基于资源的策略。

有关 IAM 策略语法和说明的更多信息，请参阅 IAM 用户指南 中的[AWS IAM 策略参考](#)。

有关显示所有 Amazon Elastic File System API 操作的表，请参阅[Amazon EFS API 权限：操作、资源和条件参考 \(p. 151\)](#)。



## 在策略中指定条件

当您授予权限时，可使用 IAM 策略语言来指定规定策略何时生效的条件。例如，您可能希望策略仅在特定日期后应用。有关使用策略语言指定条件的更多信息，请参阅 IAM 用户指南 中的 [条件](#)。

要表示条件，您可以使用预定义的条件键。没有特定于 Amazon Elastic File System 的条件键。但有 AWS 范围内的条件密钥，您可以根据需要使用。有关 AWS 范围内的键的完整列表，请参阅 IAM 用户指南 中的 [条件的可用键](#)。

### Note

不要对 `CreateMountTarget`、`DeleteMountTarget` 或 `ModifyMountTargetSecurityGroup` 操作使用 `aws:SourceIp` AWS 范围的条件。Amazon EFS 使用其自身的 IP 地址 (而不是原始请求的 IP 地址) 来配置挂载目标。

## 为 Amazon Elastic File System 使用基于身份的策略 (IAM 策略)

本主题提供了基于身份的策略的示例，这些示例展示了账户管理员如何将权限策略附加到 IAM 身份 (即用户、组和角色)，从而授予对 Amazon EFS 资源执行操作的权限。

### Important

我们建议您首先阅读以下介绍性主题，这些主题讲解了管理 Amazon Elastic File System 资源访问权限的基本概念和选项。有关更多信息，请参阅 [管理您的 Amazon EFS 资源的访问权限概述 \(p. 145\)](#)。

本主题的各个部分涵盖以下内容：

- [使用 Amazon EFS 控制台所需要的权限 \(p. 149\)](#)
- [适用于 Amazon EFS 的 AWS 托管 \(预定义\) 策略 \(p. 150\)](#)
- [客户托管策略示例 \(p. 150\)](#)

下面介绍权限策略示例。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowFileSystemPermissions",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:CreateMountTarget"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-west-2:account-id:file-system/*"
    },
    {
      "Sid": "AllowEC2Permissions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces"
      ],
    }
  ]
}
```



```
    "Resource": "*"
  }
]
}
```

该策略包含两条语句：

- 第一个语句通过对文件系统使用 Amazon 资源名称 (ARN) 来授予对资源执行两个 Amazon EFS 操作 (elasticfilesystem:CreateFileSystem 和 elasticfilesystem:CreateMountTarget) 的权限。ARN 将指定一个通配符 (\*), 因为您在创建文件系统之前不知道文件系统 ID。
- 第二条语句授予对某些 Amazon EC2 操作的权限, 因为第一条语句中的 elasticfilesystem:CreateMountTarget 操作需要特定 Amazon EC2 操作的权限。由于这些 Amazon EC2 操作不支持资源级权限, 因此该策略将指定通配符 (\*) 作为 Resource 值, 而不是指定一个文件系统 ARN。

该策略不指定 Principal 元素, 因为在基于身份的策略中, 您未指定获取权限的委托人。挂载了策略的用户是隐式委托人。向 IAM 角色挂载权限策略后, 该角色的信任策略中标识的委托人获取权限。

有关显示所有 Amazon Elastic File System API 操作及其所适用资源的表, 请参阅[Amazon EFS API 权限：操作、资源和条件参考 \(p. 151\)](#)。

## 使用 Amazon EFS 控制台所需要的权限

权限参考表列出了 Amazon EFS API 操作以及每个操作所需的权限。有关 Amazon EFS API 操作的更多信息, 请参阅 [Amazon EFS API 权限：操作、资源和条件参考 \(p. 151\)](#)。

要使用 Amazon EFS 控制台, 需要如下权限策略中所示, 授予执行其他操作的权限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid" : "Stmt1AdditionalEC2PermissionsForConsole",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcAttribute"
      ],
      "Resource": "*"
    },
    {
      "Sid" : "Stmt2AdditionalKMSPermissionsForConsole",
      "Effect": "Allow",
      "Action": [
        "kms:ListAliases",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon EFS 控制台出于以下原因需要上述其他权限：

- Amazon EFS 操作权限使控制台可以显示账户中的 Amazon EFS 资源。
- 控制台需要 ec2 操作权限来查询 Amazon EC2, 以便可以显示可用区、VPC、安全组及账户属性。

- 控制台需要具有 kms 操作权限以创建加密的文件系统。有关加密的文件系统的更多信息，请参阅[在 EFS 中加密数据和元数据 \(p. 86\)](#)。

## 适用于 Amazon EFS 的 AWS 托管 (预定义) 策略

AWS 通过提供由 AWS 创建和管理的独立 IAM 策略来解决很多常用案例。托管策略可授予常用案例的必要权限，因此，您可以免去调查都需要哪些权限的工作。有关更多信息，请参阅 IAM 用户指南 中的 [AWS 托管策略](#)。

以下 AWS 托管策略 (您可以将它们附加到自己账户中的用户) 是特定于 Amazon EFS 的：

- AmazonElasticFileSystemReadOnlyAccess – 授予对 Amazon EFS 资源的只读访问权限。
- AmazonElasticFileSystemFullAccess – 授予对 Amazon EFS 资源的完全访问权限。

### Note

您可以通过登录到 IAM 控制台并在该控制台中搜索特定策略来查看这些权限策略。

此外，您还可以创建自定义 IAM 策略，以授予执行 Amazon EFS API 操作的相关权限。您可以将这些自定义策略挂载到需要这些权限的 IAM 用户或组。

## 客户托管策略示例

本节的用户策略示例介绍如何授予各 Amazon EFS 操作的权限。当您使用 AWS SDK 或 AWS CLI 时，可以使用这些策略。当您使用控制台时，您需要授予特定于控制台的其他权限，[使用 Amazon EFS 控制台所需要的权限 \(p. 149\)](#)中对此进行了讨论。

### Note

所有示例都使用 us-west-2 区域和虚构的账户 ID。

### 示例

- [示例 1：允许用户在现有文件系统上创建挂载目标和标签 \(p. 150\)](#)
- [示例 2：允许用户执行所有 Amazon EFS 操作 \(p. 151\)](#)

## 示例 1：允许用户在现有文件系统上创建挂载目标和标签

以下权限策略授予用户在 us-west-2 区域的特定文件系统上创建挂载目标和标签的权限。要创建挂载目标，还需要执行特定 Amazon EC2 操作的权限，并将其包含在权限策略中。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1CreateMountTargetAndTag",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:CreateMountTarget",
        "elasticfilesystem:DescribeMountTargets",
        "elasticfilesystem:CreateTags",
        "elasticfilesystem:DescribeTags"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/file-system-ID"
    },
    {
```

```
    "Sid" : "Stmt2AdditionalEC2PermissionsToCreateMountTarget",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "*"
  }
}
```

## 示例 2：允许用户执行所有 Amazon EFS 操作

以下权限策略使用通配符 ("elasticfilesystem:\*") 以允许 us-west-2 区域内的所有 Amazon EFS 操作。由于某些 Amazon EFS 操作还需要 Amazon EC2 操作的权限，因此该策略还会为所有这些操作授予权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid" : "Stmt1PermissionForAllEFSActions",
      "Effect": "Allow",
      "Action": "elasticfilesystem:*",
      "Resource": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/*"
    },
    {
      "Sid" : "Stmt2RequiredEC2PermissionsForAllEFSActions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2>DeleteNetworkInterface",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DescribeNetworkInterfaceAttribute"
      ],
      "Resource": "*"
    }
  ]
}
```

## Amazon EFS API 权限：操作、资源和条件参考

在设置 [访问控制](#) (p. 145) 和编写可附加到 IAM 身份的权限策略 (基于身份的策略) 时，可使用下面的列表作为参考。该列表包含每个 Amazon EFS API 操作、您可授予执行权限的对应操作以及您可为其授予权限的 AWS 资源。您可以在策略的 Action 字段中指定这些操作，并在策略的 Resource 字段中指定资源值。

您可以在 Amazon EFS 策略中使用 AWS 范围的条件键来表达条件。有关 AWS 范围内的密钥的完整列表，请参阅 IAM 用户指南 中的 [可用密钥](#)。

### Note

要指定操作，请在 API 操作名称之前使用 elasticfilesystem: 前缀 (例如，elasticfilesystem:CreateFileSystem)。

Amazon EFS API 和必需的操作权限

#### [CreateFileSystem \(p. 157\)](#)

操作 : elasticfilesystem:CreateFileSystem

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/\*

#### [CreateMountTarget \(p. 164\)](#)

操

作 : elasticfilesystem:CreateMountTarget、ec2:DescribeSubnets、ec2:DescribeNetworkInterf

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [CreateTags \(p. 171\)](#)

操作 : elasticfilesystem:CreateTags

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [DeleteFileSystem \(p. 174\)](#)

操作 : elasticfilesystem>DeleteFileSystem

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [DeleteMountTarget \(p. 176\)](#)

操作 : elasticfilesystem>DeleteMountTarget、ec2>DeleteNetworkInterface

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [DeleteTags \(p. 179\)](#)

操作 : elasticfilesystem>DeleteTags

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [DescribeFileSystems \(p. 181\)](#)

操作 : elasticfilesystem:DescribeFileSystems

资源: arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id,  
arn:aws:elasticfilesystem:region:account-id:file-system/\*

#### [DescribeMountTargetSecurityGroups \(p. 188\)](#)

操

作 : elasticfilesystem:DescribeMountTargetSecurityGroups、ec2:DescribeNetworkInterfaceAt

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [DescribeMountTargets \(p. 185\)](#)

操作 : elasticfilesystem:DescribeMountTargets

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [DescribeTags \(p. 191\)](#)

操作 : elasticfilesystem:DescribeTags

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

#### [ModifyMountTargetSecurityGroups \(p. 194\)](#)

操作 : elasticfilesystem:ModifyMountTargetSecurityGroups、  
ec2:ModifyNetworkInterfaceAttribute

资源 : arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id

[UpdateFileSystem \(p. 197\)](#)

操作 : elasticfilesystem:UpdateFileSystem、elasticfilesystem:UpdateFileSystem

资源 : arn:aws:elasticfilesystem:*region*:*account-id*:file-system/*file-system-id*

# 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 终端节点发出 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 所匹配的模式共同确定。如果操作为 PUT 或 POST，则 Amazon EFS 将从请求正文内的请求 URI 路径段、查询参数和 JSON 对象中提取调用自变量。

## 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 的身份验证和访问控制 \(p. 144\)](#)。

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 `create-file-system` 的 AWS CLI 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

API 终端节点是在 API 调用的 HTTP URI 中用作主机的 DNS 名称。这些 API 终端节点是 AWS 区域特有的，并采用以下形式。

```
elasticfilesystem.aws-region.amazonaws.com
```

例如，美国西部（俄勒冈）区域的 Amazon EFS API 终端节点如下所示。

```
elasticfilesystem.us-west-2.amazonaws.com
```

有关 Amazon EFS 支持的 AWS 区域（可在其中创建和管理文件系统）列表，请参阅 AWS General Reference 中的 [Amazon Elastic File System](#)。

特定于区域的 API 终端节点定义在执行 API 调用时可访问的 Amazon EFS 资源的范围。例如，当使用上述终端节点调用 `DescribeFileSystems` 操作时，您将获得已在您的账户中创建的位于美国西部（俄勒冈）区域的文件系统列表。

## 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 \(p. 157\)](#)。

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 的身份验证和访问控制 \(p. 144\)](#)
- [Actions \(p. 155\)](#)
- [Data Types \(p. 201\)](#)

## 使用 Amazon EFS 的查询 API 请求速率

将针对每个区域限制每个 AWS 账户的 Amazon EFS API 请求以帮助提高服务性能。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。AWS Identity and Access Management (IAM) 用户发出的 API 请求受基础 AWS 账户的限制。

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。指数退避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 [AWS 中的错误重试和指数回退](#)。

## Actions

The following actions are supported:

- [CreateFileSystem \(p. 157\)](#)
- [CreateMountTarget \(p. 164\)](#)
- [CreateTags \(p. 171\)](#)
- [DeleteFileSystem \(p. 174\)](#)
- [DeleteMountTarget \(p. 176\)](#)
- [DeleteTags \(p. 179\)](#)
- [DescribeFileSystems \(p. 181\)](#)



- [DescribeMountTargets](#) (p. 185)
- [DescribeMountTargetSecurityGroups](#) (p. 188)
- [DescribeTags](#) (p. 191)
- [ModifyMountTargetSecurityGroups](#) (p. 194)
- [UpdateFileSystem](#) (p. 197)

## CreateFileSystem

Creates a new, empty file system. The operation requires a creation token in the request that Amazon EFS uses to ensure idempotent creation (calling the operation with same creation token has no effect). If a file system does not currently exist that is owned by the caller's AWS account with the specified creation token, this operation does the following:

- Creates a new, empty file system. The file system will have an Amazon EFS assigned ID, and an initial lifecycle state `creating`.
- Returns with the description of the created file system.

Otherwise, this operation returns a `FileSystemAlreadyExists` error with the ID of the existing file system.

### Note

For basic use cases, you can use a randomly generated UUID for the creation token.

The idempotent operation allows you to retry a `CreateFileSystem` call without risk of creating an extra file system. This can happen when an initial call fails in a way that leaves it uncertain whether or not a file system was actually created. An example might be that a transport level timeout occurred or your connection was reset. As long as you use the same creation token, if the initial call had succeeded in creating a file system, the client can learn of its existence from the `FileSystemAlreadyExists` error.

### Note

The `CreateFileSystem` call returns while the file system's lifecycle state is still `creating`. You can check the file system creation status by calling the [DescribeFileSystems](#) (p. 181) operation, which among other things returns the file system state.

This operation also takes an optional `PerformanceMode` parameter that you choose for your file system. We recommend `generalPurpose` performance mode for most file systems. File systems using the `maxIO` performance mode can scale to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for most file operations. The performance mode can't be changed after the file system has been created. For more information, see [Amazon EFS: Performance Modes](#).

After the file system is fully created, Amazon EFS sets its lifecycle state to `available`, at which point you can create one or more mount targets for the file system in your VPC. For more information, see [CreateMountTarget](#) (p. 164). You mount your Amazon EFS file system on an EC2 instances in your VPC by using the mount target. For more information, see [Amazon EFS: How it Works](#).

This operation requires permissions for the `elasticfilesystem:CreateFileSystem` action.

## Request Syntax

```
POST /2015-02-01/file-systems HTTP/1.1
Content-type: application/json

{
  "CreationToken": "string",
  "Encrypted": boolean,
  "KmsKeyId": "string",
  "PerformanceMode": "string",
  "ProvisionedThroughputInMibps": number,
  "ThroughputMode": "string"
}
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in JSON format.

