

AWS EBS – Elastic Block Storage

EBS (Elastic Block Store) is a block-level storage service provided by AWS that you can use with EC2 instances. Think of it as a virtual hard disk you can attach and detach to your server (EC2) in the cloud.

- **Persistent Storage:** Your data remains even if the instance is stopped or terminated.
- **Attach/Detach:** You can attach EBS to any EC2 in the same Availability Zone.
- **Snapshot Support:** You can take backups (snapshots) and restore anytime.
- **Resize and Modify:** Change size, performance, and volume type even while in use.
- **Encryption:** Supports encryption for secure data.
- Pricing is based on storage you use.

Example Use Case:

You launch an EC2 instance but the default root disk(**8GB**) isn't enough. You create an EBS volume, attach it, format it, and now you have additional storage space available on your server.

Types of AWS EBS Volumes:

- **gp3** – General Purpose SSD (default, for most use cases like Web servers, dev/test, general use)
- **io2/io1** – High-performance SSD (for databases, high IOPS)
- **st1** – Throughput-optimized HDD (for big data, logs)
- **sc1** – Cold HDD (Lowest cost, for infrequently accessed data)

Now let's see how to practically attach an EBS volume to an EC2 instance and mount it

Step 1: Launch 3- EC2 Instances (2 Instances in same Availability Zones and 1 in different Az)

- Go to AWS Console → EC2 → **Launch Instance**
- Select AMI: Ubuntu or Amazon Linux
- Instance Type: t2.micro

- Key Pair: Create or use existing one
- Network default
- Add additional storage of 15 GiB to only one server.
- Click on create instance

Instances (2) [Info](#) Last updated 9 minutes ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[All states](#) < 1 > [Settings](#)

<input type="checkbox"/>	Name ↗	Instance ID	Instance state ↕	Instance type ↕	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	EBS-Lab	i-0c5c4082fb2bf119b	Running 🔗 🔍	t2.micro	2/2 checks passed	View alarms +	us-east-1d
<input type="checkbox"/>	EBS-Data Transfer	i-0207284dc61aea643	Running 🔗 🔍	t2.micro	2/2 checks passed	View alarms +	us-east-1d

Step2 : Connect to your EC2 instance and check if the additional EBS volume has been attached.

```
ubuntu@ip-172-31-87-171:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0    0  27.2M  1 loop /snap/amazon-ssm-agent/11320
loop1        7:1    0  73.9M  1 loop /snap/core22/1981
loop2        7:2    0  50.9M  1 loop /snap/snapd/24505
xvda        202:0    0    8G    0 disk
├─xvda1      202:1    0    7G    0 part /
├─xvda14     202:14   0    4M    0 part
├─xvda15     202:15   0   106M  0 part /boot/efi
└─xvda16     259:0    0   913M  0 part /boot
xvdb        202:16   0   15G    0 disk
```

Step3 : Format the volume with ext4 filesystem (We format the EBS volume with ext4 so the operating system can store and manage files on it.)

Command to format : `sudo mkfs -t ext4 /dev/xvdb`

```
ubuntu@ip-172-31-87-171:~$ sudo mkfs -t ext4 /dev/xvdb
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 3932160 4k blocks and 983040 inodes
Filesystem UUID: 0c73c689-b595-45c3-b517-9494fcf720ea
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step4 : Create a mount directory (We create a mount directory to provide a location in the file system where the EBS volume can be accessed)

Command to create directory : `sudo mkdir ebs-`

Step5 : Mount the volume (We mount the volume to link the formatted EBS volume to the mount directory so we can read and write data to it)

Command to mount: `sudo mount /dev/xvdb ebs-volume/`

- Check whether the additional EBS volume is attached and properly mounted to the directory

Command to Check : `df -h`

