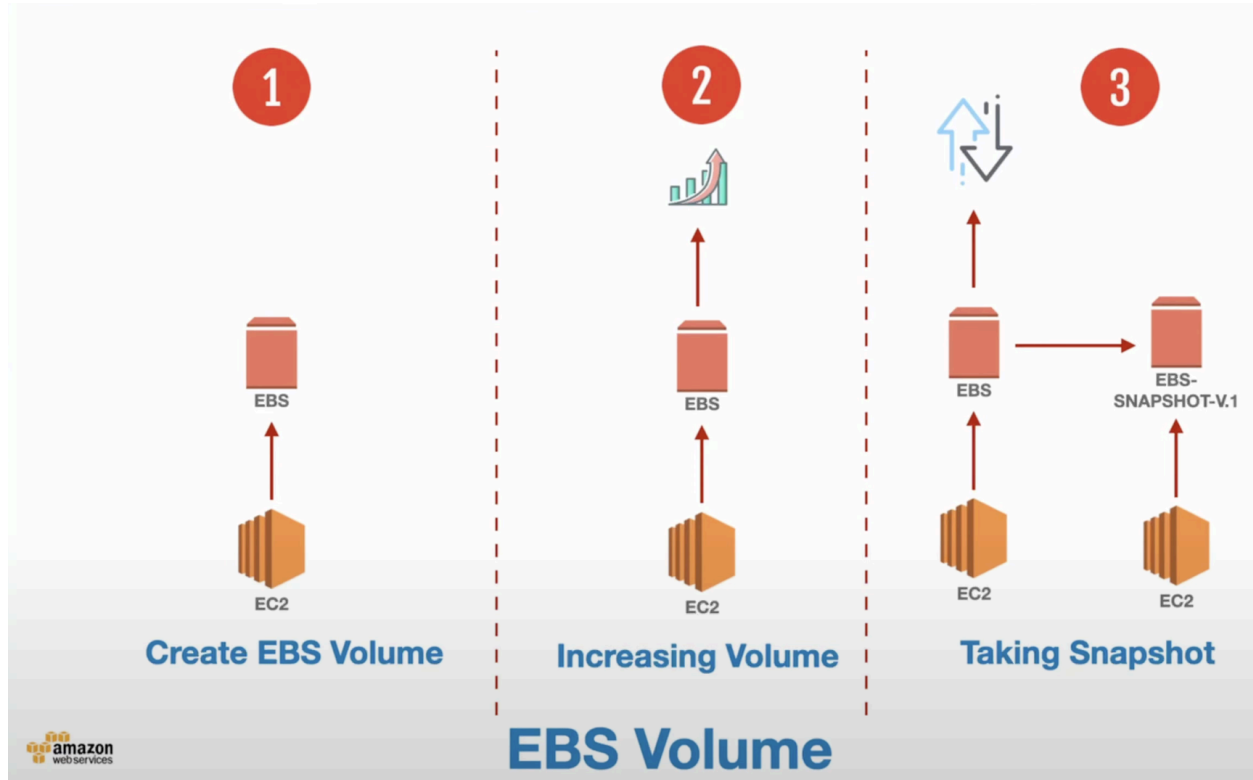


Amazon EBS (Elastic Block Store)



What is EBS?

- **Elastic Block Store (EBS)** is a storage service provided by AWS that allows you to create virtual hard drives and attach them to **EC2 instances**.
- EC2 comes with a default root volume (usually **8 GB gp3**).
- EBS volumes are **persistent**, **scalable**, and can be **backed up using snapshots**.

Types of EBS Volumes

1. **General Purpose Volumes:**
 - **gp3**, **gp2** – suitable for everyday workloads with moderate I/O.
 - **gp3** is recommended in most cases.
2. **Provisioned IOPS Volumes:**

- **io2, io1** – used for high-performance and I/O-intensive applications like large databases.

Note: EC2 and EBS must be in the same Availability Zone (AZ) to attach the volume.

Part 1: Creating and Mounting an EBS Volume

Step-by-step:

1. Launch an EC2 instance (Amazon Linux or Ubuntu)

The screenshot shows the 'Launch an instance' page in the AWS Management Console. The breadcrumb navigation is 'EC2 > Instances > Launch an instance'. The page title is 'Launch an instance' with an 'Info' link. A sub-header states: 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.'

The main form is divided into sections:

- Name and tags:** The 'Name' field contains 'ec2-1'. There is an 'Add additional tags' link.
- Application and OS Images (Amazon Machine Image):** A search bar contains the text 'Search our full catalog including 1000s of application and OS images'. Below the search bar are tabs for 'Recents' and 'Quick Start'. Under 'Quick Start', there are tiles for various operating systems: Amazon Linux, macOS, Ubuntu (selected), Windows, Red Hat, SUSE Linux, and Debian. A 'Browse more AMIs' link is also present.