### [CreationToken \(p. 157\)](#)

String of up to 64 ASCII characters. Amazon EFS uses this to ensure idempotent creation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Required: Yes

### [Encrypted \(p. 157\)](#)

A Boolean value that, if true, creates an encrypted file system. When creating an encrypted file system, you have the option of specifying a [CreateFileSystem:KmsKeyId \(p. 158\)](#) for an existing AWS Key Management Service (AWS KMS) customer master key (CMK). If you don't specify a CMK, then the default CMK for Amazon EFS, `/aws/elasticfilesystem`, is used to protect the encrypted file system.

Type: Boolean

Required: No

### [KmsKeyId \(p. 157\)](#)

The ID of the AWS KMS CMK to be used to protect the encrypted file system. This parameter is only required if you want to use a non-default CMK. If this parameter is not specified, the default CMK for Amazon EFS is used. This ID can be in one of the following formats:

- Key ID - A unique identifier of the key, for example, `1234abcd-12ab-34cd-56ef-1234567890ab`.
- ARN - An Amazon Resource Name (ARN) for the key, for example, `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- Key alias - A previously created display name for a key. For example, `alias/projectKey1`.
- Key alias ARN - An ARN for a key alias, for example, `arn:aws:kms:us-west-2:444455556666:alias/projectKey1`.

If `KmsKeyId` is specified, the [CreateFileSystem:Encrypted \(p. 158\)](#) parameter must be set to true.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

### [PerformanceMode \(p. 157\)](#)

The `PerformanceMode` of the file system. We recommend `generalPurpose` performance mode for most file systems. File systems using the `maxIO` performance mode can scale to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for most file operations. This can't be changed after the file system has been created.

Type: String

Valid Values: `generalPurpose` | `maxIO`

Required: No

#### [ProvisionedThroughputInMibps \(p. 157\)](#)

The throughput, measured in MiB/s, that you want to provision for a file system that you're creating. The limit on throughput is 1024 MiB/s. You can get these limits increased by contacting AWS Support. For more information, see [Amazon EFS Limits That You Can Increase](#) in the Amazon EFS User Guide.

Type: Double

Valid Range: Minimum value of 0.0.

Required: No

#### [ThroughputMode \(p. 157\)](#)

The throughput mode for the file system to be created. There are two throughput modes to choose from for your file system: bursting and provisioned. You can decrease your file system's throughput in Provisioned Throughput mode or change between the throughput modes as long as it's been more than 24 hours since the last decrease or throughput mode change.

Type: String

Valid Values: bursting | provisioned

Required: No

## Response Syntax

```
HTTP/1.1 201
Content-type: application/json

{
  "CreationTime": number,
  "CreationToken": "string",
  "Encrypted": boolean,
  "FileSystemId": "string",
  "KmsKeyId": "string",
  "LifecycleState": "string",
  "Name": "string",
  "NumberOfMountTargets": number,
  "OwnerId": "string",
  "PerformanceMode": "string",
  "ProvisionedThroughputInMibps": number,
  "SizeInBytes": {
    "Timestamp": number,
    "Value": number
  },
  "ThroughputMode": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 201 response.

The following data is returned in JSON format by the service.

#### [CreationTime \(p. 159\)](#)

Time that the file system was created, in seconds (since 1970-01-01T00:00:00Z).

Type: Timestamp

#### [CreationToken \(p. 159\)](#)

Opaque string specified in the request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

#### [Encrypted \(p. 159\)](#)

A Boolean value that, if true, indicates that the file system is encrypted.

Type: Boolean

#### [FileSystemId \(p. 159\)](#)

ID of the file system, assigned by Amazon EFS.

Type: String

#### [KmsKeyId \(p. 159\)](#)

The ID of an AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the encrypted file system.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

#### [LifecycleState \(p. 159\)](#)

Lifecycle phase of the file system.

Type: String

Valid Values: `creating` | `available` | `updating` | `deleting` | `deleted`

#### [Name \(p. 159\)](#)

You can add tags to a file system, including a `Name` tag. For more information, see [CreateTags \(p. 171\)](#). If the file system has a `Name` tag, Amazon EFS returns the value in this field.

Type: String

Length Constraints: Maximum length of 256.

#### [NumberOfMountTargets \(p. 159\)](#)

Current number of mount targets that the file system has. For more information, see [CreateMountTarget \(p. 164\)](#).

Type: Integer

Valid Range: Minimum value of 0.

#### [OwnerId \(p. 159\)](#)

AWS account that created the file system. If the file system was created by an IAM user, the parent account to which the user belongs is the owner.

Type: String

#### [PerformanceMode \(p. 159\)](#)

The `PerformanceMode` of the file system.

Type: String

Valid Values: `generalPurpose` | `maxIO`

[ProvisionedThroughputInMibps \(p. 159\)](#)

The throughput, measured in MiB/s, that you want to provision for a file system. The limit on throughput is 1024 MiB/s. You can get these limits increased by contacting AWS Support. For more information, see [Amazon EFS Limits That You Can Increase](#) in the Amazon EFS User Guide.

Type: Double

Valid Range: Minimum value of 0.0.

[SizeInBytes \(p. 159\)](#)

Latest known metered size (in bytes) of data stored in the file system, in its `Value` field, and the time at which that size was determined in its `Timestamp` field. The `Timestamp` value is the integer number of seconds since 1970-01-01T00:00:00Z. The `SizeInBytes` value doesn't represent the size of a consistent snapshot of the file system, but it is eventually consistent when there are no writes to the file system. That is, `SizeInBytes` represents actual size only if the file system is not modified for a period longer than a couple of hours. Otherwise, the value is not the exact size that the file system was at any point in time.

Type: [FileSystemSize \(p. 205\)](#) object

[ThroughputMode \(p. 159\)](#)

The throughput mode for a file system. There are two throughput modes to choose from for your file system: bursting and provisioned. You can decrease your file system's throughput in Provisioned Throughput mode or change between the throughput modes as long as it's been more than 24 hours since the last decrease or throughput mode change.

Type: String

Valid Values: `bursting` | `provisioned`

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### FileSystemAlreadyExists

Returned if the file system you are trying to create already exists, with the creation token you provided.

HTTP Status Code: 409

### FileSystemLimitExceeded

Returned if the AWS account has already created the maximum number of file systems allowed per account.

HTTP Status Code: 403

### InsufficientThroughputCapacity

Returned if there's not enough capacity to provision additional throughput. This value might be returned when you try to create a file system in provisioned throughput mode, when you attempt to increase the provisioned throughput of an existing file system, or when you attempt to change an existing file system from bursting to provisioned throughput mode.

HTTP Status Code: 503

InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

ThroughputLimitExceeded

Returned if the throughput mode or amount of provisioned throughput can't be changed because the throughput limit of 1024 MiB/s has been reached.

HTTP Status Code: 400

## Example

### Create a file system

The following example sends a POST request to create a file system in the `us-west-2` region. The request specifies `myFileSystem1` as the creation token.

#### Sample Request

```
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"
}
```

#### Sample Response

```
HTTP/1.1 201 Created
x-amzn-RequestId: 7560489e-8bc7-4a56-a09a-757ce6f4832a
Content-Type: application/json
Content-Length: 319

{
  "ownerId": "251839141158",
  "creationToken": "myFileSystem1",
  "PerformanceMode" : "generalPurpose",
  "fileSystemId": "fs-47a2c22e",
  "CreationTime": "1403301078",
  "LifecycleState": "creating",
  "numberOfMountTargets": 0,
  "sizeInBytes": {
    "value": 1024,
    "timestamp": "1403301078"
  }
}
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:



- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## CreateMountTarget

Creates a mount target for a file system. You can then mount the file system on EC2 instances by using the mount target.

You can create one mount target in each Availability Zone in your VPC. All EC2 instances in a VPC within a given Availability Zone share a single mount target for a given file system. If you have multiple subnets in an Availability Zone, you create a mount target in one of the subnets. EC2 instances do not need to be in the same subnet as the mount target in order to access their file system. For more information, see [Amazon EFS: How it Works](#).

In the request, you also specify a file system ID for which you are creating the mount target and the file system's lifecycle state must be `available`. For more information, see [DescribeFileSystems \(p. 181\)](#).

In the request, you also provide a subnet ID, which determines the following:

- VPC in which Amazon EFS creates the mount target
- Availability Zone in which Amazon EFS creates the mount target
- IP address range from which Amazon EFS selects the IP address of the mount target (if you don't specify an IP address in the request)

After creating the mount target, Amazon EFS returns a response that includes, a `MountTargetId` and an `IpAddress`. You use this IP address when mounting the file system in an EC2 instance. You can also use the mount target's DNS name when mounting the file system. The EC2 instance on which you mount the file system by using the mount target can resolve the mount target's DNS name to its IP address. For more information, see [How it Works: Implementation Overview](#).

Note that you can create mount targets for a file system in only one VPC, and there can be only one mount target per Availability Zone. That is, if the file system already has one or more mount targets created for it, the subnet specified in the request to add another mount target must meet the following requirements:

- Must belong to the same VPC as the subnets of the existing mount targets
- Must not be in the same Availability Zone as any of the subnets of the existing mount targets

If the request satisfies the requirements, Amazon EFS does the following:

- Creates a new mount target in the specified subnet.
- Also creates a new network interface in the subnet as follows:
  - If the request provides an `IpAddress`, Amazon EFS assigns that IP address to the network interface. Otherwise, Amazon EFS assigns a free address in the subnet (in the same way that the Amazon EC2 `CreateNetworkInterface` call does when a request does not specify a primary private IP address).
  - If the request provides `SecurityGroups`, this network interface is associated with those security groups. Otherwise, it belongs to the default security group for the subnet's VPC.
  - Assigns the description `Mount target fsmt-id for file system fs-id` where `fsmt-id` is the mount target ID, and `fs-id` is the `FileSystemId`.
  - Sets the `requesterManaged` property of the network interface to `true`, and the `requesterId` value to `EFS`.

Each Amazon EFS mount target has one corresponding requester-managed EC2 network interface. After the network interface is created, Amazon EFS sets the `NetworkInterfaceId` field in the mount target's description to the network interface ID, and the `IpAddress` field to its address. If network interface creation fails, the entire `CreateMountTarget` operation fails.

#### Note

The `CreateMountTarget` call returns only after creating the network interface, but while the mount target state is still `creating`, you can check the mount target creation status by calling the [DescribeMountTargets \(p. 185\)](#) operation, which among other things returns the mount target state.

We recommend you create a mount target in each of the Availability Zones. There are cost considerations for using a file system in an Availability Zone through a mount target created in another Availability Zone. For more information, see [Amazon EFS](#). In addition, by always using a mount target local to the instance's Availability Zone, you eliminate a partial failure scenario. If the Availability Zone in which your mount target is created goes down, then you won't be able to access your file system through that mount target.

