

*Note that several EC2 instances have already been pre-configured. Servers one through three are deployed in the US East (N. Virginia) region, though they span different Availability Zones. Servers one and two reside in the same zone, while server three is located in a different zone. Server four is in an entirely separate region. This setup demonstrates that for an EBS volume to be movable between instances, both the volume and the target instance must reside within the same Availability Zone.*

## **Creating and Attaching an EBS Volume**

Create an EBS volume for server one:

1. Identify the Availability Zone of server one (for example, US East 1a).
2. Navigate to the Volumes page on the AWS console and create a new volume:
  - Choose the desired volume type (default is GP2).
  - Set the volume size.
  - Ensure the volume is created in the same Availability Zone as server one.
  - Leave encryption disabled for this demonstration.

Once created, the volume's status will be "available."

You may assign a descriptive name to the volume—such as "demo volume." After naming it, select Actions and choose Attach Volume. The console will display EC2 instances in the same Availability Zone (server one and server two). Select server one and note that the device is designated as `/dev/sdf`. On newer Linux kernels, this device might be automatically renamed (for example, to `/dev/xvdf`).

## **Verifying and Formatting the Volume on Server One**

1. Open a terminal connected to server one.

List the block devices using:

```
lsblk
```

2. You should see the root device (commonly listed as xvda) and the new 10 GB device (likely listed as xvdf).

Verify if a filesystem exists on the new device with:

```
sudo file -s /dev/xvdf
```

3. If the output shows data, no filesystem exists yet, which is expected.

Create an XFS filesystem on the device:

```
sudo mkfs -t xfs /dev/xvdf
```

4. Verify that the filesystem has been created successfully by running:

```
sudo file -s /dev/xvdf
```

5. The expected output should mention "SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)."

Below is a sample terminal session demonstrating these steps:

```
$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda         202:0    0   8G  0 disk
├─xvda1      202:1    0    8G  0 part /
├─xvda127    259:0    0    1M  0 part
├─xvda128    259:1    0   10M  0 part
xvdf         202:80    0  10G  0 disk
```

```
$ sudo file -s /dev/xvdf
/dev/xvdf: data
```

```
$ sudo mkfs -t xfs /dev/xvdf
meta-data=/dev/xvdf          isize=512  agcount=4,
agsize=655360 blks
        =                    sectsz=512  attr=2,
projid32bit=1
        =                    finobt=1, sparse=1,
rmapbt=0
data      =                    bsize=4096
blocks=2621440, imaxpct=25
        =                    sunid=0 blks
naming    =                    version 2
log        =                    internal log    bsize=4096
blocks=16384, version=2
realtime  =                    none    extsz=4096
blocks=0, rtexts=0
```

```
$ sudo file -s /dev/xvdf
/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512,
v2 dirs)
```

---

## Mounting the Filesystem

### Temporary Mount

The mount operation demonstrated below is temporary. If the instance is rebooted, the mount will be lost unless it is added to `/etc/fstab` for persistence.

Create a directory to serve as the mount point (for this demo, we will use `/ebdemo`):

```
sudo mkdir /ebdemo
sudo mount /dev/xvdf /ebdemo
```

1. Confirm the mount with the command:

```
df -h
```

2. This output confirms that `/dev/xvdf` is mounted at `/ebdemo`.

---

## Updating `/etc/fstab` for Persistent Mounting

To ensure the volume stays mounted across reboots, follow these steps:

Retrieve the UUID of the EBS volume:

```
sudo blkid
```

Look for the entry corresponding to `/dev/xvdf`. For example, you might see:

```
/dev/xvdf: UUID="04fdc8e-3441-4518-986c-a32254c0e925"  
TYPE="xfs"
```

1. Edit the `/etc/fstab` file:

```
sudo vi /etc/fstab
```

2. Add the following line to the file, making sure to replace the UUID with your own:

```
UUID=04fdc8e-3441-4518-986c-a32254c0e925 /ebsdemo xfs  
defaults,nofail 0 0
```

3. To apply the changes without rebooting, run:

```
sudo mount -a
```

4. Verify the mount and the volume's UUID:

```
df -k  
sudo blkid
```

---

## Testing the Mount with File Operations

To confirm that the EBS volume is working correctly:

Change to the mount directory and create a test file:

```
cd /ebdemo/  
echo "I made this file on server1" | sudo tee demo.txt  
sudo vi demo.txt # (Edit the file as needed)  
ls -l demo.txt
```

1. This confirms write access to the volume, ensuring that the data is stored on the EBS volume rather than on the instance's root storage.

