

## AMAZON WEB SERVICES

### Regions in AWS

#### 1. North America:

- US East (Northern Virginia) - us-east-1
- US East (Ohio) - us-east-2
- US West (Northern California) - us-west-1
- US West (Oregon) - us-west-2
- Canada (Central) - ca-central-1

#### 2. South America:

- South America (Sao Paulo) - sa-east-1

#### 3. Europe:

- Europe (Ireland) - eu-west-1
- Europe (London) - eu-west-2
- Europe (Frankfurt) - eu-central-1
- Europe (Paris) - eu-west-3
- Europe (Stockholm) - eu-north-1

#### 4. Asia Pacific:

- Asia Pacific (Mumbai) - ap-south-1
- Asia Pacific (Singapore) - ap-southeast-1
- Asia Pacific (Sydney) - ap-southeast-2
- Asia Pacific (Tokyo) - ap-northeast-1
- Asia Pacific (Seoul) - ap-northeast-2
- Asia Pacific (Osaka-Local) - ap-northeast-3

5. Middle East and Africa:

- Middle East (Bahrain) - me-south-1
- Africa (Cape Town) - af-south-1

6. China:

- China (Beijing) - cn-north-1
- China (Ningxia) - cn-northwest-1



These codes are often used when working with AWS services and resources programmatically or through the AWS Command Line Interface (CLI).

## Amazon Elastic Block Store (EBS)

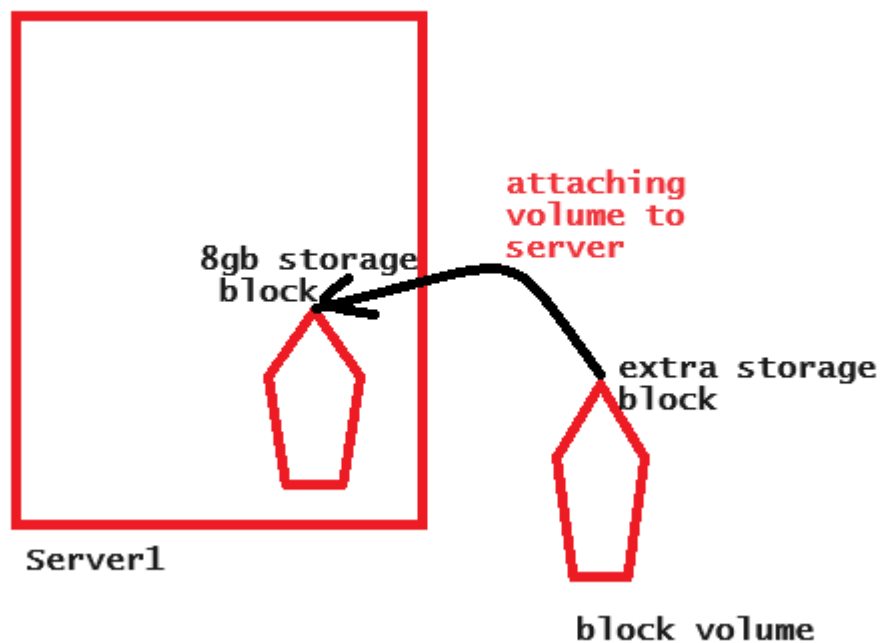
Amazon Elastic Block Store (EBS) is a block storage service provided by Amazon Web Services (AWS). It's designed to provide scalable and highly available block-level storage volumes for use with Amazon Elastic Compute Cloud (EC2) instances and other AWS services. Here are some key points about Amazon EBS:

