

调整卷大小后扩展 Linux 文件系统

PDF ([ec2-ug.pdf#recognize-expanded-volume-linux](#))

RSS ([amazon-ec2-release-notes.rss](#))

在增加 EBS 卷的大小后 ([./requesting-ebs-volume-modifications.html](#))，您必须使用特定于文件系统的命令来将文件系统扩展到较大大小。一旦卷进入 `optimizing` 状态，您即可调整文件系统的大小。

⚠重要

在扩展包含有用数据的文件系统之前，最佳实践是创建卷的快照（如果您需要回滚您的更改）。有关更多信息，请参阅[创建 Amazon EBS 快照](#) ([./ebs-creating-snapshot.html](#))。如果您的 Linux AMI 使用 MBR 分区方案，您的引导卷大小限制为最高 2 TiB。有关更多信息，请参阅[针对 Linux 卷的要求](#) ([./modify-volume-requirements.html#linux-volumes](#)) 和[针对 EBS 卷的大小和配置的限制](#) ([./volume_constraints.html](#))。

在 Linux 上扩展文件系统的过程如下：

1. 您的 EBS 卷可能有一个包含文件系统和数据的分区。增加卷的大小不会增加分区的大小。在调整大小后的卷上扩展文件系统之前，请检查卷是否具有必须扩展到卷的新大小的分区。
2. 可使用特定于文件系统的命令将每个文件系统调整为新的卷容量。

有关扩展 Windows 文件系统的信息，请参阅 *Amazon EC2 用户指南（适用于 Windows 实例）* 中的[调整卷大小后扩展 Windows 文件系统](#) (<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/recognize-expanded-volume-windows.html>)。

以下示例将引导您完成扩展 Linux 文件系统的过程。关于此处所示的文件系统和分区方案以外的其他文件系统和分区方案，请参阅相关文件系统和分区方案的文档以获取说明。

示例

- [示例：扩展 NVMe EBS 卷的文件系统 \(#extend-file-system-nvme\)](#)
- [示例：扩展 EBS 卷的文件系统 \(#extend-file-system\)](#)

示例：扩展 NVMe EBS 卷的文件系统

对于此示例，假设您有一个基于 [Nitro 系统](#) ([./instance-types.html#ec2-nitro-instances](#)) 构建的实例，例如 M5 实例。您将引导卷的大小从 8 GB 调整为 16 GB，并将一个附加卷的大小从 8 GB 调整为 30 GB。按照以下过程扩展调整大小后的卷的文件系统。

扩展 NVMe EBS 卷的文件系统

1. [连接到您的实例](#) ([./AccessingInstances.html](#))。
2. 要验证每个卷的文件系统，请使用 `df -hT` 命令。

```
[ec2-user ~]$ df -hT
```

以下是一个实例的示例输出，该实例具有带 XFS 文件系统的引导卷和带 XFS 文件系统的附加卷。命名约定 `/dev/nvme[0-26]n1` 指示卷作为 NVMe 块储存设备公开。

```
[ec2-user ~]$ df -hT
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/nvme0n1p1  xfs   8.0G  1.6G  6.5G  20% /
/dev/nvme1n1     xfs   8.0G   33M  8.0G   1% /data
...
```

3. 要检查卷是否具有必须扩展的分区，请使用 `lsblk` 命令显示有关附加到实例的 NVMe 块储存设备的信息。

```
[ec2-user ~]$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme1n1       259:0    0   30G  0 disk /data
nvme0n1       259:1    0   16G  0 disk
└─nvme0n1p1   259:2    0    8G  0 part /
└─nvme0n1p128 259:3    0    1M  0 part
```

此示例输出显示以下内容：

- 根卷 `/dev/nvme0n1` 具有一个分区 `/dev/nvme0n1p1`。当根卷的大小反映新大小 16 GB 时，分区的大小会反映原始大小 8 GB 并且必须先进行扩展，然后才能扩展文件系统。
- 卷 `/dev/nvme1n1` 没有分区。卷的大小反映新大小 30 GB。

4. 要在根卷上扩展分区，请使用以下 **growpart** 命令。请注意，设备名称和分区编号之间有空格。

```
[ec2-user ~]$ sudo growpart /dev/nvme0n1 1
```

