Managing Amazon EC2 & EBS Volume With Snapshort Backup And Restore Operations.

.

Kartik Kakde

Amazon EBS

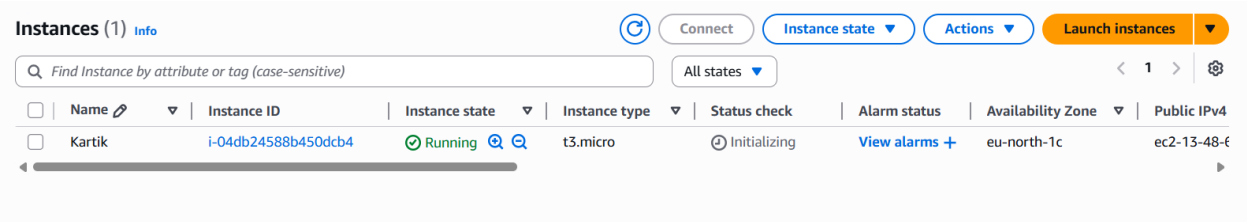
An Amazon Elastic Block Store (EBS) volume is a durable, block-level storage device that you can attach to an EC2 instance. It behaves like a physical hard drive, persists independently of the instance, and can be resized or reconfigured without downtime.

Amazon Snapshot

An AWS snapshot is a point-in-time, incremental backup of an Amazon EBS (Elastic Block Store) volume, stored in Amazon S3. It helps with backup, disaster recovery, migration, and cloning workloads.

Step 1 : Create EC2 Instance

- Sign in to AWS Management Console.
- Launch Instance.
- Configure EC2 instance.
- Select AMI.
- Choose Instance Type.
- Key Pair (Default).
- Network Settings.
- Security Group (Choose Select existing security group→ pick default security group.)
- Launch the Instance.

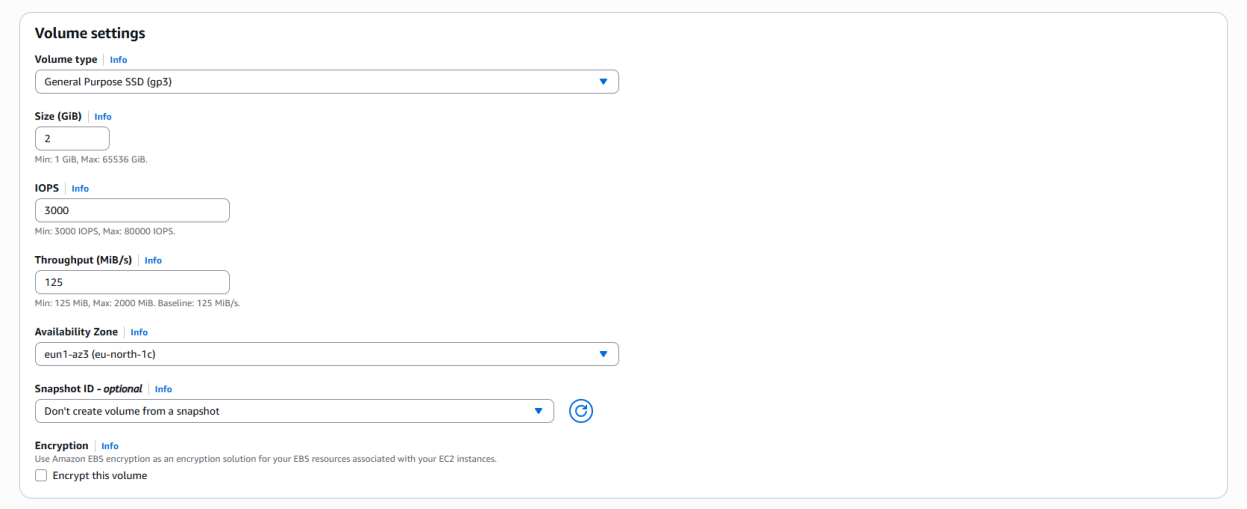


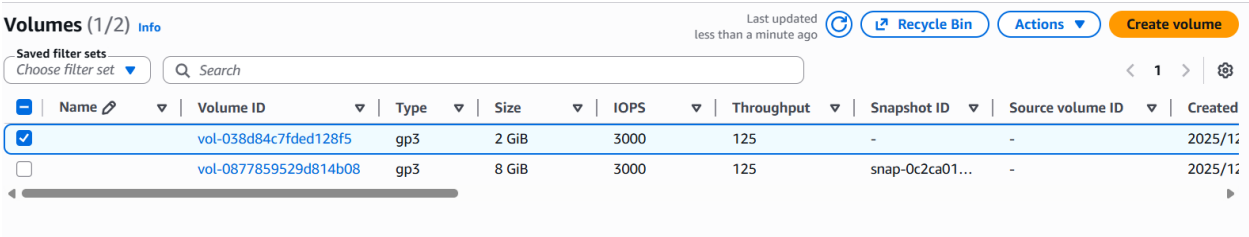
Step 2 : Create a new EBS volume and make sure it is in the same Availability Zone as the EC2 instance.