This operation requires permissions for the following action on the file system:

- `elasticfilesystem:CreateMountTarget`

This operation also requires permissions for the following Amazon EC2 actions:

- `ec2:DescribeSubnets`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`

## Request Syntax

```
POST /2015-02-01/mount-targets HTTP/1.1
Content-type: application/json

{
  "FileSystemId": "string",
  "IpAddress": "string",
  "SecurityGroups": [ "string" ],
  "SubnetId": "string"
}
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in JSON format.

#### [FileSystemId \(p. 165\)](#)

ID of the file system for which to create the mount target.

Type: String

Required: Yes

#### [IpAddress \(p. 165\)](#)

Valid IPv4 address within the address range of the specified subnet.

Type: String

Required: No

#### [SecurityGroups \(p. 165\)](#)

Up to five VPC security group IDs, of the form `sg-xxxxxxx`. These must be for the same VPC as subnet specified.

Type: Array of strings

Array Members: Maximum number of 5 items.

Required: No

#### [SubnetId \(p. 165\)](#)

ID of the subnet to add the mount target in.

Type: String

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "FileSystemId": "string",
  "IpAddress": "string",
  "LifecycleState": "string",
  "MountTargetId": "string",
  "NetworkInterfaceId": "string",
  "OwnerId": "string",
  "SubnetId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### [FileSystemId \(p. 166\)](#)

ID of the file system for which the mount target is intended.

Type: String

#### [IpAddress \(p. 166\)](#)

Address at which the file system can be mounted by using the mount target.

Type: String

#### [LifecycleState \(p. 166\)](#)

Lifecycle state of the mount target.

Type: String

Valid Values: `creating` | `available` | `updating` | `deleting` | `deleted`

#### [MountTargetId \(p. 166\)](#)

System-assigned mount target ID.

Type: String

[NetworkInterfaceId](#) (p. 166)

ID of the network interface that Amazon EFS created when it created the mount target.

Type: String

[OwnerId](#) (p. 166)

AWS account ID that owns the resource.

Type: String

[SubnetId](#) (p. 166)

ID of the mount target's subnet.

Type: String

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

### IncorrectFileSystemLifecycleState

Returned if the file system's lifecycle state is not "available".

HTTP Status Code: 409

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

### IpAddressInUse

Returned if the request specified an `IpAddress` that is already in use in the subnet.

HTTP Status Code: 409

### MountTargetConflict

Returned if the mount target would violate one of the specified restrictions based on the file system's existing mount targets.

HTTP Status Code: 409

### NetworkInterfaceLimitExceeded

The calling account has reached the limit for elastic network interfaces for the specific AWS Region. The client should try to delete some elastic network interfaces or get the account limit raised. For more information, see [Amazon VPC Limits](#) in the Amazon VPC User Guide (see the Network interfaces per VPC entry in the table).

HTTP Status Code: 409

NoFreeAddressesInSubnet

Returned if `IpAddress` was not specified in the request and there are no free IP addresses in the subnet.

HTTP Status Code: 409

SecurityGroupLimitExceeded

Returned if the size of `SecurityGroups` specified in the request is greater than five.

HTTP Status Code: 400

SecurityGroupNotFound

Returned if one of the specified security groups doesn't exist in the subnet's VPC.

HTTP Status Code: 400

SubnetNotFound

Returned if there is no subnet with ID `SubnetId` provided in the request.

HTTP Status Code: 400

UnsupportedAvailabilityZone

HTTP Status Code: 400

## Examples

### Example 1: Add a mount target to a file system

The following request creates a mount target for a file system. The request specifies values for only the required `FileSystemId` and `SubnetId` parameters. The request does not provide the optional `IpAddress` and `SecurityGroups` parameters. For `IpAddress`, the operation uses one of the available IP addresses in the specified subnet. And, the operation uses the default security group associated with the VPC for the `SecurityGroups`.

#### Sample Request

```
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-e2a6438b"}
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: c3616af3-33fa-40ad-ae0d-d3895a2c3a1f
Content-Type: application/json
Content-Length: 252

{
  "MountTargetId": "fsmt-55a4413c",
```

```
"NetworkInterfaceId": "eni-d95852af",  
"FileSystemId": "fs-e2a6438b",  
"LifeCycleState": "available",  
"SubnetId": "subnet-748c5d03",  
"OwnerId": "231243201240",  
"IpAddress": "172.31.22.183"  
}
```

## Example 2: Add a mount target to a file system

The following request specifies all the request parameters to create a mount target.

### Sample Request

```
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-47a2c22e",  
  "SubnetId": "subnet-fd04ff94",  
  "IpAddress": "10.0.2.42",  
  "SecurityGroups": [  
    "sg-1a2b3c4d"  
  ]  
}
```

### Sample Response

```
HTTP/1.1 200 OK  
x-amzn-RequestId: c3616af3-33fa-40ad-ae0d-d3895a2c3a1f  
Content-Type: application/json  
Content-Length: 252  
  
{  
  "OwnerId": "251839141158",  
  "MountTargetId": "fsmt-9a13661e",  
  "FileSystemId": "fs-47a2c22e",  
  "SubnetId": "subnet-fd04ff94",  
  "LifeCycleState": "available",  
  "IpAddress": "10.0.2.42",  
  "NetworkInterfaceId": "eni-1bcb7772"  
}
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)



- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## CreateTags

Creates or overwrites tags associated with a file system. Each tag is a key-value pair. If a tag key specified in the request already exists on the file system, this operation overwrites its value with the value provided in the request. If you add the `Name` tag to your file system, Amazon EFS returns it in the response to the [DescribeFileSystems](#) (p. 181) operation.

This operation requires permission for the `elasticfilesystem:CreateTags` action.

## Request Syntax

```
POST /2015-02-01/create-tags/FileSystemId HTTP/1.1
Content-type: application/json

{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

## URI Request Parameters

The request requires the following URI parameters.

[FileSystemId](#) (p. 171)

ID of the file system whose tags you want to modify (String). This operation modifies the tags only, not the file system.

## Request Body

The request accepts the following data in JSON format.

[Tags](#) (p. 171)

Array of `Tag` objects to add. Each `Tag` object is a key-value pair.

Type: Array of [Tag](#) (p. 208) objects

Required: Yes

## Response Syntax

```
HTTP/1.1 204
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

## Example

### Create tags on a file system

The following request creates three tags ("key1", "key2", and "key3") on the specified file system.

#### Sample Request

```
POST /2015-02-01/create-tags/fs-e2a6438b HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160
```

```
{
  "Tags": [
    {
      "Value": "value1",
      "Key": "key1"
    },
    {
      "Value": "value2",
      "Key": "key2"
    },
    {
      "Value": "value3",
      "Key": "key3"
    }
  ]
}
```

#### Sample Response

```
HTTP/1.1 204 no content
x-amzn-RequestId: c3616af3-33fa-40ad-ae0d-d3895a2c3a1f
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## DeleteFileSystem

Deletes a file system, permanently severing access to its contents. Upon return, the file system no longer exists and you can't access any contents of the deleted file system.

You can't delete a file system that is in use. That is, if the file system has any mount targets, you must first delete them. For more information, see [DescribeMountTargets](#) (p. 185) and [DeleteMountTarget](#) (p. 176).

### Note

The `DeleteFileSystem` call returns while the file system state is still `deleting`. You can check the file system deletion status by calling the [DescribeFileSystems](#) (p. 181) operation, which returns a list of file systems in your account. If you pass file system ID or creation token for the deleted file system, the [DescribeFileSystems](#) (p. 181) returns a 404 `FileSystemNotFound` error.

This operation requires permissions for the `elasticfilesystem:DeleteFileSystem` action.

## Request Syntax

```
DELETE /2015-02-01/file-systems/FileSystemId HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

[FileSystemId](#) (p. 174)

ID of the file system you want to delete.

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 204
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### FileSystemInUse

Returned if a file system has mount targets.

HTTP Status Code: 409

FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

## Example

### Delete a file system

The following example sends a DELETE request to the `file-systems` endpoint (`elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems/fs-47a2c22e`) to delete a file system whose ID is `fs-47a2c22e`.

#### Sample Request

```
DELETE /2015-02-01/file-systems/fs-47a2c22e HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T233021Z
Authorization: <...>
```

#### Sample Response

```
HTTP/1.1 204 No Content
x-amzn-RequestId: a2d125b3-7ebd-4d6a-ab3d-5548630bff33
Content-Length: 0
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## DeleteMountTarget

Deletes the specified mount target.

This operation forcibly breaks any mounts of the file system by using the mount target that is being deleted, which might disrupt instances or applications using those mounts. To avoid applications getting cut off abruptly, you might consider unmounting any mounts of the mount target, if feasible. The operation also deletes the associated network interface. Uncommitted writes might be lost, but breaking a mount target using this operation does not corrupt the file system itself. The file system you created remains. You can mount an EC2 instance in your VPC by using another mount target.

This operation requires permissions for the following action on the file system:

- `elasticfilesystem:DeleteMountTarget`

### Note

The `DeleteMountTarget` call returns while the mount target state is still `deleting`. You can check the mount target deletion by calling the [DescribeMountTargets \(p. 185\)](#) operation, which returns a list of mount target descriptions for the given file system.

The operation also requires permissions for the following Amazon EC2 action on the mount target's network interface:

- `ec2:DeleteNetworkInterface`

## Request Syntax

```
DELETE /2015-02-01/mount-targets/MountTargetId HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

[MountTargetId \(p. 176\)](#)

ID of the mount target to delete (String).

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 204
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.



## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### DependencyTimeout

The service timed out trying to fulfill the request, and the client should try the call again.

HTTP Status Code: 504

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

### MountTargetNotFound

Returned if there is no mount target with the specified ID found in the caller's account.

HTTP Status Code: 404

## Example

### Remove a file system's mount target

The following example sends a DELETE request to delete a specific mount target.

#### Sample Request

```
DELETE /2015-02-01/mount-targets/fsmt-9a13661e HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T232908Z
Authorization: <...>
```

#### Sample Response

```
HTTP/1.1 204 No Content
x-amzn-RequestId: 76787670-2797-48ee-a34f-fce2ce122fef
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)

- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## DeleteTags

Deletes the specified tags from a file system. If the `DeleteTags` request includes a tag key that does not exist, Amazon EFS ignores it and doesn't cause an error. For more information about tags and related restrictions, see [Tag Restrictions](#) in the AWS Billing and Cost Management User Guide.

This operation requires permissions for the `elasticfilesystem:DeleteTags` action.

## Request Syntax

```
POST /2015-02-01/delete-tags/FileSystemId HTTP/1.1
Content-type: application/json

{
  "TagKeys": [ "string" ]
}
```

## URI Request Parameters

The request requires the following URI parameters.

[FileSystemId](#) (p. 179)

ID of the file system whose tags you want to delete (String).

## Request Body

The request accepts the following data in JSON format.

[TagKeys](#) (p. 179)

List of tag keys to delete.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

## Response Syntax

```
HTTP/1.1 204
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

**BadRequest**

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

## Example

### Delete tags from a file system

The following request deletes the tag `key2` from the tag set associated with the file system.

#### Sample Request

```
POST /2015-02-01/delete-tags/fs-e2a6438b HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215123Z
Authorization: <...>
Content-Type: application/json
Content-Length: 223

{
  "TagKeys": [
    "key2"
  ]
}
```

#### Sample Response

```
HTTP/1.1 204 No Content
x-amzn-RequestId: ec08ae47-3409-49f3-9e90-64a5f981bb2b
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## DescribeFileSystems

Returns the description of a specific Amazon EFS file system if either the file system `CreationToken` or the `FileSystemId` is provided. Otherwise, it returns descriptions of all file systems owned by the caller's AWS account in the AWS Region of the endpoint that you're calling.

When retrieving all file system descriptions, you can optionally specify the `MaxItems` parameter to limit the number of descriptions in a response. Currently, this number is automatically set to 10. If more file system descriptions remain, Amazon EFS returns a `NextMarker`, an opaque token, in the response. In this case, you should send a subsequent request with the `Marker` request parameter set to the value of `NextMarker`.

To retrieve a list of your file system descriptions, this operation is used in an iterative process, where `DescribeFileSystems` is called first without the `Marker` and then the operation continues to call it with the `Marker` parameter set to the value of the `NextMarker` from the previous response until the response has no `NextMarker`.

The order of file systems returned in the response of one `DescribeFileSystems` call and the order of file systems returned across the responses of a multi-call iteration is unspecified.

This operation requires permissions for the `elasticfilesystem:DescribeFileSystems` action.

## Request Syntax

```
GET /2015-02-01/file-systems?  
CreationToken=CreationToken&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems  
HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

### [CreationToken](#) (p. 181)

(Optional) Restricts the list to the file system with this creation token (String). You specify a creation token when you create an Amazon EFS file system.

Length Constraints: Minimum length of 1. Maximum length of 64.

### [FileSystemId](#) (p. 181)

(Optional) ID of the file system whose description you want to retrieve (String).

### [Marker](#) (p. 181)

(Optional) Opaque pagination token returned from a previous `DescribeFileSystems` operation (String). If present, specifies to continue the list from where the returning call had left off.

### [MaxItems](#) (p. 181)

(Optional) Specifies the maximum number of file systems to return in the response (integer). Currently, this number is automatically set to 10.

Valid Range: Minimum value of 1.

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "FileSystems": [
    {
      "CreationTime": number,
      "CreationToken": "string",
      "Encrypted": boolean,
      "FileSystemId": "string",
      "KmsKeyId": "string",
      "LifecycleState": "string",
      "Name": "string",
      "NumberOfMountTargets": number,
      "OwnerId": "string",
      "PerformanceMode": "string",
      "ProvisionedThroughputInMibps": number,
      "SizeInBytes": {
        "Timestamp": number,
        "Value": number
      },
      "ThroughputMode": "string"
    }
  ],
  "Marker": "string",
  "NextMarker": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [FileSystems \(p. 182\)](#)

Array of file system descriptions.

Type: Array of [FileSystemDescription \(p. 202\)](#) objects

### [Marker \(p. 182\)](#)

Present if provided by caller in the request (String).

Type: String

### [NextMarker \(p. 182\)](#)

Present if there are more file systems than returned in the response (String). You can use the `NextMarker` in the subsequent request to fetch the descriptions.

Type: String

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

## Example

### Retrieve list of ten file systems

The following example sends a GET request to the `file-systems` endpoint (`elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems`). The request specifies a `MaxItems` query parameter to limit the number of file system descriptions to 10.