5. (可选) 再次使用 **lsblk** 命令来验证分区是否反映增大后的卷大小。

```
[ec2-user ~]$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme1n1       259:0    0   30G  0 disk /data
nvme0n1       259:1    0   16G  0 disk
└─nvme0n1p1   259:2    0   16G  0 part /
└─nvme0n1p128 259:3    0    1M  0 part
```

6. 使用 **df -h** 命令验证每个卷的文件系统的大小。在此示例输出中，每个文件系统均反映 8 GB 的原始卷大小。

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p1  8.0G  1.6G  6.5G  20% /
/dev/nvme1n1    8.0G   33M  8.0G   1% /data
...
```

7. 使用 **xfs_growfs** 命令扩展每个卷上的 XFS 文件系统。在此示例中，`/` 和 `/data` 是 **df -h** 的输出中显示的卷挂载点。

```
[ec2-user ~]$ sudo xfs_growfs -d /
[ec2-user ~]$ sudo xfs_growfs -d /data
```

如果尚未安装 XFS 工具，可以按如下方式安装。

```
[ec2-user ~]$ sudo yum install xfsprogs
```

8. (可选) 再次使用 **df -h** 命令来验证每个文件系统是否反映增大后的卷大小。

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p1  16G  1.6G   15G  10% /
/dev/nvme1n1    30G   33M   30G   1% /data
...
```

示例：扩展 EBS 卷的文件系统

在本示例中，假设您将某个实例（例如 T2 实例）的引导卷的大小从 8 GB 调整为 16 GB，并将一个附加卷的大小从 8 GB 调整为 30 GB。按照以下过程扩展调整大小后的卷的文件系统。

扩展 EBS 卷的文件系统

1. [连接到您的实例 \(./AccessingInstances.html\)](#)。
2. 要验证每个卷使用的文件系统，请使用 **df -hT** 命令。

```
[ec2-user ~]$ df -hT
```

以下是一个实例的示例输出，该实例具有带 ext4 文件系统的引导卷和带 XFS 文件系统的附加卷。

```
[ec2-user ~]$ df -hT
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/xvda1      ext4  8.0G  1.9G  6.2G  24% /
/dev/xvdf1      xfs   8.0G   45M  8.0G   1% /data
...
```

3. 要检查卷是否具有必须扩展的分区，请使用 **lsblk** 命令显示有关附加到实例的块储存设备的信息。

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   16G  0 disk
└─xvda1     202:1    0    8G  0 part /
xvdf        202:80   0   30G  0 disk
└─xvdf1     202:81   0    8G  0 part /data
```

此示例输出显示以下内容：

- 根卷 `/dev/xvda` 具有一个分区 `/dev/xvda1`。当卷的大小为 16 GB 时，分区的大小仍为 8 GB 且必须进行扩展。
- 卷 `/dev/xvdf` 具有一个分区 `/dev/xvdf1`。当卷的大小为 30GB 时，分区的大小仍为 8 GB 且必须进行扩展。

4. 要在每个卷上扩展分区，请使用以下 **growpart** 命令。请注意，设备名称和分区编号之间有空格。

```
[ec2-user ~]$ sudo growpart /dev/xvda 1
[ec2-user ~]$ sudo growpart /dev/xvdf 1
```

5. （可选）再次使用 **lsblk** 命令来验证分区是否反映增大后的卷大小。

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   16G  0 disk
└─xvda1     202:1    0   16G  0 part /
xvdf        202:80   0   30G  0 disk
└─xvdf1     202:81   0   30G  0 part /data
```

6. 使用 **df -h** 命令验证每个卷的文件系统的大小。在此示例输出中，`/dev/xvda1` 和 `/dev/xvdf` 均反映卷的原始大小 8 GB。

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      8.0G  1.9G  6.2G  24% /
```

```
/dev/xvdf1      8.0G   45M   8.0G    1% /data
...
```

7. 使用 **resize2fs** 命令扩展每个卷上的文件系统。

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
[ec2-user ~]$ sudo resize2fs /dev/xvdf1
```

8. (可选) 再次使用 **df -h** 命令来验证每个文件系统是否反映增大后的卷大小。

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      16G   1.9G   14G   12% /
/dev/xvdf1      30G    45M   30G    1% /data
...
```

© 2020, Amazon Web Services, Inc. 或其附属公司。保留所有权利。