```
ubuntu@ip-172-31-87-171:~$  
ubuntu@ip-172-31-87-171:~$ sudo mkdir ebs-volume  
ubuntu@ip-172-31-87-171:~$ ls  
ebs-volume  
ubuntu@ip-172-31-87-171:~$  
ubuntu@ip-172-31-87-171:~$ sudo mount /dev/xvdb ebs-volume/  
ubuntu@ip-172-31-87-171:~$  
ubuntu@ip-172-31-87-171:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/root        6.8G  1.8G  5.0G  26% /  
tmpfs            479M    0  479M   0% /dev/shm  
tmpfs            192M  872K  191M   1% /run  
tmpfs            5.0M    0   5.0M   0% /run/lock  
/dev/xvda16      881M   86M  734M  11% /boot  
/dev/xvda15      105M   6.2M   99M   6% /boot/efi  
tmpfs            96M   12K   96M   1% /run/user/1000  
/dev/xvdb        15G   24K  14G   1% /home/ubuntu/ebs-volume
```

Step 6 : Create a file to test (We create a file to test in order to verify that the EBS volume is writable and mounted correctly)

Command to create file : `vi my-volume` (add some data into the file)

- Check the contents of the file

Command to check : `cat my-volume`

```
ubuntu@ip-172-31-87-171:~/ebs-volume$  
ubuntu@ip-172-31-87-171:~/ebs-volume$ vi My-volume  
ubuntu@ip-172-31-87-171:~/ebs-volume$ sudo vi my-volume  
ubuntu@ip-172-31-87-171:~/ebs-volume$  
ubuntu@ip-172-31-87-171:~/ebs-volume$ cat my-volume  
Hello I'm Arun Kumar Akula  
working as a DevOps Engineer.!
```

As of now, we have created the instance, formatted, and mounted the storage. Now, we will proceed to reattach the existing EBS volume to another EC2 instance.

How to Reattach an Existing EBS Volume to Another EC2 Instance

we need to unmount from the old server before mounting to the new one because EBS volumes can only be attached to one EC2 instance at a time in read/write mode.

- If you don't unmount and detach properly AWS **won't allow** attaching the volume to another EC2 until it's detached and You **risk data corruption**.

Step 1: Unmount the volume.