Go to EC2 Dashboard → Elastic Block Store → Volumes.

- Click Create Volume.
- Enter size (e.g., 2 GiB).
- Choose volume type (gp3 recommended).

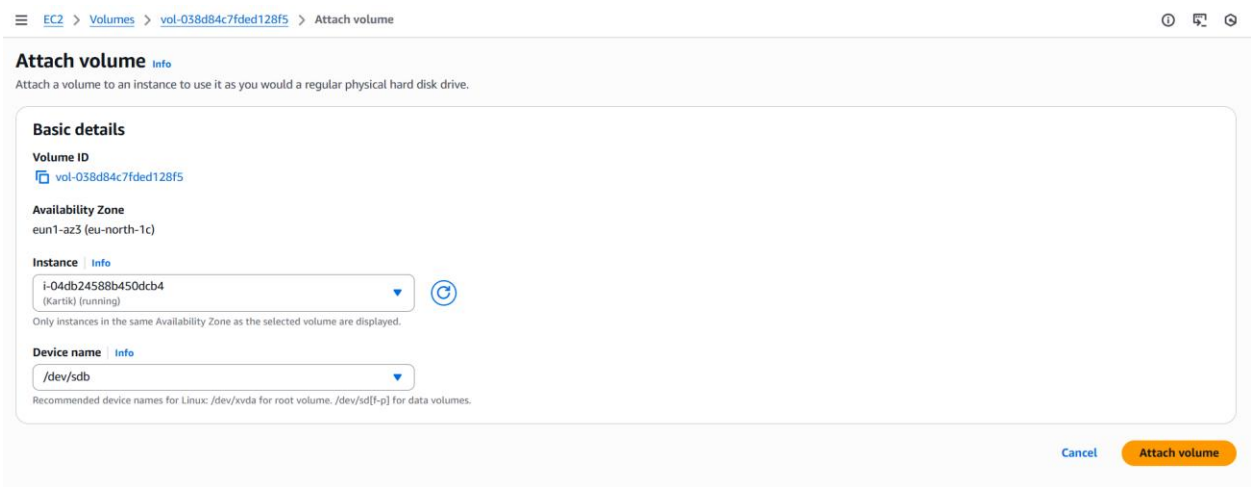
- Very important → Select Availability Zone matching your EC2 instance (e.g., eu-north-1c).
- Click Create Volume.



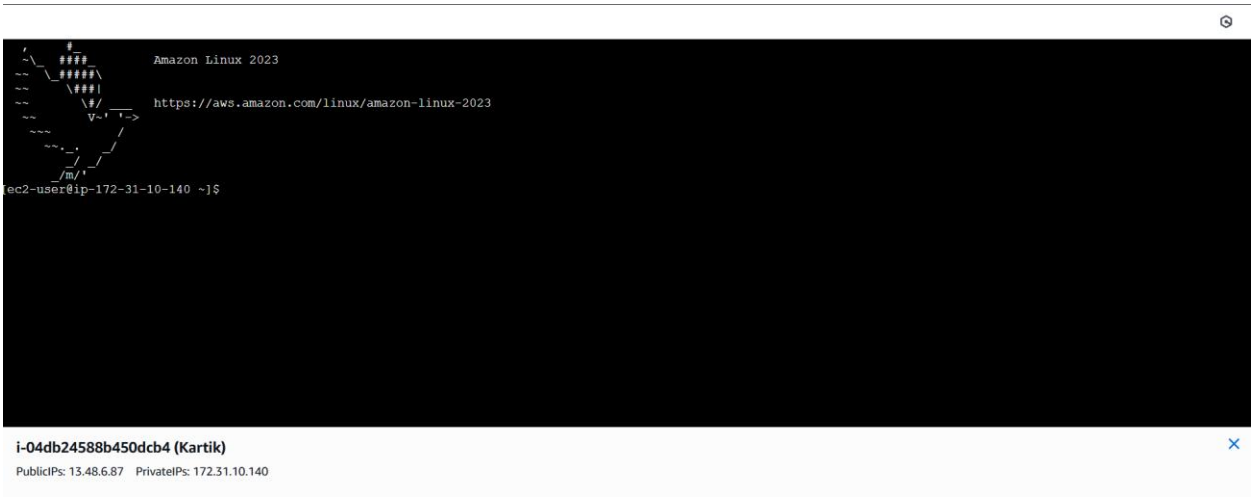


Step 3 : Attach the EBS volume to the EC2 instance.