---

## Detaching and Re-Attaching the EBS Volume

To demonstrate the detachable nature of an EBS volume, perform these steps:

Unmount the EBS volume from server one:

```
cd ~  
sudo umount /ebdemo
```

Verify the unmount by running:

```
df -k
```

1. The /ebdemo mount point should no longer be listed.
  2. In the AWS console, navigate to the list of EBS volumes. Select the volume, then choose Actions > Detach Volume.
  3. Confirm the detachment using the dialog provided:
  4. Reattach the volume to server two by selecting Attach Volume from the Actions menu and specifying server two as the target.
-

## Verifying the Volume on Server Two

On server two, perform the following steps to ensure that the reattached volume is functioning as expected:

Check the list of block devices:

```
lsblk  
sudo file -s /dev/xvdf
```

1. The device `/dev/xvdf` should still show the SGI XFS filesystem.

Create the mount point and mount the volume:

```
sudo mkdir /ebsdemo  
sudo mount /dev/xvdf /ebsdemo
```

- 2.

Navigate to the mount directory, list its contents, and check the test file:

```
cd /ebsdemo/  
ls  
cat demo.txt
```

3. If the file `demo.txt` exists and displays the expected content ("I made this file on server1"), the persistent data has been successfully preserved on the EBS volume independently of a particular EC2 instance.

On the current server where the volume is in use, unmount it by executing:

```
sudo umount /ebsdemo
```

After unmounting, verify that no file systems are mounted by listing block devices:

```
lsblk
```

Verify the file system type on the volume:

```
sudo file -s /dev/xvdf
```

You should see output similar to:

/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)

Next, create a mount point and mount the volume to verify the data inside:

```
sudo mkdir /ebsdemo
sudo mount /dev/xvdf /ebsdemo
cd /ebsdemo
ls
cat demo.txt
```

The file "demo.txt" should display:

I made this on server

Finally, unmount the volume again:

```
cd ..
```



```
sudo umount /ebdemo
```

Confirm that the volume is not mounted by checking:

```
df -h
```

## Detaching and Moving the Volume

With the volume unmounted, log into your AWS console and navigate to the EBS Volumes section. Select the volume, and ensure it is detached or in the process of detaching. Wait until its status changes to "available." Refresh the page if necessary.

Next, click the "Attach volume" button to try attaching the volume to Server Three. Note that only servers in the same availability zone (Server One and Server Two) will appear in the instance selection dropdown.

## Creating a Snapshot to Overcome Availability Zone Limitations

Because you cannot directly attach an EBS volume to an instance in a different availability zone, the solution is to create a snapshot of your volume. Follow these steps:

1. Select the volume.
2. Choose "Actions" > "Create snapshot."
3. (Optional) Add a description (e.g., "my snapshot") and confirm.

Creating a snapshot produces an exact copy of your volume's data.

Monitor the snapshot's progress in the Snapshots section. It may initially show a status like "pending" or a progress percentage until it turns "available."

## Creating a New Volume in a Different Availability Zone

Once the snapshot is available, follow these steps to create a new volume in the target availability zone:

1. Select the snapshot.
2. From the "Actions" dropdown, choose "Create volume from snapshot."
3. Change the availability zone to the target zone (e.g., US East 1B for Server Three). You can adjust volume type, size, and other settings as needed.
4. (Optional) Add tags (for example, name the volume "EBS clone").

After creating the new volume, confirm in the Volumes section that the clone appears in the correct availability zone.

Now, attach the new volume to Server Three:

Verifying on Server Three

Log into Server Three via your terminal and run the following commands to verify that the block device is attached and to confirm the file system:

```
lsblk
```

Check the filesystem on the new volume:

```
sudo file -s /dev/xvdf
```

Expected output:

```
/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
```

Then, create the mount point and mount the volume:

```
sudo mkdir /ebsdemo  
sudo mount /dev/xvdf /ebsdemo
```

Change directory to the mount point and verify the transferred data:

```
cd /ebsdemo/  
ls  
cat demo.txt
```

The file "demo.txt" should display:

I made this on server

This confirms that the original volume's data has been successfully cloned and attached to Server Three in a different availability zone.

## Moving Volumes Across Regions

If you need to migrate the volume to a completely different AWS region (for example, from Northern Virginia to Ohio), you cannot directly create a volume from a snapshot in another region. Instead, you must:

1. Copy the snapshot from the original region to the desired region.
2. Create a volume from the copied snapshot in the target region.

To copy the snapshot:

1. Select the snapshot.
2. Click "Actions" and select "Copy snapshot."
3. Provide a description such as "copy of my snapshot."
4. Specify the destination region (e.g., US East 2 for Ohio).

Once the snapshot copy is complete, switch to the target region (e.g., us-east-2), locate the snapshot in the Snapshots section, and then click "Create volume from snapshot." Choose the correct availability zone (e.g., us-east-2a) for your EC2 instance. After creation, the new volume will be available.

Finally, attach the newly created volume to your target EC2 instance:

This completes the process of migrating an EBS volume across different availability zones and regions.

Happy configuring!