#### Sample Request

```
GET /2015-02-01/file-systems?MaxItems=10 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191208Z
Authorization: <...>
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: ab5f2427-3ab3-4002-868e-30a77a88f739
Content-Type: application/json
Content-Length: 499
{
  "FileSystems":[
    {
      "OwnerId":"251839141158",
      "CreationToken":"MyFileSystem1",
      "FileSystemId":"fs-47a2c22e",
      "PerformanceMode" : "generalPurpose",
      "CreationTime":"1403301078",
      "LifeCycleState":"created",
      "Name":"my first file system",
      "NumberOfMountTargets":1,
      "SizeInBytes":{"
        "Value":29313417216,
        "Timestamp":"1403301078"
      }
    }
  ]
}
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)



- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## DescribeMountTargets

Returns the descriptions of all the current mount targets, or a specific mount target, for a file system. When requesting all of the current mount targets, the order of mount targets returned in the response is unspecified.

This operation requires permissions for the `elasticfilesystem:DescribeMountTargets` action, on either the file system ID that you specify in `FileSystemId`, or on the file system of the mount target that you specify in `MountTargetId`.

## Request Syntax

```
GET /2015-02-01/mount-targets?
FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems&MountTargetId=MountTargetId
HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

[FileSystemId](#) (p. 185)

(Optional) ID of the file system whose mount targets you want to list (String). It must be included in your request if `MountTargetId` is not included.

[Marker](#) (p. 185)

(Optional) Opaque pagination token returned from a previous `DescribeMountTargets` operation (String). If present, it specifies to continue the list from where the previous returning call left off.

[MaxItems](#) (p. 185)

(Optional) Maximum number of mount targets to return in the response. Currently, this number is automatically set to 10.

Valid Range: Minimum value of 1.

[MountTargetId](#) (p. 185)

(Optional) ID of the mount target that you want to have described (String). It must be included in your request if `FileSystemId` is not included.

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "Marker": "string",
  "MountTargets": [
    {
      "FileSystemId": "string",
      "IpAddress": "string",
```

```
        "LifecycleState": "string",  
        "MountTargetId": "string",  
        "NetworkInterfaceId": "string",  
        "OwnerId": "string",  
        "SubnetId": "string"  
    },  
    ],  
    "NextMarker": "string"  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### Marker (p. 185)

If the request included the `Marker`, the response returns that value in this field.

Type: String

### MountTargets (p. 185)

Returns the file system's mount targets as an array of `MountTargetDescription` objects.

Type: Array of `MountTargetDescription` (p. 206) objects

### NextMarker (p. 185)

If a value is present, there are more mount targets to return. In a subsequent request, you can provide `Marker` in your request with this value to retrieve the next set of mount targets.

Type: String

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

### MountTargetNotFound

Returned if there is no mount target with the specified ID found in the caller's account.

HTTP Status Code: 404

## Example

### Retrieve descriptions mount targets created for a file system

The following request retrieves descriptions of mount targets created for the specified file system.

#### Sample Request

```
GET /2015-02-01/mount-targets?FileSystemId=fs-47a2c22e HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191252Z
Authorization: <...>
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: ab5f2427-3ab3-4002-868e-30a77a88f739
Content-Type: application/json
Content-Length: 357

{
  "MountTargets": [
    {
      "OwnerId": "251839141158",
      "MountTargetId": "fsmt-9a13661e",
      "FileSystemId": "fs-47a2c22e",
      "SubnetId": "subnet-fd04ff94",
      "LifeCycleState": "added",
      "IpAddress": "10.0.2.42",
      "NetworkInterfaceId": "eni-1bcb7772"
    }
  ]
}
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## DescribeMountTargetSecurityGroups

Returns the security groups currently in effect for a mount target. This operation requires that the network interface of the mount target has been created and the lifecycle state of the mount target is not `deleted`.

This operation requires permissions for the following actions:

- `elasticfilesystem:DescribeMountTargetSecurityGroups` action on the mount target's file system.
- `ec2:DescribeNetworkInterfaceAttribute` action on the mount target's network interface.

## Request Syntax

```
GET /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

[MountTargetId](#) (p. 188)

ID of the mount target whose security groups you want to retrieve.

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "SecurityGroups": [ "string" ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[SecurityGroups](#) (p. 188)

Array of security groups.

Type: Array of strings

Array Members: Maximum number of 5 items.

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### IncorrectMountTargetState

Returned if the mount target is not in the correct state for the operation.

HTTP Status Code: 409

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

### MountTargetNotFound

Returned if there is no mount target with the specified ID found in the caller's account.

HTTP Status Code: 404

## Example

### Retrieve security groups in effect for a file system

The following example retrieves the security groups that are in effect for the network interface associated with a mount target.

#### Sample Request

```
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: <...>
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 088fb0b4-0c1d-4af7-9de1-933207fbdb46
Content-Length: 57

{
  "SecurityGroups" : [
    "sg-188d9f74"
  ]
}
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)



## DescribeTags

Returns the tags associated with a file system. The order of tags returned in the response of one `DescribeTags` call and the order of tags returned across the responses of a multi-call iteration (when using pagination) is unspecified.

This operation requires permissions for the `elasticfilesystem:DescribeTags` action.

## Request Syntax

```
GET /2015-02-01/tags/FileSystemId?Marker=Marker&MaxItems=MaxItems HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

`FileSystemId` (p. 191)

ID of the file system whose tag set you want to retrieve.

`Marker` (p. 191)

(Optional) Opaque pagination token returned from a previous `DescribeTags` operation (String). If present, it specifies to continue the list from where the previous call left off.

`MaxItems` (p. 191)

(Optional) Maximum number of file system tags to return in the response. Currently, this number is automatically set to 10.

Valid Range: Minimum value of 1.

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "Marker": "string",
  "NextMarker": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### Marker (p. 191)

If the request included a `Marker`, the response returns that value in this field.

Type: String

#### NextMarker (p. 191)

If a value is present, there are more tags to return. In a subsequent request, you can provide the value of `NextMarker` as the value of the `Marker` parameter in your next request to retrieve the next set of tags.

Type: String

#### Tags (p. 191)

Returns tags associated with the file system as an array of `Tag` objects.

Type: Array of [Tag \(p. 208\)](#) objects

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

## Example

### Retrieve tags associated with a file system

The following request retrieves tags (key-value pairs) associated with the specified file system.

#### Sample Request

```
GET /2015-02-01/tags/fs-e2a6438b/ HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215404Z
Authorization: <...>
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: f264e454-7859-4f15-8169-1c0d5b0b04f5
Content-Type: application/json
Content-Length: 288
```

```
{
  "Tags": [
    {
      "Key": "Name",
      "Value": "my first file system"
    },
    {
      "Key": "Fleet",
      "Value": "Development"
    },
    {
      "Key": "Developer",
      "Value": "Alice"
    }
  ]
}
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## ModifyMountTargetSecurityGroups

Modifies the set of security groups in effect for a mount target.

When you create a mount target, Amazon EFS also creates a new network interface. For more information, see [CreateMountTarget](#) (p. 164). This operation replaces the security groups in effect for the network interface associated with a mount target, with the `SecurityGroups` provided in the request. This operation requires that the network interface of the mount target has been created and the lifecycle state of the mount target is not `deleted`.

The operation requires permissions for the following actions:

- `elasticfilesystem:ModifyMountTargetSecurityGroups` action on the mount target's file system.
- `ec2:ModifyNetworkInterfaceAttribute` action on the mount target's network interface.

## Request Syntax

```
PUT /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
Content-type: application/json

{
  "SecurityGroups": [ "string" ]
}
```

## URI Request Parameters

The request requires the following URI parameters.

[MountTargetId](#) (p. 194)

ID of the mount target whose security groups you want to modify.

## Request Body

The request accepts the following data in JSON format.

[SecurityGroups](#) (p. 194)

Array of up to five VPC security group IDs.

Type: Array of strings

Array Members: Maximum number of 5 items.

Required: No

## Response Syntax

```
HTTP/1.1 204
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### IncorrectMountTargetState

Returned if the mount target is not in the correct state for the operation.

HTTP Status Code: 409

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

### MountTargetNotFound

Returned if there is no mount target with the specified ID found in the caller's account.

HTTP Status Code: 404

### SecurityGroupLimitExceeded

Returned if the size of `SecurityGroups` specified in the request is greater than five.

HTTP Status Code: 400

### SecurityGroupNotFound

Returned if one of the specified security groups doesn't exist in the subnet's VPC.

HTTP Status Code: 400

## Example

### Replace a mount target's security groups

The following example replaces security groups in effect for the network interface associated with a mount target.

#### Sample Request

```
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"
  ]
}
```

### Sample Response

```
HTTP/1.1 204 No Content
x-amzn-RequestId: 088fb0b4-0c1d-4af7-9de1-933207fbdb46
```

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

# UpdateFileSystem

Updates the throughput mode or the amount of provisioned throughput of an existing file system.

## Request Syntax

```
PUT /2015-02-01/file-systems/FileSystemId HTTP/1.1
Content-type: application/json

{
  "ProvisionedThroughputInMibps": number,
  "ThroughputMode": "string"
}
```

## URI Request Parameters

The request requires the following URI parameters.

[FileSystemId \(p. 197\)](#)

The ID of the file system that you want to update.

## Request Body

The request accepts the following data in JSON format.

[ProvisionedThroughputInMibps \(p. 197\)](#)

(Optional) The amount of throughput, in MiB/s, that you want to provision for your file system. If you're not updating the amount of provisioned throughput for your file system, you don't need to provide this value in your request.

Type: Double

Valid Range: Minimum value of 0.0.

Required: No

[ThroughputMode \(p. 197\)](#)

(Optional) The throughput mode that you want your file system to use. If you're not updating your throughput mode, you don't need to provide this value in your request.

Type: String

Valid Values: `bursting` | `provisioned`

Required: No

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "CreationTime": number,
```

```
"CreationToken": "string",
"Encrypted": boolean,
"FileSystemId": "string",
"KmsKeyId": "string",
"LifecycleState": "string",
"Name": "string",
"NumberOfMountTargets": number,
"OwnerId": "string",
"PerformanceMode": "string",
"ProvisionedThroughputInMibps": number,
"SizeInBytes": {
  "Timestamp": number,
  "Value": number
},
"ThroughputMode": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### [CreationTime \(p. 197\)](#)

Time that the file system was created, in seconds (since 1970-01-01T00:00:00Z).

Type: Timestamp

### [CreationToken \(p. 197\)](#)

Opaque string specified in the request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

### [Encrypted \(p. 197\)](#)

A Boolean value that, if true, indicates that the file system is encrypted.

Type: Boolean

### [FileSystemId \(p. 197\)](#)

ID of the file system, assigned by Amazon EFS.

Type: String

### [KmsKeyId \(p. 197\)](#)

The ID of an AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the encrypted file system.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

### [LifecycleState \(p. 197\)](#)

Lifecycle phase of the file system.

Type: String

Valid Values: `creating` | `available` | `updating` | `deleting` | `deleted`



#### [Name \(p. 197\)](#)

You can add tags to a file system, including a `Name` tag. For more information, see [CreateTags \(p. 171\)](#). If the file system has a `Name` tag, Amazon EFS returns the value in this field.

Type: String

Length Constraints: Maximum length of 256.

#### [NumberOfMountTargets \(p. 197\)](#)

Current number of mount targets that the file system has. For more information, see [CreateMountTarget \(p. 164\)](#).

Type: Integer

Valid Range: Minimum value of 0.

#### [OwnerId \(p. 197\)](#)

AWS account that created the file system. If the file system was created by an IAM user, the parent account to which the user belongs is the owner.

Type: String

#### [PerformanceMode \(p. 197\)](#)

The `PerformanceMode` of the file system.

Type: String

Valid Values: `generalPurpose` | `maxIO`

#### [ProvisionedThroughputInMibps \(p. 197\)](#)

The throughput, measured in MiB/s, that you want to provision for a file system. The limit on throughput is 1024 MiB/s. You can get these limits increased by contacting AWS Support. For more information, see [Amazon EFS Limits That You Can Increase](#) in the Amazon EFS User Guide.

Type: Double

Valid Range: Minimum value of 0.0.

#### [SizeInBytes \(p. 197\)](#)

Latest known metered size (in bytes) of data stored in the file system, in its `Value` field, and the time at which that size was determined in its `Timestamp` field. The `Timestamp` value is the integer number of seconds since 1970-01-01T00:00:00Z. The `SizeInBytes` value doesn't represent the size of a consistent snapshot of the file system, but it is eventually consistent when there are no writes to the file system. That is, `SizeInBytes` represents actual size only if the file system is not modified for a period longer than a couple of hours. Otherwise, the value is not the exact size that the file system was at any point in time.

Type: [FileSystemSize \(p. 205\)](#) object

#### [ThroughputMode \(p. 197\)](#)

The throughput mode for a file system. There are two throughput modes to choose from for your file system: bursting and provisioned. You can decrease your file system's throughput in Provisioned Throughput mode or change between the throughput modes as long as it's been more than 24 hours since the last decrease or throughput mode change.

Type: String

Valid Values: `bursting` | `provisioned`

## Errors

### BadRequest

Returned if the request is malformed or contains an error such as an invalid parameter value or a missing required parameter.

HTTP Status Code: 400

### FileSystemNotFound

Returned if the specified `FileSystemId` value doesn't exist in the requester's AWS account.

HTTP Status Code: 404

### IncorrectFileSystemLifecycleState

Returned if the file system's lifecycle state is not "available".

HTTP Status Code: 409

### InsufficientThroughputCapacity

Returned if there's not enough capacity to provision additional throughput. This value might be returned when you try to create a file system in provisioned throughput mode, when you attempt to increase the provisioned throughput of an existing file system, or when you attempt to change an existing file system from bursting to provisioned throughput mode.

HTTP Status Code: 503

### InternalServerError

Returned if an error occurred on the server side.

HTTP Status Code: 500

### ThroughputLimitExceeded

Returned if the throughput mode or amount of provisioned throughput can't be changed because the throughput limit of 1024 MiB/s has been reached.