- Go to EC2 → Volumes.
- Select the newly created volume.
- Click Actions → Attach Volume.
- Choose your instance from the dropdown.
- Device name will auto-fill (e.g., /dev/sdb).
- Click Attach Volume.



Step 4 : Connect to the Instance.



Check the Attached Disk Using (lsblk) Command.

```
[ec2-user@ip-172-31-10-140 ~]$ 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   2G  0 disk
[ec2-user@ip-172-31-10-140 ~]$
```

Kartik Kakde

Create a Partition :

1.Start parted on the Disk.

```
[ec2-user@ip-172-31-10-140 ~]$ sudo parted /dev/nvme1n1
GNU Parted 3.4
Using /dev/nvme1n1
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

You will enter the parted shell.

2. Set Disk Label (msdos).

```
(parted) p
Error: /dev/nvme1n1: unrecognised disk label
Model: Amazon Elastic Block Store (nvme)
Disk /dev/nvme1n1: 2147MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
(parted) mklabel msdos
(parted) p
Model: Amazon Elastic Block Store (nvme)
Disk /dev/nvme1n1: 2147MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End  Size  Type  File system  Flags

(parted)
```

3. Create a New Partition.

```
(parted) mkpart
Partition type?  primary/extended? p
File system type?  [ext2]?
Start? 1M
End? 2000M
(parted) p
Model: Amazon Elastic Block Store (nvme)
Disk /dev/nvme1n1: 2147MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  2000MB  1999MB  primary  ext2         lba
```

4.Format the New Partition.

Example ext4 format :

```
[ec2-user@ip-172-31-10-140 ~]$ sudo mkfs.ext4 /dev/nvme1n1p1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 487936 4k blocks and 122160 inodes
Filesystem UUID: 82291e17-e2d5-4c18-b08a-d9ae1a84e54a
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-172-31-10-140 ~]$
```

5.Create a Mount Point & Mount the Partition.

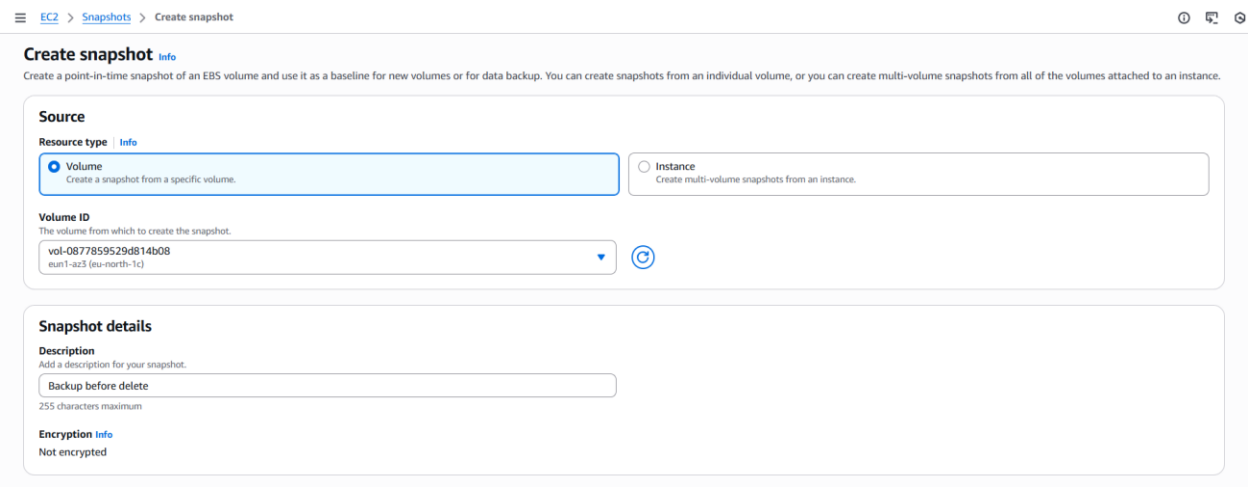
```
[ec2-user@ip-172-31-10-140 ~]$ sudo mkdir /sample
[ec2-user@ip-172-31-10-140 ~]$ sudo mount /dev/nvme1n1p1 /sample
[ec2-user@ip-172-31-10-140 ~]$ df -h
Filesystem        Size  Used Avail Use% Mounted on
devtmpfs          4.0M     0  4.0M   0% /dev
tmpfs             459M     0  459M   0% /dev/shm
tmpfs             184M  436K  183M   1% /run
/dev/nvme0n1p1    8.0G  1.6G  6.4G  20% /
tmpfs             459M     0  459M   0% /tmp
/dev/nvme0n1p128   10M  1.3M  8.7M  13% /boot/efi
tmpfs              92M     0   92M   0% /run/user/1000
/dev/nvme1n1p1    1.8G   24K  1.7G   1% /sample
[ec2-user@ip-172-31-10-140 ~]$
```

6. Create a few empty files in the directory where the partition is mounted.

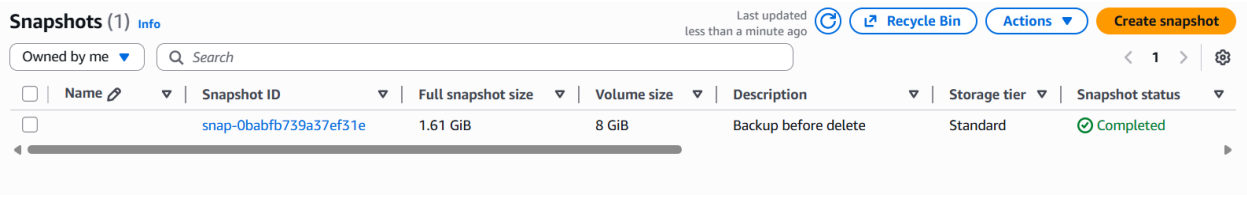
```
[ec2-user@ip-172-31-10-140 ~]$ cd /sample
[ec2-user@ip-172-31-10-140 sample]$ sudo touch file{1..10}.txt
[ec2-user@ip-172-31-10-140 sample]$ ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt  file8.txt  file9.txt  lost+found
[ec2-user@ip-172-31-10-140 sample]$
```

Step 5 : Create Snapshot of the EBS Volume.

- Go to EC2 → Volumes.
- Select your volume.
- Click Actions → Create Snapshot.
- Enter description (e.g., “Backup before delete”).
- Click Create Snapshot.



- Snapshot Created.

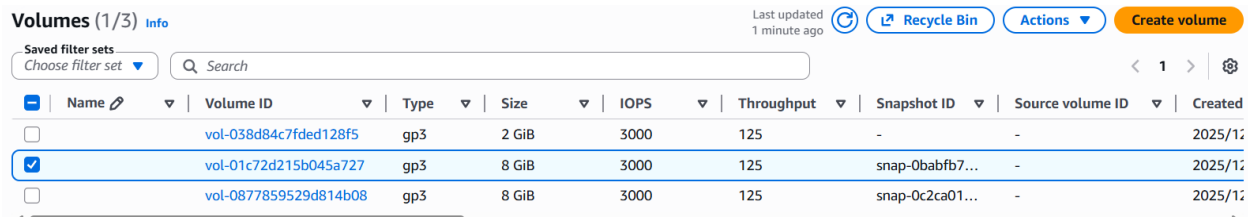


Step 6 : Delete the Files from the Directory.

