

# Create EBS and Attach Volume to EC2 Instance

## What is Elastic Block Storage (EBS)

Amazon Elastic Block Store (Amazon EBS) provides scalable, high-performance block storage resources that can be used with Amazon Elastic Compute Cloud (Amazon EC2) Instances.

## What is EBS Volume?

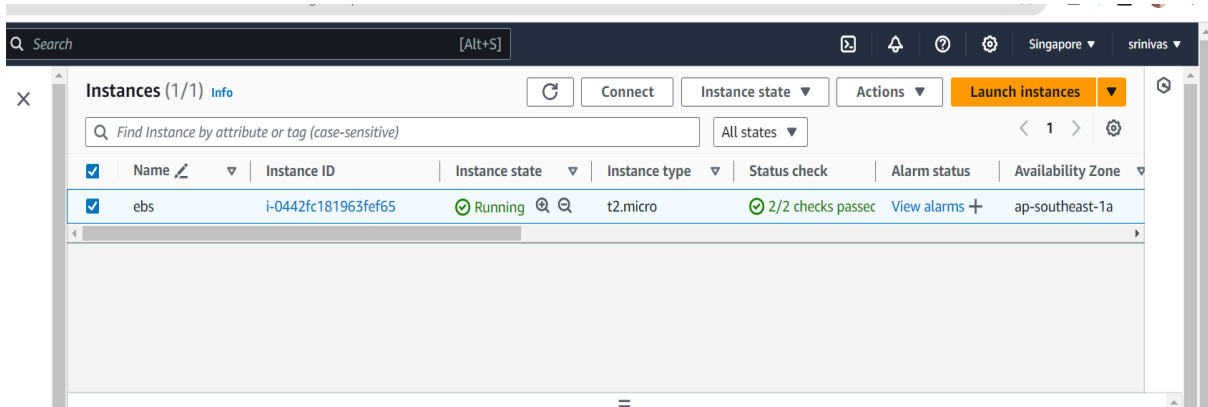
These are storage volumes that you attach to Amazon EC2 instances. After you attach a volume to an instance, you can use it in the same way you would use a local hard drive attached to a computer, for example to store files or to install applications.

## What is EC2 Instance?

Amazon Web Service EC2 (Amazon Elastic Compute Cloud), one of Amazon Web Services' most well-known services, offers businesses the ability to run applications on the public cloud. An EC2 instance is simply a virtual server in Amazon Web Services terminology

## Create EC2 Instance:

1. Create EC2 Instances.
2. Go to instance -- Launch Instance – create key pair – Select Security Group – Launch Instance.
3. Some EC2 Snapshots are attached below.
4. After creating EC2 Instances & Copy the SSH.
5. Paste it in the Git bash and Connect.



## EBS Volumes:

1. Go to the Volumes and Modify the existing Volume.
2. Click on Actions – Modify volume – size (16) – Modify.
3. Create a Volume – size (10) – same AZ which is taken by Instance.
4. Select volume - Actions - Attach Volume - Running Instance (EBS) - /dev/sdf.  
- Create Volume.

### Attach volume

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

#### Basic details

Volume ID

vol-040d2407d292d41f5

Availability Zone

ap-southeast-1a

Instance [Info](#)

i-0442fc181963fef65



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

Device name [Info](#)

Select a device name

Recommended device names for Linux: /dev/xvda 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**.

5. Go to git bash, connect the server and use this command `df -h`.
6. Check there is a file system “ `file -s /dev/xvdf` ”.
7. Create file system “ `mkfs -t xfs /dev/xvdf` ”.
8. Create a directory `mkdir -p apps/volume` .
9. `mount /dev/xvdf apps/volume`.
10. `df -h`.

```
15 root      20  0      0      0      0 I    0.0    0.0
16 root      rt   0      0      0      0 S    0.0    0.0
[ec2-user@ip-172-31-28-141 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0    4.0M   0% /dev
tmpfs           475M   0    475M   0% /dev/shm
tmpfs           190M 436K   190M   1% /run
/dev/xvda1      16G  1.6G   15G  10% /
tmpfs           475M   0    475M   0% /tmp
/dev/xvda128    10M  1.3M   8.7M  13% /boot/efi
tmpfs           95M   0     95M   0% /run/user/1000
[ec2-user@ip-172-31-28-141 ~]$ |
```

```

[ec2-user@ip-172-31-28-141 ~]$ sudo -i
[root@ip-172-31-28-141 ~]# mkfs -t xfs /dev/xvdf
meta-data=/dev/xvdf            isize=512    agcount=4, agsize=655360 blks
        =                       sectsz=512    attr=2, projid32bit=1
        =                       crc=1          finobt=1, sparse=1, rmapbt=0
        =                       reflink=1      bigtime=1 inobtcount=1
data      =                       bsize=4096   blocks=2621440, 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
[root@ip-172-31-28-141 ~]# file -s /dev/xvdf
/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
[root@ip-172-31-28-141 ~]#

```

```

[root@ip-172-31-28-141 ~]# df -h

```

Filesystem	Size	Used	Avail	Use%
Mounted on				
devtmpfs	4.0M	0	4.0M	0%
/dev				
tmpfs	475M	0	475M	0%
/dev/shm				
tmpfs	190M	440K	190M	1%
/run				
/dev/xvda1	16G	1.6G	15G	10%
/				
tmpfs	475M	0	475M	0%
/tmp				
/dev/xvda128	10M	1.3M	8.7M	13%
/boot/efi				
tmpfs	95M	0	95M	0%
/run/user/1000				
/dev/xvdf	10G	104M	9.9G	2%
/root/srinu/data				

```

[root@ip-172-31-28-141 ~]#

```