HTTP Status Code: 400

### TooManyRequests

Returned if you don't wait at least 24 hours before changing the throughput mode, or decreasing the Provisioned Throughput value.

HTTP Status Code: 429

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)

- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## Data Types

The following data types are supported:

- [FileSystemDescription](#) (p. 202)
- [FileSystemSize](#) (p. 205)
- [MountTargetDescription](#) (p. 206)
- [Tag](#) (p. 208)

## FileSystemDescription

Description of the file system.

### Contents

#### CreationTime

Time that the file system was created, in seconds (since 1970-01-01T00:00:00Z).

Type: Timestamp

Required: Yes

#### CreationToken

Opaque string specified in the request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Required: Yes

#### Encrypted

A Boolean value that, if true, indicates that the file system is encrypted.

Type: Boolean

Required: No

#### FileSystemId

ID of the file system, assigned by Amazon EFS.

Type: String

Required: Yes

#### KmsKeyId

The ID of an AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the encrypted file system.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

#### LifeCycleState

Lifecycle phase of the file system.

Type: String

Valid Values: `creating` | `available` | `updating` | `deleting` | `deleted`

Required: Yes

#### Name

You can add tags to a file system, including a `Name` tag. For more information, see [CreateTags \(p. 171\)](#). If the file system has a `Name` tag, Amazon EFS returns the value in this field.

Type: String

Length Constraints: Maximum length of 256.

Required: No

#### NumberOfMountTargets

Current number of mount targets that the file system has. For more information, see [CreateMountTarget \(p. 164\)](#).

Type: Integer

Valid Range: Minimum value of 0.

Required: Yes

#### OwnerId

AWS account that created the file system. If the file system was created by an IAM user, the parent account to which the user belongs is the owner.

Type: String

Required: Yes

#### PerformanceMode

The `PerformanceMode` of the file system.

Type: String

Valid Values: `generalPurpose` | `maxIO`

Required: Yes

#### ProvisionedThroughputInMibps

The throughput, measured in MiB/s, that you want to provision for a file system. The limit on throughput is 1024 MiB/s. You can get these limits increased by contacting AWS Support. For more information, see [Amazon EFS Limits That You Can Increase](#) in the Amazon EFS User Guide.

Type: Double

Valid Range: Minimum value of 0.0.

Required: No

#### SizeInBytes

Latest known metered size (in bytes) of data stored in the file system, in its `Value` field, and the time at which that size was determined in its `Timestamp` field. The `Timestamp` value is the integer number of seconds since 1970-01-01T00:00:00Z. The `SizeInBytes` value doesn't represent the size of a consistent snapshot of the file system, but it is eventually consistent when there are no writes to the file system. That is, `SizeInBytes` represents actual size only if the file system is not modified for a period longer than a couple of hours. Otherwise, the value is not the exact size that the file system was at any point in time.

Type: [FileSystemSize \(p. 205\)](#) object

Required: Yes

#### ThroughputMode

The throughput mode for a file system. There are two throughput modes to choose from for your file system: bursting and provisioned. You can decrease your file system's throughput in Provisioned

Throughput mode or change between the throughput modes as long as it's been more than 24 hours since the last decrease or throughput mode change.

Type: String

Valid Values: `bursting` | `provisioned`

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

## FileSystemSize

Latest known metered size (in bytes) of data stored in the file system, in its `value` field, and the time at which that size was determined in its `timestamp` field. Note that the value does not represent the size of a consistent snapshot of the file system, but it is eventually consistent when there are no writes to the file system. That is, the value will represent the actual size only if the file system is not modified for a period longer than a couple of hours. Otherwise, the value is not necessarily the exact size the file system was at any instant in time.

### Contents

#### Timestamp

Time at which the size of data, returned in the `value` field, was determined. The value is the integer number of seconds since 1970-01-01T00:00:00Z.

Type: Timestamp

Required: No

#### Value

Latest known metered size (in bytes) of data stored in the file system.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

# MountTargetDescription

Provides a description of a mount target.

## Contents

### FileSystemId

ID of the file system for which the mount target is intended.

Type: String

Required: Yes

### IpAddress

Address at which the file system can be mounted by using the mount target.

Type: String

Required: No

### LifecycleState

Lifecycle state of the mount target.

Type: String

Valid Values: `creating` | `available` | `updating` | `deleting` | `deleted`

Required: Yes

### MountTargetId

System-assigned mount target ID.

Type: String

Required: Yes

### NetworkInterfaceId

ID of the network interface that Amazon EFS created when it created the mount target.

Type: String

Required: No

### OwnerId

AWS account ID that owns the resource.

Type: String

Required: No

### SubnetId

ID of the mount target's subnet.

Type: String

Required: Yes



## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

## Tag

A tag is a key-value pair. Allowed characters: letters, white space, and numbers, representable in UTF-8, and the following characters: + - = . \_ : /

## Contents

### Key

Tag key (String). The key can't start with `aws :`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

### Value

Value of the tag key.

Type: String

Length Constraints: Maximum length of 256.

Required: Yes

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

# Amazon EFS 的其他信息

您可以在下文中找到有关 Amazon EFS 的一些其他信息，包括仍支持但不一定建议使用的功能。

## 主题

- [使用 AWS Data Pipeline 备份 Amazon EFS 文件系统 \(p. 209\)](#)
- [在没有 EFS 挂载帮助程序的情况下挂载文件系统 \(p. 220\)](#)

## 使用 AWS Data Pipeline 备份 Amazon EFS 文件系统

如果您需要能够从 Amazon EFS 文件系统的意外更改或删除中恢复，我们建议您使用 [EFS 到 EFS 备份解决方案](#)。该解决方案适用于所有 AWS 区域中的所有 Amazon EFS 文件系统。它包括一个 AWS CloudFormation 模板以启动、配置和运行部署该解决方案所需的 AWS 服务。该解决方案遵循 AWS 的安全和可用性最佳实践。

您还可以使用 AWS Data Pipeline 备份 EFS 文件系统。在该备份解决方案中，您使用 AWS Data Pipeline 服务创建一个数据管道。该管道将数据从您的 Amazon EFS 文件系统（称为生产文件系统）复制到另一个 Amazon EFS 文件系统（称为备份文件系统）中。

该解决方案包含实施以下操作的 AWS Data Pipeline 模板：

- 基于您定义的计划（例如，每小时、每天、每周或每月）的自动化 EFS 备份。
- 自动轮换备份，在此情况下，会基于您希望保留的备份数将最旧的备份替换为最新的备份。
- 使用 rsync 进行更快的备份，它仅备份在两次备份之间进行的更改。
- 使用硬链接高效存储备份。硬链接是一个目录项，该目录项会将一个名称与文件系统中的文件相关联。通过设置硬链接，您可以通过任何备份执行完整数据还原，同时仅存储在两次备份之间进行的更改。

在设置备份解决方案后，此演练将向您演示如何访问备份以还原您的数据。此备份解决方案依赖于运行 GitHub 上托管的脚本，因此受 GitHub 可用性的约束。如果您希望消除该依赖性并在 Amazon S3 存储桶中托管脚本，请参阅 [在 Amazon S3 存储桶中托管 rsync 脚本 \(p. 218\)](#)。

## Important

该解决方案要求在与您的文件系统相同的 AWS 区域中使用 AWS Data Pipeline。由于在美国东部（俄亥俄州）中不支持 AWS Data Pipeline，因此，无法在此 AWS 区域中使用该解决方案。如果要使用该解决方案备份您的文件系统，我们建议您在其他支持的 AWS 区域之一中使用您的文件系统。

## 主题

- [使用 AWS Data Pipeline 的 Amazon EFS 备份的性能 \(p. 210\)](#)
- [使用 AWS Data Pipeline 的 Amazon EFS 备份的注意事项 \(p. 210\)](#)
- [使用 AWS Data Pipeline 的 Amazon EFS 备份假设 \(p. 210\)](#)
- [如何使用 AWS Data Pipeline 备份 Amazon EFS 文件系统 \(p. 211\)](#)

- [其他备份资源 \(p. 216\)](#)

## 使用 AWS Data Pipeline 的 Amazon EFS 备份的性能

在执行数据备份和还原时，您的文件系统性能会受到[Amazon EFS 性能 \(p. 78\)](#)的约束，包括基线和突发吞吐量容量。备份解决方案使用的吞吐量将计入您的总文件系统吞吐量。下表提供了适用于此解决方案的 Amazon EFS 文件系统和 Amazon EC2 实例大小的一些建议 (假定您的备份窗口为 15 分钟)。

EFS 大小 (平均文件大小为 30 MB)	每日更改量	剩余突发小时数	备份代理的最小数量
256GB	不到 25 GB	6.75	1 - m3.medium
512GB	不到 50 GB	7.75	1 - m3.large
1.0 TB	不到 75 GB	11.75	2 - m3.large*
1.5 TB	不到 125 GB	11.75	2 - m3.xlarge*
2.0 TB	不到 175 GB	11.75	3 - m3.large*
3.0 TB	不到 250 GB	11.75	4 - m3.xlarge*

\* 这些估计数字基于以下假设：存储在大小为 1 TB 或更大的 EFS 文件系统的数据经过组织整理，以便可将备份分散在多个备份节点中。多节点示例脚本基于 EFS 文件系统一级目录的内容跨节点分散备份负载。

例如，如果具有两个备份节点，一个节点备份位于一级目录中的所有偶数文件和目录。奇数节点为奇数文件和目录执行相同的操作。在另一个示例中，在 Amazon EFS 文件系统中具有 6 个目录和 4 个备份节点，第一个节点备份第一个和第 5 个目录。第二个节点备份第二个和第 6 个目录，第 3 个和第 4 个节点分别备份第 3 个和第 4 个目录。

## 使用 AWS Data Pipeline 的 Amazon EFS 备份的注意事项

在您决定是否实施使用 AWS Data Pipeline 的 Amazon EFS 备份解决方案时，请考虑以下因素：

- 这种 EFS 备份方法涉及多种 AWS 资源。对于此解决方案，您需要创建以下内容：
  - 一个生产文件系统以及一个包含生产文件系统的完整副本的备份文件系统。该系统还包含在备份轮换期间对数据进行的任何增量更改。
  - 执行还原和计划备份的 Amazon EC2 实例 (其生命周期由 AWS Data Pipeline 进行管理)。
  - 用于备份数据的一个定期计划的 AWS Data Pipeline。
  - 用于还原备份的 AWS Data Pipeline。

实施此解决方案后，将会在您的账户中对这些服务计费。有关更多信息，请参阅 [Amazon EFS](#)、[Amazon EC2](#) 和 [AWS Data Pipeline](#) 的定价页面。

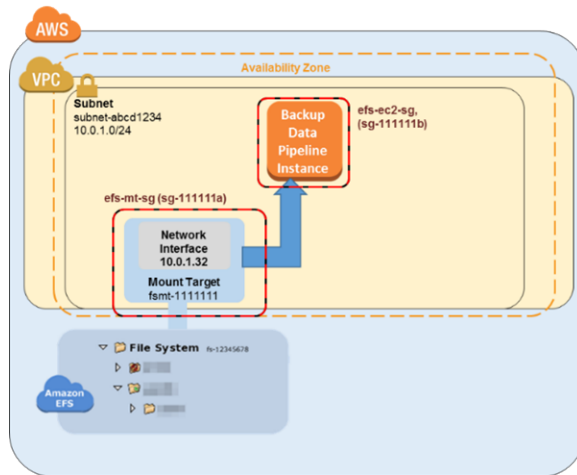
- 该解决方案不是脱机备份解决方案。为确保进行完全一致且完整的备份，请在备份发生时暂停到文件系统的任何文件写入或卸载文件系统。我们建议您在计划的停机时间或下班时间执行所有备份。

## 使用 AWS Data Pipeline 的 Amazon EFS 备份假设

本演练提供了多个假设，并声明了如下示例值：

- 在您开始操作之前，本演练假定您已完成[入门 \(p. 9\)](#)。
- 完成入门练习后，您有两个安全组、一个 VPC 子网和您要备份的文件系统的文件系统挂载目标。在本演练的其余部分中，您将使用以下示例值：
  - 您在本演练中备份的文件系统的 ID 是 fs-12345678。
  - 与挂载目标关联的文件系统的安全组称为 efs-mt-sg (sg-1111111a)。
  - 授权 Amazon EC2 实例连接到生产 EFS 挂载点的安全组名称为 efs-ec2-sg (sg-1111111b)。
  - VPC 子网的 ID 值为 subnet-abcd1234。
  - 您要备份的文件系统的源文件系统挂载目标 IP 地址为 10.0.1.32: /。
  - 该示例假定生产文件系统是一个内容管理系统，该系统提供平均大小为 30 MB 的媒体文件。

在以下初始设置图中反映了上述假设和示例。



## 如何使用 AWS Data Pipeline 备份 Amazon EFS 文件系统

请按照本节中的步骤使用 AWS Data Pipeline 备份或还原您的 Amazon EFS 文件系统。

### 主题

- [步骤 1：创建您的备份 Amazon EFS 文件系统 \(p. 211\)](#)
- [第 2 步：下载用于备份的 AWS Data Pipeline 模板 \(p. 212\)](#)
- [第 3 步：创建用于备份的数据管道 \(p. 212\)](#)
- [步骤 4：访问您的 Amazon EFS 备份 \(p. 213\)](#)