1. **Block Storage:** EBS provides block-level storage, which is used for storing data that requires a file system, such as the operating system, databases, and application data.
2. **EBS Volume Types:**
  - **General Purpose (SSD):** These volumes, represented by the gp2 type, are designed for balanced performance and are suitable for a wide range of workloads.
  - **Provisioned IOPS (SSD):** These volumes, represented by the io1 type, are designed for applications that require high I/O performance. You can specify the number of IOPS needed.
  - **Throughput Optimized (HDD):** These volumes, represented by the st1 type, are optimized for high throughput workloads, such as big data and data warehousing.
  - **Cold HDD:** These volumes represented by the sc1 type, offer low-cost storage for less frequently accessed data.
3. **Snapshots:** EBS volumes can be backed up using snapshots. Snapshots are point-in-time copies of EBS volumes that are stored in Amazon S3. Snapshots are essential for data backup and disaster recovery.
4. **Encryption:** EBS volumes can be encrypted using AWS Key Management Service (KMS) keys. Encryption enhances data security at rest.
5. **High Durability:** EBS volumes are designed for high durability, with a 99.999% durability rating. This means that data stored on EBS volumes is highly protected against loss.
6. **Attachment to EC2 Instances:** EBS volumes can be attached to EC2 instances, providing persistent storage for the instances. You can attach, detach, and reattach EBS volumes as needed.
7. **Volume Resizing:** You can easily resize EBS volumes to increase or decrease their capacity and performance. Resizing can be done without detaching the volume from the instance.

8. **Multi-Attach (I/O2 Block Express):** Some EBS volume types, like the io2 Block Express, support multi-attach. This allows you to attach a single volume to multiple EC2 instances simultaneously.
9. **Local Snapshots:** EBS Local Snapshots enable creating point-in-time snapshots directly on Amazon EC2 instances without the need for Amazon S3 storage.

Amazon EBS is a fundamental component of AWS's storage services, and it plays a crucial role in ensuring data durability, availability, and scalability in various cloud-based applications and services.

Amazon Elastic Block Store is a versatile and essential component for storing data in the AWS cloud. It offers various volume types to cater to different workloads, making it suitable for a wide range of applications and scenarios.



## Steps to create volume

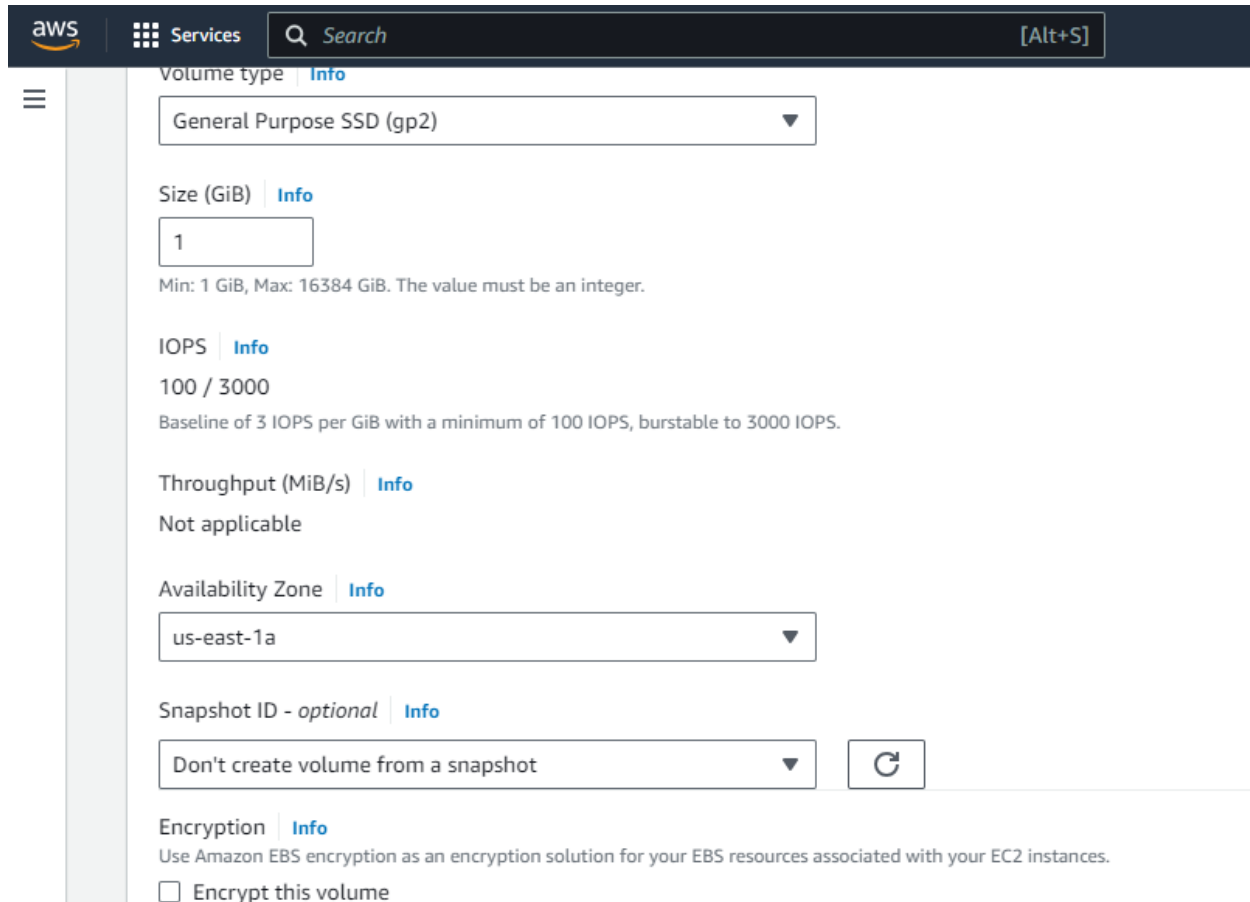
Creating and attaching a volume to an EC2 instance involves a few steps

