

볼륨 크기 조정 후 Linux 파일 시스템 확장

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

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

EBS 볼륨 크기를 늘리고 ([./requesting-ebs-volume-modifications.html](#)) 난 후에는 파일 시스템 관련 명령을 사용하여 파일 시스템의 크기를 늘려야 합니다. 볼륨이 `optimizing` 상태가 되자마자 파일 시스템 크기 조정을 시작할 수 있습니다.

⚠️ 중요

중요한 데이터가 포함된 파일 시스템을 확장하려면 먼저 변경 내용을 롤백해야 할 경우를 대비하여 파일 시스템이 저장된 볼륨 스냅샷을 생성하는 것이 바람직합니다. 자세한 내용은 [Amazon EBS 스냅샷 생성 \(./ebs-creating-snapshot.html\)](#) 단원을 참조하십시오. Linux AMI가 MBR 파티셔닝 체계를 사용하는 경우 부팅 볼륨 크기는 2TiB로 제한됩니다. 자세한 내용은 [Linux 볼륨에 대한 요구 사항 \(./modify-volume-requirements.html#linux-volumes\)](#) 및 [EBS 볼륨의 크기 및 구성에 대한 제약 \(./volume_constraints.html\)](#) 단원을 참조하십시오.

Linux에서 파일 시스템을 확장하는 프로세스는 다음과 같습니다.

1. EBS 볼륨에 파일 시스템과 데이터를 포함하는 파티션이 있을 수 있습니다. 볼륨 크기를 늘려도 파티션 크기가 늘어나지 않습니다. 크기가 조정된 볼륨에서 파일 시스템을 확장하기 전에 볼륨에 볼륨 크기를 늘려야 하는 파티션이 있는지 여부를 확인하십시오.
2. 파일 시스템 관련 명령을 사용하여 새 볼륨 용량에 맞게 파일 시스템의 크기를 조정합니다.

Windows 파일 시스템 확장에 대한 자세한 내용은 *Windows* 인스턴스용 *Amazon EC2* 사용 설명서의 [볼륨 크기 조정 후 Windows 파일 시스템 확장](#) (<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/recognize-expanded-volume-windows.html>) 을 참조하십시오.

다음 예제에서는 Linux 파일 시스템을 확장하는 프로세스를 안내합니다. 여기에 표시된 것과 다른 파일 시스템 및 파티셔닝 체계에 대한 자세한 내용은 해당 파일 시스템 및 파티셔닝 체계 설명서를 참조하세요.

예제:

- 예제: NVMe EBS 볼륨의 파일 시스템 확장 ([#extend-file-system-nvme](#))
- 예제: EBS 볼륨의 파일 시스템 확장 ([#extend-file-system](#))

예제: NVMe EBS 볼륨의 파일 시스템 확장

이 예제에서는 M5 인스턴스와 같이 [Nitro 시스템 \(./instance-types.html#ec2-nitro-instances\)](#) 에 빌드된 인스턴스가 있다고 가정합니다. 부팅 볼륨의 크기를 8GB에서 16GB로 조정하고 추가 볼륨의 크기를 8GB에서 30GB로 조정했습니다. 크기가 조정된 볼륨의 파일 시스템을 확장하려면 다음 절차를 따르세요.

NVMe EBS 볼륨의 파일 시스템을 확장하려면

1. [인스턴스에 연결합니다 \(./AccessingInstances.html\)](#) .
2. 각 볼륨의 파일 시스템을 확인하려면 `df -hT` 명령을 사용합니다.

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

다음은 XFS 파일 시스템에 부팅 볼륨과 추가 볼륨을 가지고 있는 인스턴스의 예제 출력을 보여줍니다. 명령 규칙 `/dev/nvme[0-26]n1`은 볼륨이 NVMe 블록 디바이스로 표시됨을 나타냅니다.

```
[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. 볼륨에 확장해야 하는 파티션이 있는지 확인하려면 **lsblk** 명령을 사용하여 인스턴스에 연결된 NVMe 블록 디바이스에 대한 정보를 표시합니다.

```
[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
```

이 예제 출력은 다음 내용을 나타냅니다.

- 루트 볼륨 `/dev/nvme0n1`에는 `/dev/nvme0n1p1`라는 파티션이 있습니다. 루트 볼륨에 새 크기인 16GB가 반영되는 동안 파티션의 크기에 원래 크기인 8GB가 반영되기 때문에 파일 시스템을 확장하려면 먼저 파티션 크기를 늘려야 합니다.
 - 볼륨 `/dev/nvme1n1`에는 파티션이 없습니다. 볼륨 크기에 새 크기 30GB가 반영됩니다.
4. 루트 볼륨에서 파티션을 확장하려면 다음 **growpart** 명령을 사용합니다. 디바이스 이름과 파티션 번호 사이에 공백이 있다는 점에 유의하십시오.

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

5. (선택 사항) 다시 **lsblk** 명령을 사용하여 파티션에 늘어난 볼륨 크기가 반영되었는지 확인합니다.