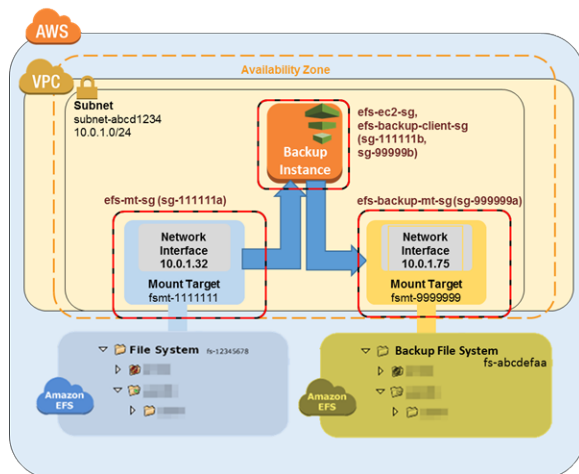
## 步骤 1：创建您的备份 Amazon EFS 文件系统

在本演练中，您将创建单独的安全组、文件系统和挂载点，以将您的备份与您的数据源分离。在该第一步中，您创建以下资源：

1. 首先，创建两个新安全组。备份挂载目标的示例安全组为 efs-backup-mt-sg (sg-9999999a)。用于访问挂载目标的 EC2 实例的示例安全组为 efs-backup-ec2-sg (sg-9999999b)。请记住，在您要备份的 EFS 卷所在的同一 VPC 中创建这些安全组。在该示例中为与 subnet-abcd1234 子网相关联的 VPC。有关创建安全组的更多信息，请参阅[创建安全组 \(p. 24\)](#)。
2. 接下来，创建备份 Amazon EFS 文件系统。在该示例中，文件系统 ID 为 fs-abcdefaa。有关创建文件系统的更多信息，请参阅[创建 Amazon Elastic File System \(p. 16\)](#)。

3. 最后，创建 EFS 备份文件系统的挂载点，并假定其值为 10.0.1.75:/。有关创建挂载目标的更多信息，请参阅[创建挂载目标](#) (p. 20)。

完成该第一步后，您的设置应类似于以下示例图。



## 第 2 步：下载用于备份的 AWS Data Pipeline 模板

AWS Data Pipeline 有助于您按指定间隔可靠地处理数据并在不同 AWS 计算和存储服务之间移动数据。通过使用 AWS Data Pipeline 控制台，您可以创建预配置的管道定义（称为模板）。您可以使用这些模板快速开始使用 AWS Data Pipeline。针对本演练提供了一个模板，以便更易于设置您的备份管道。

实施该模板后，它会创建一个数据管道，该数据管道会根据您指定的计划启动单个 Amazon EC2 实例，以将生产文件系统中的数据备份到备份文件系统中。该模板有许多占位符值。您可以在 AWS Data Pipeline 控制台的参数部分中为这些占位符提供匹配的值。从 GitHub 的 [1-Node-EFSBackupDataPipeline.json](#) 中下载用于备份的 AWS Data Pipeline 模板。

### Note

该模板还引用并运行一个脚本，以执行备份命令。您可以下载该脚本，然后再创建管道，以查看其用途。要查看该脚本，请从 GitHub 中下载 [efs-backup.sh](#)。此备份解决方案依赖于运行 GitHub 上托管的脚本，并受 GitHub 可用性的约束。如果您希望消除该依赖性并在 Amazon S3 存储桶中托管脚本，请参阅[在 Amazon S3 存储桶中托管 rsync 脚本 \(p. 218\)](#)。

### 第 3 步：创建用于备份的数据管道

采用下列步骤创建您的数据管道。

## 为 Amazon EFS 备份创建数据管道

1. 通过以下网址打开 AWS Data Pipeline 控制台：<https://console.aws.amazon.com/datapipeline/>。

## Important

确保在与您的 Amazon EFS 文件系统相同的 AWS 区域中工作。

2. 选择 **Create new pipeline**。
3. 添加名称和描述 ( 可选 ) 的值。
4. 对于源, 请选择导入定义, 然后选择加载本地文件。
5. 在文件资源管理器中, 导航到您在 [第 2 步: 下载用于备份的 AWS Data Pipeline 模板 \(p. 212\)](#) 中保存的模板, 然后选择打开。

6. 在参数中，提供您的备份和生产 EFS 文件系统的详细信息。

**Parameters**

Production EFS mount target IP address.	10.0.1.32/
Security group that can connect to the Production EFS mount point.	sg-1111111b
Interval for backups.	daily
Security group that can connect to the Backup EFS mount point.	sg-9999999b
Number of backups to retain.	7
Backup EFS mount target IP address.	10.0.1.75/
VPC subnet for your backup EC2 instance (ideally the same subnet used for the production EFS mount point).	subnet-1234abcd
Instance type for creating backups.	m3.medium
Name for the directory that will contain your backups.	backup-fs-12345678
Shell command to run.	wget https://raw.githubusercontent.com/aws-labs/data-pipeline-

7. 在计划中配置选项以定义您的 Amazon EFS 备份计划。示例中的备份每天运行一次，并且备份将保留一周时间。当备份保留时间达到七天时，它将被替换为下一个最旧的备份。

**Schedule**

**You can run your pipeline once or specify a schedule. [More](#)**

**Run**

☐ once on pipeline activation

☒ on a schedule

**Run every**  day(s)

**Starting**

☒ on pipeline activation

☐ 2016-05-28 02:46 UTC (Current time is 02:48 UTC)

YYYY-MM-DD HH.MM

**Ending**

☒ never

☐ after  occurrence(s)

☐ 2016-05-29 02:46 UTC (Current time is 02:48 UTC)

YYYY-MM-DD HH.MM

#### Note

我们建议您指定在非高峰时间进行备份。

8. (可选) 指定一个 Amazon S3 位置以存储管道日志，配置一个自定义 IAM 角色，或者添加标签以描述您的管道。
9. 在配置管道后，请选择激活。

现已配置并激活您的 Amazon EFS 备份数据管道。有关 AWS Data Pipeline 的更多信息，请参阅 [AWS Data Pipeline 开发人员指南](#)。在此阶段，您可以立即执行备份作为测试，也可以等待，直至在计划的时间执行备份。

## 步骤 4：访问您的 Amazon EFS 备份

现已创建并激活您的 Amazon EFS 备份，并按照您定义的计划运行该备份。该步骤简要说明了如何才能访问您的 EFS 备份。您的备份存储在采用以下格式创建的 EFS 备份文件系统中。



```
backup-efs-mount-target:/efs-backup-id/[backup interval].[0-backup retention]-->
```

以示例场景中的值为例，文件系统的备份位于 `10.1.0.75:/fs-12345678/daily.[0-6]` 中，其中 `daily.0` 是七个轮换备份中最新的备份，而 `daily.6` 是最旧的备份。

通过访问您的备份，您能够将数据还原到您的生产文件系统中。您可以选择还原整个文件系统，也可以选择还原各个文件。

## 步骤 4.1：还原整个 Amazon EFS 备份

还原 Amazon EFS 文件系统的备份副本需要另一个 AWS Data Pipeline，它与您在 [第 3 步：创建用于备份的数据管道 \(p. 212\)](#) 中创建的管道类似。但是，该还原管道与备份管道的方向相反。通常，不会计划自动开始执行这些还原。

与备份一样，可并行执行还原，以满足您的恢复时间目标。请记住，创建数据管道时，需要计划您希望其运行的时间。如果您选择激活后运行，则可以立即启动还原过程。建议您仅在需要执行还原操作，或已经想好具体的时段时，才创建还原管道。

突发容量由备份 EFS 和还原 EFS 使用。有关性能的更多信息，请参阅 [Amazon EFS 性能 \(p. 78\)](#)。以下步骤介绍了如何创建和实施您的还原管道。

### 创建用于 EFS 数据还原的数据管道

1. 下载用于从您的备份 EFS 文件系统中还原数据的数据管道模板。该模板可基于指定大小启动单个 Amazon EC2 实例。仅当您指定它启动时，它才会启动。从 GitHub 的 [1-Node-EFSRestoreDataPipeline.json](#) 中下载用于备份的 AWS Data Pipeline 模板。

#### Note

该模板还引用并运行一个脚本，以执行还原命令。您可以下载该脚本，然后再创建管道，以查看其用途。要查看该脚本，请从 GitHub 中下载 [efs-restore.sh](#)。

2. 通过以下网址打开 AWS Data Pipeline 控制台：<https://console.aws.amazon.com/datapipeline/>。

#### Important

确保在与您的 Amazon EFS 文件系统和 Amazon EC2 相同的 AWS 区域中工作。

3. 选择 Create new pipeline。
4. 添加名称和描述（可选）的值。
5. 对于源，请选择导入定义，然后选择加载本地文件。
6. 在文件资源管理器中，导航到您在 [步骤 1：创建您的备份 Amazon EFS 文件系统 \(p. 211\)](#) 中保存的模板，然后选择打开。
7. 在参数中，提供您的备份和生产 EFS 文件的详细信息。



The screenshot shows the 'Parameters' section of the AWS Data Pipeline console. It contains the following fields and values:

- Production EFS mount target IP address: 10.0.1.32/
- Security group that can connect to the Production EFS mount point: sg-1111111b
- Instance type for performing the restore: m3.large
- Security group that can connect to the Backup EFS mount point: sg-9999999b
- Name for the directory that already contains your backups: backup-fs-12345678
- Backup number to restore (0 = the most recent backup): 0
- Backup EFS mount target IP address: 10.0.1.75/
- Interval that you chose for the backup your going to restore: daily
- VPC subnet for your restoration EC2 instance (ideally the same subnet used for the backup EFS mount point): subnet-1234abcd

8. 由于您通常仅在需要时执行还原，因此，您可以计划在激活管道时运行一次还原。或者，计划在所选的将来时间执行一次性还原，例如，非高峰时段。
9. （可选）指定一个 Amazon S3 位置以存储管道日志，配置一个自定义 IAM 角色，或者添加标签以描述您的管道。
10. 在配置管道后，请选择激活。

现已配置并激活您的 Amazon EFS 还原数据管道。现在，在需要将备份还原到生产 EFS 文件系统时，您只需从 AWS Data Pipeline 控制台中激活该管道。有关更多信息，请参阅 [AWS Data Pipeline 开发人员指南](#)。

## 步骤 4.2：通过 Amazon EFS 备份还原各个文件

您可以通过启动 Amazon EC2 实例以临时挂载生产和备份 EFS 文件系统，来从您的 Amazon EFS 文件系统备份中还原文件。EC2 实例必须是两个 EFS 客户端安全组（在该示例中为 `efs-ec2-sg` 和 `efs-backup-clients-sg`）的成员。两个 EFS 挂载目标均可由此还原实例挂载。例如，恢复 EC2 实例可以创建以下挂载点。此处，`-o ro` 选项用于将备份 EFS 挂载为只读文件系统，以防止在尝试通过备份进行还原时意外修改备份。

```
mount -t nfs source-efs-mount-target:/ /mnt/data
```

```
mount -t nfs -o ro backup-efs-mount-target:/fs-12345678/daily.0 /mnt/backup>
```

挂载目标后，您可以使用 `cp -p` 命令将文件从 `/mnt/backup` 复制到终端的 `/mnt/data` 中的适当位置。例如，可以使用以下命令以递归方式复制整个主目录（及其文件系统权限）。

```
sudo cp -rp /mnt/backup/users/my_home /mnt/data/users/my_home
```

您可以运行以下命令以还原单个文件。

```
sudo cp -p /mnt/backup/user/my_home/.profile /mnt/data/users/my_home/.profile
```

### Warning

在手动还原各个数据文件时，请格外小心以免意外修改备份本身。否则，您可能会损坏该备份。

## 其他备份资源

此演练中提供的备份解决方案使用 AWS Data Pipeline 模板。用于[第 2 步：下载用于备份的 AWS Data Pipeline 模板 \(p. 212\)](#)和[步骤 4.1：还原整个 Amazon EFS 备份 \(p. 214\)](#)的模板均使用单个 Amazon EC2 实例来执行其工作。但是，对于您可以运行以备份或还原 Amazon EFS 文件系统中的数据的并行实例的数量，并没有真正的限制。在该主题中，您可以找到指向为多个 EC2 实例配置的其他 AWS Data Pipeline 模板的链接，您可以下载这些模板并将其用于您的备份解决方案。您还可以找到如何修改模板以包含其他实例的说明。

### 主题

- [使用其他模板 \(p. 216\)](#)
- [添加更多备份实例 \(p. 216\)](#)
- [添加更多还原实例 \(p. 217\)](#)
- [在 Amazon S3 存储桶中托管 rsync 脚本 \(p. 218\)](#)

## 使用其他模板

您可以从 GitHub 中下载以下更多模板：

- [2-Node-EFSBackupPipeline.json](#) – 该模板启动两个并行 Amazon EC2 实例以备份生产 Amazon EFS 文件系统。
- [2-Node-EFSRestorePipeline.json](#) – 该模板可启动两个并行 Amazon EC2 实例，以还原您的生产 Amazon EFS 文件系统的备份。

## 添加更多备份实例

您可以将更多节点添加到本演练中使用的备份模板中。要添加节点，请修改 `2-Node-EFSBackupDataPipeline.json` 模板的以下部分。

### Important

如果您要使用更多节点，则不能在存储于顶级目录的文件名和目录中使用空格。如果使用，则不会备份或还原这些文件和目录。将按预期方式备份和还原至少位于顶级目录下一级的所有文件和子目录。