On the right side, there is a 'Summary' panel with the following details:

- Number of instances:** 1
- Software Image (AMI):** Canonical, Ubuntu, 24.04, amd64... (with a 'read more' link and AMI ID ami-084568db4383264d4)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

At the bottom right, there is a 'Free tier' notification: 'Free tier: In your first year of opening an AWS account, you get 750 hours per month of...'. Below this are 'Cancel' and 'Launch instance' buttons, and a 'Preview code' link.

2. Create an EBS volume from AWS Console

The screenshot shows the 'Create volume' page in the AWS Management Console. The breadcrumb navigation is 'EC2 > Volumes > Create volume'. The page title is 'Create volume' with an 'Info' link. A sub-header states: 'Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.'

The main form is titled 'Volume settings' and contains the following fields:

- Volume type:** A dropdown menu showing 'General Purpose SSD (gp3)'.
- Size (GiB):** A text input field containing '100'. Below it, the range 'Min: 1 GiB, Max: 16384 GiB' is displayed.
- IOPS:** A text input field containing '3000'. Below it, the range 'Min: 3000 IOPS, Max: 16000 IOPS' is displayed.
- Throughput (MiB/s):** A text input field containing '125'. Below it, the range 'Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s' is displayed.
- Availability Zone:** A dropdown menu showing 'us-east-1a'.

3. Attach the EBS volume to the EC2 instance (ensure both are in the same AZ)

Availability Zone | [Info](#)

us-east-1a ▼

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-01d7baa4b3b633e6c (ebs1)

Availability Zone
us-east-1a

Instance | [Info](#)
i-0d41ccda4964cf80b (ec2-1) (running) ⓘ

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

Device name | [Info](#)
/dev/sdk ▼

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.

4. Connect to EC2 using SSH

```
[→ desktop ssh -i "test-key.pem" ubuntu@ec2-52-70-1-56.compute-1.amazonaws.com ]
The authenticity of host 'ec2-52-70-1-56.compute-1.amazonaws.com (52.70.1.56)' c
an't be established.
ED25519 key fingerprint is SHA256:uPAzKpag0ljiSIqIQrCGEX8TnOqweUPjXiGLxe2FrC4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-70-1-56.compute-1.amazonaws.com' (ED25519) to
the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)
```

Inside the EC2 terminal:

sudo su	: switch to root user
lsblk	: to show all the block devices
(volumes)	
fdisk -l	: to show all disk partitions
file -s /dev/xvdf	: to check if the EBS volume has a
file system	
mkfs -t xfs /dev/xvdf	: to create an XFS file system on the
volume (only if file system is 'data')	

file -s /dev/xvdf : to verify that file system is now created

mkdir /myebsvol : to create a mount directory for the EBS volume

mount /dev/xvdf /myebsvol : to mount the EBS volume to the created directory

```
root@ip-172-31-46-114:/home/ubuntu# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0  26.3M  1 loop /snap/amazon-ssm-agent/9881
loop1       7:1      0  73.9M  1 loop /snap/core22/1748
loop2       7:2      0  44.4M  1 loop /snap/snapd/23545
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
xvdk        202:160   0  100G  0 disk
root@ip-172-31-46-114:/home/ubuntu# fdisk -l
Disk /dev/loop0: 26.32 MiB, 27602944 bytes, 53912 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 73.89 MiB, 77479936 bytes, 151328 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 44.44 MiB, 46596096 bytes, 91008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E3478E01-32E3-4FC2-8E79-1BCCDE89C2D7

Device        Start      End  Sectors  Size Type
/dev/xvda1    2099200 16777182 14677983   7G Linux filesystem
/dev/xvda14      2048    10239    8192    4M BIOS boot
/dev/xvda15     10240   227327   217088  106M EFI System
/dev/xvda16    227328   2097152  1869825  913M Linux extended boot

Partition table entries are not in disk order.

Disk /dev/xvdk: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@ip-172-31-46-114:/home/ubuntu# file -s /dev/xvdk
/dev/xvdk: data
root@ip-172-31-46-114:/home/ubuntu# mkfs -t xfs /dev/xvdk
meta-data=/dev/xvdk             isize=512    agcount=4, agsize=6553600 blks
      =                       sectsz=512    attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=1, rmapbt=1
      =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
data      =                       bsize=4096   blocks=26214400, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0, ftype=1
log       =internal log       bsize=4096   blocks=16384, version=2
      =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                extsz=4096   blocks=0, rtextents=0
```

At this point, the EBS volume is ready to use. You can read/write files in `/myebsvol`.