### Step 1: Navigate to EC2 Dashboard

Log in to your AWS Management Console, go to the EC2 dashboard.

### Step 2: Create a Volume

- In the EC2 dashboard, under "Elastic Block Store," select "Volumes."
- Click on "Create Volume."
- Choose your specifications (size, availability zone, etc.) and click "Create."

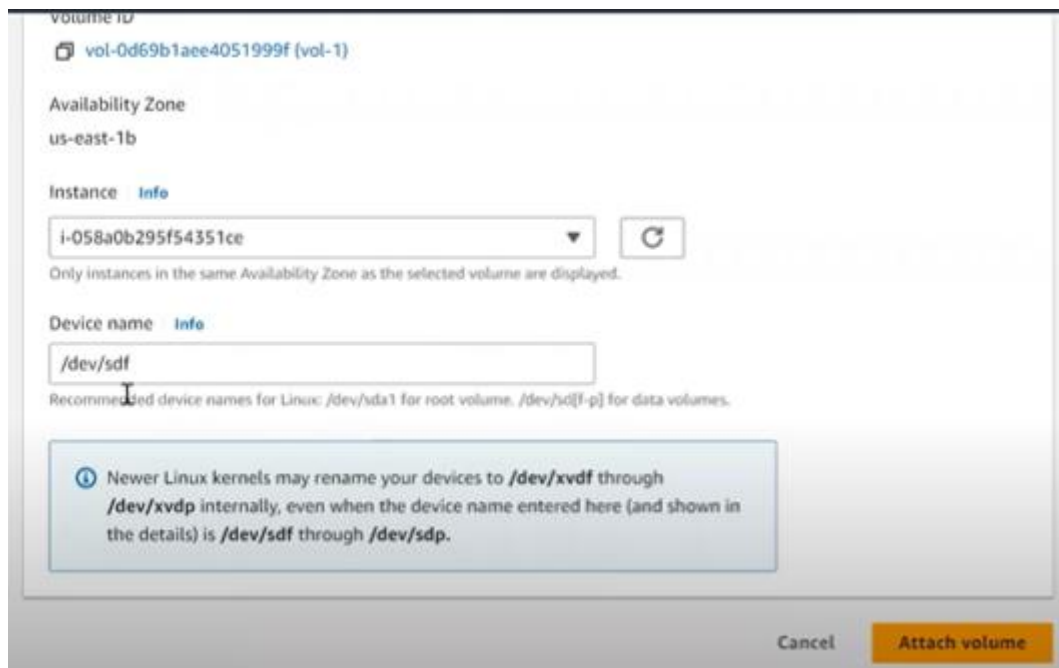
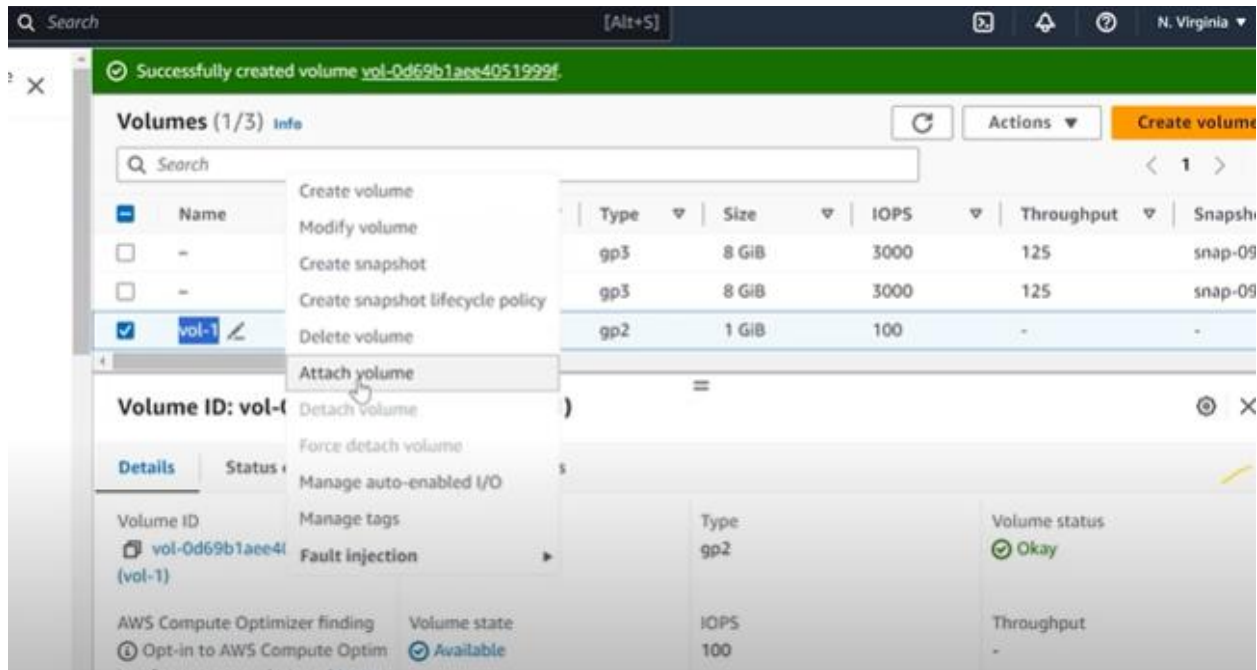