- 为要创建的每个其他节点创建一个额外的 `EC2Resource`（在该示例中，为第 4 个 EC2 实例）。

```
{
  "id": "EC2Resource4",
  "terminateAfter": "70 Minutes",
  "instanceType": "#{myInstanceType}",
  "name": "EC2Resource4",
  "type": "Ec2Resource",
  "securityGroupIds": [ "#{mySrcSecGroupID}", "#{myBackupSecGroupID}" ],
  "subnetId": "#{mySubnetID}",
  "associatePublicIpAddress": "true"
},
```

- 针对每个其他节点创建额外的数据管道活动（在这种情况下为活动 `BackupPart4`），确保配置以下部分：
  - 更新 `runsOn` 引用以指向以前创建的 `EC2Resource`（在以下示例中为 `EC2Resource4`）。
  - 增加最后两个 `scriptArgument` 值以等于每个节点负责的备份部分以及总节点数。对于以下示例中的“2”和“3”，第 4 个节点的备份部分为“3”，因为该示例中的模块逻辑需要从 0 开始计数。

```
{
```

```
"id": "BackupPart4",
"name": "BackupPart4",
"runsOn": {
  "ref": "EC2Resource4"
},
"command": "wget https://raw.githubusercontent.com/aws-labs/data-pipeline-samples/master/
samples/EFSBackup/efs-backup-rsync.sh\nchmod a+x efs-backup-rsync.sh\n./efs-backup-
rsync.sh $1 $2 $3 $4 $5 $6 $7",
"scriptArgument": [ "#{myEfsSource}", "#{myEfsBackup}", "#{myInterval}",
  "#{myRetainedBackups}", "#{myEfsID}", "3", "4" ],
"type": "ShellCommandActivity",
"dependsOn": {
  "ref": "InitBackup"
},
"stage": "true"
},
```

- 将全部现有 `scriptArgument` 值中的最后一个值增加为节点数 (在该示例中为 "4")。

```
{
  "id": "BackupPart1",
  ...
  "scriptArgument": [ "#{myEfsSource}", "#{myEfsBackup}", "#{myInterval}",
    "#{myRetainedBackups}", "#{myEfsID}", "1", "4" ],
  ...
},
{
  "id": "BackupPart2",
  ...
  "scriptArgument": [ "#{myEfsSource}", "#{myEfsBackup}", "#{myInterval}",
    "#{myRetainedBackups}", "#{myEfsID}", "2", "4" ],
  ...
},
{
  "id": "BackupPart3",
  ...
  "scriptArgument": [ "#{myEfsSource}", "#{myEfsBackup}", "#{myInterval}",
    "#{myRetainedBackups}", "#{myEfsID}", "0", "4" ],
  ...
},
```

- 更新 `FinalizeBackup` 活动并将新的备份活动添加到 `dependsOn` 列表中 (在此情况下为 `BackupPart4`)。

```
{
  "id": "FinalizeBackup", "name": "FinalizeBackup", "runsOn": { "ref":
    "EC2Resource1" }, "command": "wget
    https://raw.githubusercontent.com/aws-labs/data-pipeline-samples/master/samples/EFSBackup/
    efs-backup-end.sh\nchmod a+x
    efs-backup-end.sh\n./efs-backup-end.sh $1 $2", "scriptArgument": [ "#{myInterval}",
    "#{myEfsID}" ], "type": "ShellCommandActivity", "dependsOn": [ { "ref": "BackupPart1" },
    { "ref": "BackupPart2" }, { "ref": "BackupPart3" }, { "ref": "BackupPart4" } ], "stage":
    "true"
}
```

## 添加更多还原实例

您可以向本演练中使用的还原模板添加节点。要添加节点，请修改 `2-Node-EFSRestorePipeline.json` 模板的以下部分。

- 为要创建的每个其他节点创建一个额外的 `EC2Resource` (此处为名为 `EC2Resource3` 的第 3 个 `EC2` 实例)。

```
{
  "id": "EC2Resource3",
  "terminateAfter": "70 Minutes",
  "instanceType": "#{myInstanceType}",
  "name": "EC2Resource3",
  "type": "Ec2Resource",
  "securityGroupIds": [ "#{mySrcSecGroupID}", "#{myBackupSecGroupID}" ],
  "subnetId": "#{mySubnetID}",
  "associatePublicIpAddress": "true"
},
```

- 为每个其他节点创建一个额外的数据管道活动（此处为活动 RestorePart3）。确保配置以下部分：
  - 更新 runsOn 引用以指向以前创建的 EC2Resource（在该示例中为 EC2Resource3）。
  - 增加最后两个 scriptArgument 值以等于每个节点负责的备份部分以及总节点数。对于以下示例中的 "2" 和 "3"，第 4 个节点的备份部分为 "3"，因为该示例中的模块逻辑需要从 0 开始计数。

```
{
  "id": "RestorePart3",
  "name": "RestorePart3",
  "runsOn": {
    "ref": "EC2Resource3"
  },
  "command": "wget https://raw.githubusercontent.com/aws-labs/data-pipeline-samples/master/samples/EFSBackup/efs-restore-rsync.sh\nchmod a+x efs-restore-rsync.sh\n./efs-backup-rsync.sh $1 $2 $3 $4 $5 $6 $7",
  "scriptArgument": [ "#{myEfsSource}", "#{myEfsBackup}", "#{myInterval}",
    "#{myBackup}", "#{myEfsID}", "2", "3" ],
  "type": "ShellCommandActivity",
  "dependsOn": {
    "ref": "InitBackup"
  },
  "stage": "true"
},
```

- 将全部现有 scriptArgument 值中的最后一个值增加为节点数（在该示例中为 "3"）。

```
{
  "id": "RestorePart1",
  ...
  "scriptArgument": [ "#{myEfsSource}", "#{myEfsBackup}", "#{myInterval}",
    "#{myBackup}", "#{myEfsID}", "1", "3" ],
  ...
},
{
  "id": "RestorePart2",
  ...
  "scriptArgument": [ "#{myEfsSource}", "#{myEfsBackup}", "#{myInterval}",
    "#{myBackup}", "#{myEfsID}", "0", "3" ],
  ...
},
```

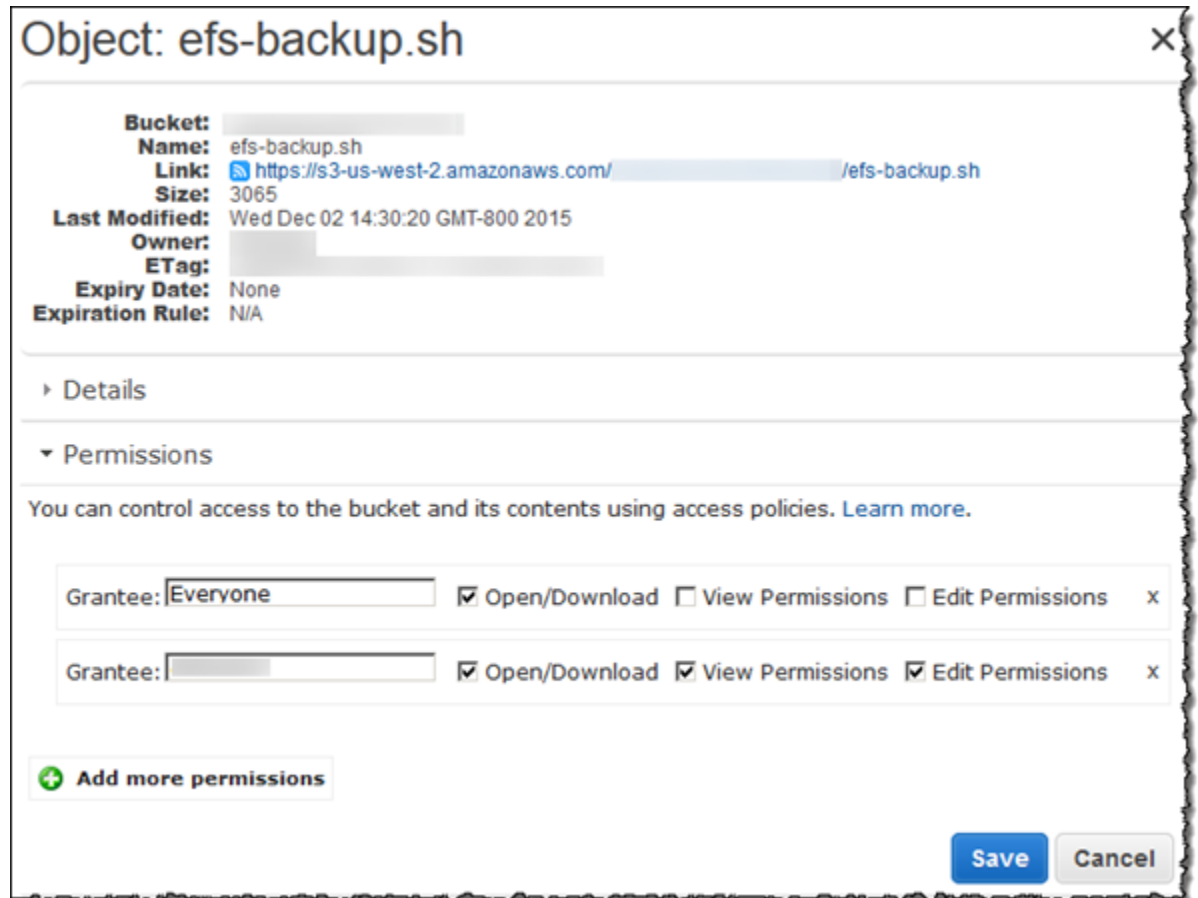
## 在 Amazon S3 存储桶中托管 rsync 脚本

此备份解决方案依赖于运行 Internet 上 GitHub 存储库中托管的 rsync 脚本。因此，此备份解决方案受 GitHub 存储库可用性的约束。该要求意味着，如果 GitHub 存储库删除这些脚本，或者 GitHub 网站脱机，按照上述方法实施的备份解决方案将无法正常工作。

如果您希望消除此 GitHub 依赖性，则可选择在您拥有的 Amazon S3 存储桶中托管这些脚本。您可以在下文中找到自行托管脚本所需的步骤。

在您自己的 Amazon S3 存储桶中托管 rsync 脚本

1. 注册 AWS – 如果您已经拥有 AWS 账户，请直接跳到下一步。否则，请参阅[注册 AWS \(p. 7\)](#)。
2. 创建 AWS Identity and Access Management 用户 – 如果已具有 IAM 用户，请直接跳到下一步。否则，请参阅[创建 IAM 用户 \(p. 7\)](#)。
3. 创建 Amazon S3 存储桶 – 如果您已经有一个要在其中托管 rsync 脚本的存储桶，请直接跳到下一步。否则，请参阅 Amazon Simple Storage Service 入门指南 中的[创建存储桶](#)。
4. 下载 rsync 脚本和模板 – 在 GitHub 的 [EFSBackup 文件夹](#) 中下载所有 rsync 脚本和模板。记下您在计算机上下载这些文件的位置。
5. 将 rsync 脚本上传至您的 S3 存储桶 – 有关如何将对象上传至 S3 存储桶的说明，请参阅 Amazon Simple Storage Service 入门指南 中的[将对象添加至存储桶](#)。
6. 更改上传的 rsync 脚本的权限以允许每个人打开/下载这些脚本。有关如何更改针对 S3 存储桶中对象的权限的说明，请参阅 Amazon Simple Storage Service 控制台用户指南 中的[编辑对象权限](#)。



7. 更新您的模板 – 将 wget 语句中的 shellCmd 参数修改为指向您放置启动脚本的 Amazon S3 存储桶。保存更新后的模板，然后在您按照[第 3 步：创建用于备份的数据管道 \(p. 212\)](#)中介绍的步骤执行操作时使用该模板。

#### Note

我们建议您限制 Amazon S3 存储桶的访问权限，以包含为该备份解决方案激活 AWS Data Pipeline 的 IAM 账户。有关更多信息，请参阅 Amazon Simple Storage Service 控制台用户指南 中的[编辑存储桶权限](#)。

现在您将托管此备份解决方案的 rsync 脚本，且您的备份不再依赖于 GitHub 可用性。

# 在没有 EFS 挂载帮助程序的情况下挂载文件系统

## Note

在本节中，您可以了解如何在没有 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，我们建议您使用 amazon-efs-utils 软件包。有关更多信息，请参阅 [使用 amazon-efs-utils 工具 \(p. 34\)](#)。

您可以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

您必须创建、配置和启动相关的 AWS 资源，然后才可以挂载文件系统。有关详细说明，请参阅 [Amazon Elastic File System 入门 \(p. 9\)](#)。

## 主题

- [NFS 支持 \(p. 220\)](#)
- [安装 NFS 客户端 \(p. 221\)](#)
- [使用 DNS 名称在 Amazon EC2 上挂载 \(p. 222\)](#)
- [使用 IP 地址挂载 \(p. 222\)](#)
- [自动挂载 \(p. 223\)](#)

## 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) 和 NFSv4.0 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。

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 2015.09 或更高版本
- RHEL 7.3 或更高版本
- 具有内核 2.6.32-696 或更高版本的 RHEL 6.9
- 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

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

## AMI 和内核版本故障排除

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 [解决 AMI 和内核问题 \(p. 97\)](#)。

## 安装 NFS 客户端

要在 Amazon EC2 实例上挂载 Amazon 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. 连接到您的 EC2 实例。连接到实例时，请注意以下情况：

- 要从运行 Mac OS 或 Linux 的计算机连接到您的实例，请在安全 Shell (SSH) 客户端中使用 `-i` 选项和私有密钥路径指定 `.pem` 文件。
- 要从运行 Windows 的计算机连接到您的实例，可以使用 MindTerm 或 PuTTY。如果您计划使用 PuTTY，则需要安装它并按以下过程将 `.pem` 文件转换为 `.ppk` 文件。