```
sudo umount /dev/xvdb
```

```
ubuntu@ip-172-31-87-171:~$  
ubuntu@ip-172-31-87-171:~$ sudo umount /dev/xvdb  
ubuntu@ip-172-31-87-171:~$  
ubuntu@ip-172-31-87-171:~$ lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS  
loop0        7:0    0  27.2M  1 loop /snap/amazon-ssm-agent/11320  
loop1        7:1    0  73.9M  1 loop /snap/core22/1981  
loop2        7:2    0  50.9M  1 loop /snap/snapd/24505  
xvda        202:0    0    8G   0 disk  
├─xvda1     202:1    0    7G   0 part /  
├─xvda14    202:14   0    4M   0 part  
├─xvda15    202:15   0  106M  0 part /boot/efi  
└─xvda16    259:0    0   913M  0 part /boot  
xvdb        202:16   0   15G   0 disk
```

Step 2: Detach the volume from Server 1

- Go to AWS Console → EC2 → Volumes
- Select the attached volume
- Click Actions → Detach Volume

- Wait until the volume state becomes "available"

The screenshot shows the AWS Management Console 'Volumes' page. At the top, there's a header with 'Volumes (1/3)' and an 'Info' link. Below this is a search bar and a table of volumes. The first volume, 'vol-08a4bd817bea74360', is selected. An 'Actions' menu is open, showing options like 'Modify volume', 'Create snapshot', 'Delete volume', 'Attach volume', 'Detach volume', 'Force detach volume', 'Manage auto-enabled I/O', 'Manage tags', and 'Fault injection'. The 'Attach volume' option is highlighted. To the right of the menu, there's a 'Create volume' button and a list of recent events.

	Name	Volume ID	Type	Size	IOPS	Throughput
<input checked="" type="checkbox"/>		vol-08a4bd817bea74360	gp3	15 GiB	3000	125
<input type="checkbox"/>		vol-03974f2d97b35e079	gp3	8 GiB	3000	125
<input type="checkbox"/>		vol-023606280fc582ff9	gp3	8 GiB	3000	125

Volume ID: vol-08a4bd817bea74360

Step : Attach the Volume to Server 2

- Select the available volume
- Click Actions → Attach Volume

This screenshot is similar to the previous one, but the 'Attach volume' option in the 'Actions' menu is highlighted with a black box. The volume 'vol-08a4bd817bea74360' is still selected in the table.

- Choose Server 2 from the instance list
- Set device name (e.g., /dev/xvdb)

Basic details

Volume ID

vol-08a4bd817bea74360

Availability Zone

us-east-1d

Instance

i-0207284dc61aea643
(EBS-Data Transfer) (running)

Only instances in the same Availability Zone as the selected volume are displayed.

Device name

/dev/xvdb

Recommended device names for Linux: /dev/sda1 for root volume, /dev/sd[f-p] for data volumes.

ⓘ Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel

Attach volume

- Now you can see that the volume is attached to your EC2 instance.

Instances (1/2) Info Last updated 11 minutes ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) [All states](#) < 1 > [Settings](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	EBS-Lab	i-0c5c4082fb2bf119b	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d
<input checked="" type="checkbox"/>	EBS-Data Transfer	i-0207284dc61aea643	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d

i-0207284dc61aea643 (EBS-Data Transfer) [Settings](#) [Refresh](#)

[/dev/sda1](#) EBS [disabled](#)

▼ **Block devices**

Filter block devices

	Volume ID	Device name	Volume size (GiB)	Volume State	Attachment status	Attachment time	Encry
<input type="checkbox"/>	vol-03974fd97b35e079	/dev/sda1	8	In-use	Attached	2025/06/27 17:05 GMT+5:30	No
<input type="checkbox"/>	vol-08a4bd817bea74360	/dev/xvdb0	15	In-use	Attached	2025/06/27 17:14 GMT+5:30	No

Step 5: Connect to Server 2 using SSH

- Create mount directory

```
Mkdir ebs-volume-2
```

Step 6 : Mount the volume

```
sudo mount /dev/xvdb0 ebs-volume-2/
```

- Check whether the additional EBS volume is attached and properly mounted to the directory and after mounting, check the data created on Server 1 it should now be available on Server 2.

Command to check storage : lsblk or df -h

Command to data : cd ebs-volume-2

Command the data into the directory : cat my-volume