Part 2: Creating Snapshot and Using It on Another EC2

Step-by-step:

1. Create files in the mounted EBS volume

```
cd /myebsvol           : navigate to mounted volume
touch file1.txt file2.txt : create test files
```


```
root@ip-172-31-46-114:/home/ubuntu# cd myebsvol
root@ip-172-31-46-114:/home/ubuntu/myebsvol# ls
root@ip-172-31-46-114:/home/ubuntu/myebsvol# touch file1.txt
root@ip-172-31-46-114:/home/ubuntu/myebsvol# touch file2.txt
root@ip-172-31-46-114:/home/ubuntu/myebsvol# ls
file1.txt  file2.txt
root@ip-172-31-46-114:/home/ubuntu/myebsvol#
```

2. Go to AWS Console → Volumes → Select volume → Create Snapshot

- This saves a backup of the volume at that point in time

Create snapshot [Info](#)
Create a point-in-time snapshot to back up the data on an Amazon EBS volume to Amazon S3.

Source volume

Volume ID
 [vol-01d7baa4b3b633e6c \(ebs1\)](#)

Availability Zone
us-east-1a

Snapshot details

Description
Add a description for your snapshot

255 characters maximum.

Encryption [Info](#)
Not encrypted


3. Launch a **second EC2 instance** in the **same AZ**

<input type="checkbox"/>	Name ↗	Instance ID	Instance state ↕	Instance type ↕	Status check	Alarm status	Availability Zone ↕
<input type="checkbox"/>	ec2-2	i-049b53acc385c2f18	Running 🔍 🔍	t2.micro	⌚ Initializing	View alarms +	us-east-1a
<input type="checkbox"/>	ec2-1	i-0d41ccda4964cf80b	Running 🔍 🔍	t2.micro	✅ 2/2 checks passed	View alarms +	us-east-1a

4. Create a new volume from the snapshot

Create volume [Info](#)
Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

Volume settings

Snapshot ID
 `snap-022e604984d7354f5` (snapshot-ebs1)

Volume type [Info](#)
General Purpose SSD (gp3) ▼

Size (GiB) [Info](#)

Min: 1 GiB, Max: 16384 GiB.

IOPS [Info](#)

Min: 3000 IOPS, Max: 16000 IOPS.


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

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


5. Attach the volume to the second EC2 instance

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-0312618b4d1fb264c` (ebs2)

Availability Zone
us-east-1a

Instance [Info](#)
 (ec2-2) (running) 
Only instances in the same Availability Zone as the selected volume are displayed.

Device name [Info](#)
 ▼
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`.

Inside the second EC2 instance:

```
sudo su           : switch to root user
lsblk             : to view all attached volumes
fdisk -l          : to check disk partitions
file -s /dev/xvdf : to check file system on the new
volume (should already exist from snapshot)
```


mkdir /myebsvol2 : create a mount point for the new volume
mount /dev/xvdf /myebsvol2 : mount the snapshot-based volume
ls /myebsvol2 : list files in the mounted volume;
should show file1.txt and file2.txt

```
ubuntu@ip-172-31-34-125:~$ sudo su
root@ip-172-31-34-125:/home/ubuntu# fdisk -l
Disk /dev/loop0: 26.32 MiB, 27602944 bytes, 53912 sectors
[Units: sectors of 1 * 512 = 512 bytes
[Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 73.89 MiB, 77479936 bytes, 151328 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 44.44 MiB, 46596096 bytes, 91008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E3478E01-32E3-4FC2-8E79-1BCCDE89C2D7

Device          Start      End  Sectors  Size Type
/dev/xvda1    2099200 16777182 14677983   7G Linux filesystem
/dev/xvda14      2048    10239    8192    4M BIOS boot
/dev/xvda15     10240   227327   217088  106M EFI System
/dev/xvda16    227328  2097152  1869825  913M Linux extended boot

Partition table entries are not in disk order.

Disk /dev/xvdk: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@ip-172-31-34-125:/home/ubuntu# file -s /dev/xvdk
/dev/xvdk: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
root@ip-172-31-34-125:/home/ubuntu# mkdir myebsvol2
root@ip-172-31-34-125:/home/ubuntu# mount /dev/xvdk myebsvol2
root@ip-172-31-34-125:/home/ubuntu# cd myebsvol2
root@ip-172-31-34-125:/home/ubuntu/myebsvol2# ls
file1.txt  file2.txt
root@ip-172-31-34-125:/home/ubuntu/myebsvol2#
```

Note :Only the data present **at the time the snapshot was taken** will be present in this volume. Any files added **after the snapshot (e.g., file3.txt, file4.txt)** will **not appear** in the volume created from the snapshot.

Conclusion

In this documentation, we explored the fundamentals of Amazon EBS, including how to create, attach, and mount EBS volumes to EC2 instances. We also covered creating snapshots for backup and demonstrated how to restore data by creating new volumes from those snapshots.

Understanding EBS is essential for managing persistent storage in AWS. Through hands-on practice with volume formatting, file system creation, and snapshot recovery, we've built a strong foundation in managing storage for scalable and reliable cloud applications.