```
[ec2-user@ip-172-31-10-140 sample]$ ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt  file8.txt  file9.txt  lost+found
[ec2-user@ip-172-31-10-140 sample]$ sudo rm -rvf file{1..10}.txt
removed 'file1.txt'
removed 'file2.txt'
removed 'file3.txt'
removed 'file4.txt'
removed 'file5.txt'
removed 'file6.txt'
removed 'file7.txt'
removed 'file8.txt'
removed 'file9.txt'
removed 'file10.txt'
[ec2-user@ip-172-31-10-140 sample]$ ls
lost+found
[ec2-user@ip-172-31-10-140 sample]$
```

Step 7 : Create a New Volume from Snapshot.

- Go to EC2 → Snapshots.
- Select your snapshot.
- Click Actions → Create Volume.
- Choose same Availability Zone as EC2.
- Click Create Volume.



Step 8 : Attach the Snapshot-Based Volume.

- Go to EC2 → Volumes.
- Select the new volume created from snapshot.
- Click Actions → Attach Volume.
- Select the same EC2 instance.
- Click Attach.

Step 9 : Connect to the Instance.

- Confirm Disk Visibility.

```
[ec2-user@ip-172-31-10-140 ~]$ 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    2G  0 disk
└─nvme1n1p1   259:6    0  1.9G  0 part /sample
nvme2n1       259:5    0    8G  0 disk
├─nvme2n1p1   259:7    0    8G  0 part
├─nvme2n1p127 259:8    0    1M  0 part
└─nvme2n1p128 259:9    0   10M  0 part
[ec2-user@ip-172-31-10-140 ~]$
```

- Mount the Restored Volume to Another Directory And Verify Restored Files.

```
[ec2-user@ip-172-31-10-140 ~]$ cd /test
[ec2-user@ip-172-31-10-140 test]$ ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt  file8.txt  file9.txt
[ec2-user@ip-172-31-10-140 test]$
```

THANK YOU

Kartik Kakde