有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的以下主题：

- [使用 PuTTY 从 Windows 连接到 Linux 实例](#)
- [使用 SSH 连接到 Linux 实例](#)

密钥文件不能对 SSH 公开可见。您可以使用 `chmod 400 ###.pem` 命令设置这些权限。有关更多信息，请参阅[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
```

如果使用自定义内核（即，如果构建自定义 AMI），您需要至少包含 NFSv4.1 客户端内核模块和相应的 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 \(p. 222\)](#)
- [使用 IP 地址挂载 \(p. 222\)](#)
- [自动挂载 Amazon EFS 文件系统 \(p. 61\)](#)



## 使用 DNS 名称在 Amazon EC2 上挂载

您可以使用 DNS 名称在 Amazon EC2 实例上挂载 Amazon EFS 文件系统。可使用文件系统的 DNS 名称或挂载目标的 DNS 名称来完成该操作。

- 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，您可以使用以下约定构造该名称。

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

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

```
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
```

- 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
```

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。

有关支持 Amazon EFS 的 AWS 区域列表，请参阅 AWS General Reference 中的 [Amazon Elastic File System](#)。

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 DHCP 选项集。
- 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](#) 中的查看您的 EC2 实例的 DNS 主机名。

### 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。在该等待时间内，将在文件系统所在的 AWS 区域中完全传播 DNS 记录。

## 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。按 IP 地址挂载适用于禁用了 DNS 的环境，例如，禁用了 DNS 主机名的 VPC 以及使用 ClassicLink 的 EC2-Classic 实例挂载。有关 ClassicLink 的更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 [ClassicLink](#)。

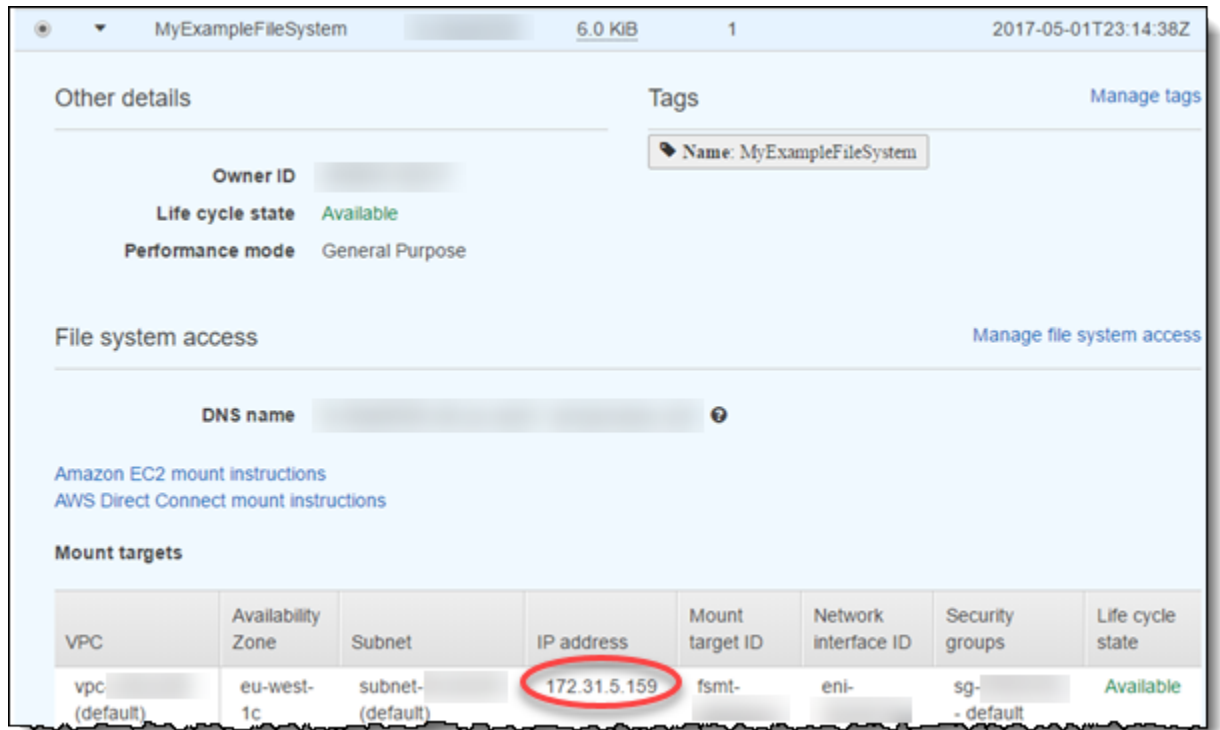
对于配置为默认使用 DNS 名称挂载文件系统的应用程序，您还可以将使用挂载目标 IP 地址挂载文件系统配置为回退选项。当连接到挂载目标 IP 地址时，EC2 实例应使用连接实例所在的同一可用区中的挂载目标 IP 地址进行挂载。

您可通过控制台使用以下步骤获取您的 EFS 文件系统的挂载目标 IP 地址。



获取您的 EFS 文件系统的挂载目标 IP 地址

1. 通过 <https://console.aws.amazon.com/efs/> 打开 Amazon Elastic File System 控制台。
2. 对于文件系统，请选择您的 EFS 文件系统的名称值。
3. 在挂载目标表中，指定要用于将 EFS 文件系统挂载到 EC2 实例的可用区。
4. 记下与所选的可用区关联的 IP 地址。



您可以在 mount 命令中指定挂载目标的 IP 地址，如下所示。

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

## 在 AWS CloudFormation 中使用 IP 地址挂载

您还可以在 AWS CloudFormation 模板中使用 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](#)，以获取 GitHub 上的社区提供配置文件。

## 自动挂载

在每次重启挂载了 Amazon EFS 文件系统的 Amazon EC2 实例时，您可以使用 `fstab` 文件自动挂载该文件系统。您可以通过两种方法设置自动挂载。您可在首次连接到 EC2 实例后更新该实例中的 `/etc/fstab` 文件，也可以在创建 EC2 实例时配置自动挂载 EFS 文件系统。

## 将现有 EC2 实例更新为自动挂载

要在 Amazon EC2 实例重启时自动重新挂载 Amazon EFS 文件系统目录，您可以使用 `fstab` 文件。`fstab` 文件包含有关文件系统和命令 `mount -a` 的信息，该命令在实例启动期间运行，并挂载 `fstab` 文件列出的文件系统。

### Note

在您可以更新 EC2 实例的 `/etc/fstab` 文件之前，请确保您已经创建了 Amazon EFS 文件系统，并且已连接到 Amazon EC2 实例。有关更多信息，请参阅 Amazon EFS 入门练习中的 [第 2 步：创建您的 Amazon EFS 文件系统 \(p. 12\)](#)。

### 更新 EC2 实例中的 `/etc/fstab` 文件

1. 连接到您的 EC2 实例，然后在编辑器中打开 `/etc/fstab` 文件。
2. 将以下行添加到 `/etc/fstab` 文件中。

```
mount-target-DNS:/ efs-mount-point nfs4
nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,_netdev,noresvport 0
0
```

如果要在不同的可用区 (AZ) 中的 EC2 实例之间复制 `/etc/fstab` 文件内容，我们建议您使用文件系统 DNS 名称。如果使用挂载目标 DNS 名称，请不要在 AZ 之间复制 `/etc/fstab` 文件。如果这样做，对于具有挂载目标的每个可用区，每个文件系统将具有唯一的 DNS 名称。有关 DNS 名称的更多信息，请参阅 [使用 DNS 名称在 Amazon EC2 上挂载 \(p. 222\)](#)。

3. 保存对文件所做的更改。

您的 EC2 实例现已配置为每次重启时都挂载 EFS 文件系统。

### Note

如果需要启动您的 Amazon EC2 实例而不考虑挂载的 Amazon 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。

添加到 `/etc/fstab` 文件的代码行将设置以下内容。

字段	描述
<code>mount-target-DNS:/</code>	您要挂载的文件系统的域名服务器 (DNS) 名称。用于挂载 EFS 文件系统子目录的 <code>mount</code> 命令中将会使用该值。
<code>efs-mount-point</code>	EFS 文件系统在 EC2 实例上的挂载点。
<code>nfs4</code>	文件系统的类型。对于 EFS，此类型始终是 <code>nfs4</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： <ul style="list-style-type: none"><li>• <code>nfsvers</code> – 指定要使用的 NFS 版本。建议将 4.1 用作该选项的值。</li><li>• <code>rsz</code> – 定义数据块的大小，用于在您的客户端与云中的文件系统之间读取数据。建议将 1048576 用作该选项的值。</li><li>• <code>wsz</code> – 定义数据块的大小，用于在您的客户端与云中的文件系统之间写入数据。建议将 1048576 用作该选项的值。</li><li>• <code>hard</code> – 指定使用文件系统上某个文件的本地应用程序在 Amazon EFS 暂时不可用的情况下，应停止并等待该文件系统恢复在线状态。</li><li>• <code>timeo</code> – 指定时长 (单位为 0.1 秒)，即 NFS 客户端在重试向云中的文件系统发送请求之前等待响应的的时间。建议将 600 分秒用作该选项的值。</li></ul>

字段	描述
	<ul style="list-style-type: none"><li>• <code>retrans</code> – 指定 NFS 客户端应重试请求的次数。建议将 2 用作该选项的值。</li><li>• <code>_netdev</code> – 它用于禁止 Amazon EC2 实例的内核在实例具有网络连接之前挂载文件系统。</li><li>• <code>noresvport</code> – 如果使用该选项，在重新建立网络连接时，NFS 客户端将使用新的传输控制协议 (TCP) 源端口。使用新端口有助于确保在网络恢复事件后具有不间断的可用性。</li></ul> <p>有关更多信息，请参阅 <a href="#">其他挂载注意事项 (p. 63)</a>。</p>
0	非零值表示应由 <code>dump</code> 备份文件系统。对于 EFS，该值应为 0。
0	<code>fsck</code> 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 <code>fsck</code> 不应在启动时运行。

# 文档历史记录

- API 版本：2015-02-01
- 文档最新更新时间：2018 年 7 月 12 日

下表介绍了 2018 年 7 月之后对 Amazon Elastic File System 用户指南 的一些重要更改。如需有关文档更新的通知，您可以订阅 RSS 源。

update-history-change	update-history-description	update-history-date
<a href="#">预配置吞吐量模式简介 (p. 226)</a>	您现在可以使用新的预配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 <a href="http://docs.aws.amazon.com/efs/latest/ug/throughput-modes.html">http://docs.aws.amazon.com/efs/latest/ug/throughput-modes.html</a> 。	July 12, 2018
<a href="#">添加了额外的 AWS 区域支持 (p. 226)</a>	Amazon EFS 现在可供亚太区域（东京）AWS 区域中的所有用户使用。	July 11, 2018

下表介绍了 2018 年 7 月之前对 Amazon Elastic File System 用户指南 的一些重要更改。

更改	描述	更改日期
添加了额外的 AWS 区域支持	Amazon EFS 现在可供亚太区域（首尔）AWS 区域中的所有用户使用。	2018 年 5 月 30 日
添加了 CloudWatch 指标数学支持	指标数学使您可以查询多个 CloudWatch 指标，并使用数学表达式基于这些指标创建新的时间序列。有关更多信息，请参阅 <a href="#">将指标数学与 Amazon EFS 一起使用 (p. 71)</a> 。	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 amazon-efs-utils 不会产生额外费用，您可以从 GitHub 下载这些工具。有关更多信息，请参阅 <a href="#">使用 amazon-efs-utils 工具 (p. 34)</a> 。  同样，在该版本中，Amazon EFS 现在支持加密通过传输层安全性 (TLS) 隧道传输的数据。有关更多信息，请参阅 <a href="#">在 EFS 中加密数据和元数据 (p. 86)</a> 。	2018 年 4 月 4 日
更新了每个 AWS 区域的文件系统数限制	Amazon EFS 增加了所有 AWS 区域中的所有账户的文件系统数限制。有关更多信息，请参阅 <a href="#">资源限制 (p. 91)</a> 。	2018 年 3 月 15 日
添加了额外的 AWS 区域支持	Amazon EFS 现在可供美国西部（加利福尼亚北部）AWS 区域中的所有用户使用。	2018 年 3 月 14 日
Amazon EFS 文件同步 (EFS 文件同步)	Amazon EFS 现在支持使用 EFS 文件同步从本地数据中心或云将文件复制到 Amazon EFS。EFS 文件同步将使用网络文件系统 (NFS) 版本 3 或 NFS 版本 4 访问的文件系统复制到 Amazon EFS 文件系统。要了解其用法，请参阅 <a href="#">Amazon EFS 文件同步 (p. 28)</a> 。	2017 年 11 月 22 日

更改	描述	更改日期
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 <a href="#">在 EFS 中加密数据和元数据 (p. 86)</a> 。	2017 年 8 月 14 日
添加了额外的区域支持	Amazon EFS 现在可供欧洲（法兰克福）区域中的所有用户使用。	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 <a href="#">使用 DNS 名称在 Amazon EC2 上挂载 (p. 222)</a> 。	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 <a href="#">管理文件系统标签 (p. 45)</a> 。	2016 年 8 月 29 日
通用版	Amazon EFS 现已面向 美国东部（弗吉尼亚北部）、美国西部（俄勒冈）和 欧洲（爱尔兰）区域中的所有用户全面提供。	2016 年 6 月 28 日
文件系统限制提升	每个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 的第一个版本。	2015 年 5 月 26 日