```
[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. **df -h** 명령을 사용하여 각 볼륨에 대한 파일 시스템의 크기를 확인합니다. 이 예제 출력에서는 두 파일 시스템 모두에 원래 볼륨 크기인 8GB가 반영되었습니다.

```
[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. [XFS 파일 시스템] **xfs_growfs** 명령을 사용하여 각 볼륨에서 파일 시스템을 확장합니다. 이 예제에서 `/` 및 `/data`는 **df -h**의 출력에 나와 있는 볼륨 탑재 지점입니다.

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

XFS 도구가 아직 설치되지 않은 경우 다음과 같이 설치할 수 있습니다.

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

8. [ex4 파일 시스템] **resize2fs** 명령을 사용하여 각 볼륨에서 파일 시스템을 확장합니다.

```
[ec2-user ~]$ sudo resize2fs /dev/nvme0n1p1
[ec2-user ~]$ sudo resize2fs /dev/nvme1n1
```

9. [기타 파일 시스템] 지침은 파일 시스템 설명서를 참조하십시오.

10. (선택 사항) 다시 **df -h** 명령을 사용하여 각 파일 시스템에 늘어난 볼륨 크기가 반영되었는지 확인합니다.

```
[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
...
```

예제: EBS 볼륨의 파일 시스템 확장

이 예제에서 인스턴스(예: T2 인스턴스)의 부팅 볼륨 크기를 8GB에서 16GB로, 추가 볼륨의 크기를 8GB에서 30GB로 조정했다고 가정해봅시다. 크기가 조정된 볼륨의 파일 시스템을 확장하려면 다음 절차를 따르세요.

EBS 볼륨의 파일 시스템을 확장하려면

1. [인스턴스에 연결합니다 \(./AccessingInstances.html\)](#).
2. 각 볼륨의 사용 중인 파일 시스템을 확인하려면 **df -hT** 명령을 사용합니다.

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

다음은 ext4 파일 시스템에 부팅 볼륨을, XFS 파일 시스템에 추가 볼륨을 가지고 있는 인스턴스의 예제 출력을 보여줍니다.

```
[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. 볼륨에 확장해야 하는 파티션이 있는지 확인하려면 **lsblk** 명령을 사용하여 인스턴스에 연결된 블록 디바이스에 대한 정보를 표시합니다.

```
[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
```

이 예제 출력은 다음 내용을 나타냅니다.

- 루트 볼륨 /dev/xvda에는 /dev/xvda1라는 파티션이 있습니다. 볼륨 크기가 16GB일 때 파티션의 크기가 여전히 8GB이기 때문에 파티션 크기를 늘려야 합니다.
- 볼륨 /dev/xvdf에는 /dev/xvdf1라는 파티션이 있습니다. 볼륨 크기가 30GB일 때 파티션의 크기가 여전히 8GB이기 때문에 파티션 크기를 늘려야 합니다.

4. 각 볼륨에서 파티션을 확장하려면 다음 **growpart** 명령을 사용합니다. 디바이스 이름과 파티션 번호 사이에 공백이 있다는 점에 유의하십시오.

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

5. (선택 사항) 다시 **lsblk** 명령을 사용하여 파티션에 늘어난 볼륨 크기가 반영되었는지 확인합니다.

```
[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. **df -h** 명령을 사용하여 각 볼륨에 대한 파일 시스템의 크기를 확인합니다. 이 예제 출력에서는 두 파일 시스템 모두에 원래 볼륨 크기인 8GB가 반영되었습니다.

```
[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. [XFS 볼륨] **xfs_growfs** 명령을 사용하여 각 볼륨에서 파일 시스템을 확장합니다. 이 예제에서 / 및 /data는 **df -h**의 출력에 나와 있는 볼륨 탑재 지점입니다.

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

XFS 도구가 아직 설치되지 않은 경우 다음과 같이 설치할 수 있습니다.

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

8. [ex4 볼륨] **resize2fs** 명령을 사용하여 각 볼륨에서 파일 시스템을 확장합니다.

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

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

9. [기타 파일 시스템] 지침은 파일 시스템 설명서를 참조하십시오.

10. (선택 사항) 다시 **df -h** 명령을 사용하여 각 파일 시스템에 늘어난 볼륨 크기가 반영되었는지 확인합니다.

```
[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. 또는 계열사. All rights reserved.