```

ubuntu@ip-172-31-82-164:~$
ubuntu@ip-172-31-82-164:~$ mkdir ebs-volume-2
ubuntu@ip-172-31-82-164:~$
ubuntu@ip-172-31-82-164:~$ ls
ebs-volume-2
ubuntu@ip-172-31-82-164:~$
ubuntu@ip-172-31-82-164:~$ sudo mount /dev/xvdb0
mount: /dev/xvdb0: can't find in /etc/fstab.
ubuntu@ip-172-31-82-164:~$
ubuntu@ip-172-31-82-164:~$
ubuntu@ip-172-31-82-164:~$ sudo mount /dev/xvdb0 ebs-volume-2/
ubuntu@ip-172-31-82-164:~$
ubuntu@ip-172-31-82-164:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        6.8G  1.8G  5.0G  26% /
tmpfs            479M   0  479M   0% /dev/shm
tmpfs            192M  880K  191M   1% /run
tmpfs            5.0M   0   5.0M   0% /run/lock
/dev/xvda16      881M   86M  734M  11% /boot
/dev/xvda15      105M   6.2M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
/dev/xvdb0       15G   28K   14G   1% /home/ubuntu/ebs-volume-2
ubuntu@ip-172-31-82-164:~$ ls
ebs-volume-2
ubuntu@ip-172-31-82-164:~$
ubuntu@ip-172-31-82-164:~$ cd ebs-volume-2/
ubuntu@ip-172-31-82-164:~/ebs-volume-2$ ls
lost+found  my-volume
ubuntu@ip-172-31-82-164:~/ebs-volume-2$ cat my-volume
Hello I'm Arun Kumar Akula
working as a DevOps Engineer.!
```

Now the volume is successfully mounted to Server 2. Next, we'll move the data to another Availability Zone. so far, all operations were within the same zone.

Move EBS Volume Data Across Availability Zones Using Snapshot

What is a Snapshot:

A **snapshot** is a backup of an EBS volume that you can use to create a new volume with the same data, even in a different Availability Zone.

- **Backup:** To save a copy of your EBS volume.
- **Recovery:** Restore data if something goes wrong.
- **Cross-AZ transfer:** Create volumes in a different Availability Zone.

Step 1: Create a Volume from Snapshot

- Go to AWS EC2 Console
- In the left menu, click **"Volumes"** under **Elastic Block Store**.
- Find and select the volume you want to snapshot
- Click **"Actions"** → **"Create Snapshot"**.

Volumes (1/3) Info

Saved filter sets: Choose filter set

Search

	Name	Volume ID	Type	Size	IOPS	Throughput
<input checked="" type="checkbox"/>		vol-08a4bd817bea74360	gp3	15 GiB	3000	125
<input type="checkbox"/>		vol-03974f2d97b35e079	gp3	8 GiB	3000	125
<input type="checkbox"/>		vol-023606280fc582ff9	gp3	8 GiB	3000	125

Volume ID: vol-08a4bd817bea74360

Actions menu:

- Modify volume
- Create snapshot**
- Create snapshot lifecycle policy
- Delete volume
- Attach volume
- Detach volume
- Force detach volume
- Manage auto-enabled I/O
- Manage tags
- Fault injection

In the popup:

- **Name** (optional): my-server1-backup
- **Description**: Snapshot from Server 1 EBS volume

Step 2 : Create Volume from Snapshot

- Go to the **AWS EC2 Console**
- In the left menu, click **"Snapshots"** under **Elastic Block Store**.
- Select the snapshot you created earlier.
- Click **"Actions"** → **"Create Volume"**.

Snapshots (1/1) Info

Owned by me

Search

	Name	Snapshot ID	Full snapshot size	Volume size	Description
<input checked="" type="checkbox"/>	snap	snap-0b188d01308d3ec81	318 MiB	15 GiB	snapshot

Actions menu:

- Create volume from snapshot**
- Create image from snapshot
- Copy snapshot
- Launch copy duration calculator
- Delete snapshot
- Manage tags
- Snapshot settings
- Archiving

Fill in the volume details:


- Select the volume type (general purpose SSD)
- **Size**: Must be equal to or greater than snapshot size (eg : 12)
- **Availability Zone**: Choose the **same AZ** as your target EC2 instance (e.g., ap-south-1b)
- Remining will be leave as the default
- Click on create volume

Create volume [Info](#)

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

Volume settings

Snapshot ID

 snap-01cafc977184b2470

Volume type [Info](#)

General Purpose SSD (gp3) ▼

Size (GiB) [Info](#)

12

Min: 1 GiB, Max: 16384 GiB.

IOPS [Info](#)

3000

Min: 3000 IOPS, Max: 16000 IOPS.

Throughput (MiB/s) [Info](#)

125

Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.

Availability Zone [Info](#)

us-east-1d ▼

Fast snapshot restore [Info](#)

Not enabled for selected snapshot

Volume initialization rate (MiB/s) - new, optional [Info](#)

Specify the rate at which the snapshot blocks are to be downloaded from Amazon S3 to the volume. [Additional costs apply](#)

Enter a value

Min: 100 MiB/s, Max: 300 MiB/s.

Encryption [Info](#)

Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.

Step 3: After Volume is Created

- Go to Volumes in the sidebar.
- Select the new volume → Click “Actions → Attach Volume”.



Volumes (1/5) [Info](#)

Saved filter sets


Choose filter set ▼

Search

<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS	Throughput	
<input type="checkbox"/>		vol-08a4bd817bea74360	gp3	15 GiB	3000	125	
<input checked="" type="checkbox"/>	snap-vol	vol-0326668c222b5c828	gp3	15 GiB	3000	125	
<input type="checkbox"/>		vol-03974f2d97b35e079	gp3	8 GiB	3000	125	
<input type="checkbox"/>		vol-004cf54726167de99	gp3	8 GiB	3000	125	
<input type="checkbox"/>		vol-023606280fc582ff9	gp3	8 GiB	3000	125	

Last updated 1 minute ago  **Actions**  **Create volume**

- Modify volume
- Create snapshot
- Create snapshot lifecycle policy
- Delete volume
- Attach volume**
- Detach volume
- Force detach volume
- Manage auto-enabled I/O
- Manage tags
- Fault injection

< 1 > 

16:54 GMT+5:...
17:43 GMT+5:...
17:05 GMT+5:...
17:38 GMT+5:...
16:54 GMT+5:...

Volume ID: vol-0326668c222b5c828 (snap-vol)

- Select your EC2 instance (e.g., Server 3), and use a device name like /dev/xvdbd
- Click on Attach volume.

EC2 > Volumes > vol-065d28c1853ac2403 > Attach volume

Attach volume [Info](#)

Attach a volume to an instance to use it as you would a regular physical hard disk drive.

Basic details

Volume ID

 vol-065d28c1853ac2403

Availability Zone

us-east-1a

Instance [Info](#)


i-0e7e67c2bebc71dea
(snapshot) (running)

Only instances in the same Availability Zone as the selected volume are displayed.

Device name [Info](#)

/dev/xvdbb

Recommended device names for Linux: /dev/sda1 for root volume, /dev/sd[f-p] for data volumes.

 Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

[Cancel](#)

[Attach volume](#)

Step 4: Connect your server-3 and mount

Commands to directory: sudo mkdir snapshot

