

## **PROJECT**

**Project Title :** Attaching and configuring Amazon EBS Volume on a Linux Instance

**Date :** 22/10/2025

**Name of the person :** Gadi Thirupathi

**Name of the project :** Attaching and configuring Amazon EBS Volume on a Linux Instance.

**Objective of the project :**

- Create an EC2 Linux instance
- Create an EBS volume
- Attach the EBS volume to the EC2 instance
- Format and mount the volume
- Test the setup

**Steps done in the project :**

**step 1 :**

- Go to easy Ec2 dashboard
- Launch a new amazon linux instance
- select appropriate instance type {eg:t3.micro}
- configure key pair for SSH access
- Allow SSH (port 22) in security group

**step 2 :**

- Go to Elastic Block Store > Volumes.
- Click Create Volume.
- Select same Availability Zone as your EC2 instance (e.g., us-east-1a).
- Choose volume type (e.g., gp2), and size (e.g., 30 GiB).

- Click Create Volume.

### step 3 :

- After creating, select the volume → Actions > Attach Volume.
- Select the EC2 instance.
- Set device name (e.g., /dev/xvdf).

### Output screenshots :

==> ec2 linux machine

The screenshot displays the AWS Management Console for the EC2 service. On the left, the navigation pane shows the 'Instances' section. The main content area shows a list of instances with one instance, 'Linux-machine01', selected. The instance is in the 'Running' state and is a 't3.micro' type. Below the list, the 'Details' tab for the instance 'i-01ad3886e52d42946 (Linux-machine01)' is shown. The instance summary includes the Instance ID, Public IPv4 address (13.222.1.184), and Private IPv4 addresses (10.1.2.32).

==> Ebs volume

The screenshot displays the AWS Management Console for the Elastic Block Store (EBS) service. On the left, the navigation pane shows the 'Volumes' section. The main content area shows a list of volumes with one volume, 'Linux', selected. The volume is in the 'Available' state and is a 'gp2' type with a size of 30 GiB and 100 IOPS. Below the list, the 'Details' tab for the volume 'vol-0a8874a20bf9a7af9 (Linux)' is shown. The volume summary includes the Volume ID, Size (30 GiB), Type (gp2), and Status check (Okay).

==> Attach Ebs volume to Ec2

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

### Basic details

Volume ID

vol-0a8874a20bf9a7af9 (Linux)

Availability Zone

use1-az4 (us-east-1b)

Instance | Info

i-01ad3886e52d42946  
(Linux-machine01) (running)


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

Device name | Info

```
/dev/sdb
```

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

```
Warning: Permanently added 'ec2-13-222-1-184.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
```



```
#_
#####
Amazon Linux 2023

#####
####|
#|
#|
V_ _ _ _ _>
https://aws.amazon.com/linux/amazon-linux-2023

      _ _ _ _ _
     /   \   \
    /     \   \
   /m/'     \
```

```
[ec2-user@ip-10-1-2-32 ~]$
```

```
[root@ip-10-1-2-32:~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1      259:0    0   8G  0 disk
├─nvme0n1p1  259:1    0   8G  0 part /
├─nvme0n1p127 259:2    0  1M  0 part
└─nvme0n1p128 259:3    0 10M  0 part /boot/efi
nvme1n1      259:4    0  30G  0 disk
```

The volume is attached to linux machine

**==> Formate the Ebs volume**

```
[root@ip-10-1-2-32 ~]# sudo mkfs -t ext4 /dev/nvme1n1
mke2fs 1.46.5 (30-Dec-2021)
/dev/nvme1n1 contains a ext4 file system
    last mounted on Wed Oct 22 11:03:07 2025
Proceed anyway? (y,N) y
Creating filesystem with 7864320 4k blocks and 1966080 inodes
Filesystem UUID: e8153110-1619-458c-9044-62bc0a38dff6
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-10-1-2-32 ~]#
```

## ==> Create Mount Point and Mount Volume

```
[root@ip-10-1-2-32 ~]# sudo mkdir /data
[root@ip-10-1-2-32 ~]# sudo mount /dev/nvme1n1 /d
data/ dev/
[root@ip-10-1-2-32 ~]# sudo mount /dev/nvme1n1 /data/
[root@ip-10-1-2-32 ~]#
```

## ==> verify the mount

```
[root@ip-10-1-2-32 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M   0% /dev
tmpfs           453M   0  453M   0% /dev/shm
tmpfs           181M  444K  181M   1% /run
/dev/nvme0n1p1  8.0G  1.6G  6.4G  20% /
tmpfs           453M   0  453M   0% /tmp
/dev/nvme0n1p128 10M  1.3M  8.7M  13% /boot/efi
tmpfs           91M   0   91M   0% /run/user/1000
/dev/nvme1n1    30G   24K  28G   1% /data
[root@ip-10-1-2-32 ~]#
```

You can see /dev/nvme1n1 mounted on /data

## Tools and services used :

- AWSVPC
- AWSEC2
- CAMMAND PROMPT
- 

**Conclusion :** In this project,i suuccessfully demonstrated how to extend the storage capabilities of an amazon ec2 instance by integrating an amazon elastic block store (EBS) volume. The proccess included creating and attaching a new EBS volume,farmating it with a suitable file system (ext4),and mounting it to a directory on linux instance.