```
ubuntu@ip-172-31-28-42:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINTS
loop0        7:0      0   27.2M 1 loop /snap/amazon-ssm-agent/11320
loop1        7:1      0   73.9M 1 loop /snap/core22/1981
loop2        7:2      0   50.9M 1 loop /snap/snapd/24505
xvda        202:0     0    8G  0 disk
├─xvda1      202:1     0    7G  0 part /
├─xvda14     202:14    0    4M  0 part
├─xvda15     202:15    0   106M 0 part /boot/efi
└─xvda16     259:0     0   913M 0 part /boot
xvdbb        202:13568 0   15G  0 disk
ubuntu@ip-172-31-28-42:~$
ubuntu@ip-172-31-28-42:~$ mkdir snapshot
```

Command to mount : sudo mount /dev/xvdbd snapshot/

```
ubuntu@ip-172-31-28-42:~$ sudo mount /dev/xvdbb snapshot/
ubuntu@ip-172-31-28-42:~$
ubuntu@ip-172-31-28-42:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        6.8G  1.8G  5.0G  26% /
tmpfs            479M   0  479M   0% /dev/shm
tmpfs            192M  872K  191M   1% /run
tmpfs            5.0M   0   5.0M   0% /run/lock
/dev/xvda16      881M   86M  734M  11% /boot
/dev/xvda15      105M   6.2M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
/dev/xvdbb       15G   28K  14G   1% /home/ubuntu/snapshot
ubuntu@ip-172-31-28-42:~$
ubuntu@ip-172-31-28-42:~$ cd snapshot/
ubuntu@ip-172-31-28-42:~/snapshot$ ls
lost+found  my-volume
ubuntu@ip-172-31-28-42:~/snapshot$ cat my-volume
Hello I'm Arun Kumar Akula
working as a DevOps Engineer.!
ubuntu@ip-172-31-28-42:~/snapshot$ |
```