The screenshot shows the AWS Management Console interface for creating a new EBS volume. The top navigation bar includes the AWS logo, 'Services', a search bar, and a keyboard shortcut '[Alt+S]'. The left sidebar shows a menu icon. The main content area is titled 'Volume type' and includes an 'Info' link. Below this, there are several configuration sections: 'General Purpose SSD (gp2)' is selected in a dropdown menu; 'Size (GiB)' is set to '1' with a minimum of 1 GiB and a maximum of 16384 GiB; 'IOPS' is set to '100 / 3000' with a baseline of 3 IOPS per GiB; 'Throughput (MiB/s)' is 'Not applicable'; 'Availability Zone' is set to 'us-east-1a'; 'Snapshot ID - optional' is set to 'Don't create volume from a snapshot'; and 'Encryption' is set to 'Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.' with a checkbox for 'Encrypt this volume'.

### Step 3: Attach the Volume to an Instance

- Still in the Volumes section, select the newly created volume.
- Click "Actions" and then "Attach Volume."

- Choose the EC2 instance to attach the volume to and click "Attach."



#### Step 4: Connect to the EC2 Instance

You'll need to connect to your EC2 instance to manage the volume:

- Go back to the EC2 Dashboard.
- Under "Instances," select your instance.

- Click on "Connect" to get the connection details.

### Step 5: Access the EC2 Instance

Connect to your EC2 instance using SSH or another method based on your instance's operating system.

### Step 6: Identify the New Volume

Once connected to your instance, you'll need to identify the newly attached volume. You can use the following command:

**lsblk**

```
[root@ser1 ~]#  
[root@ser1 ~]#  
[root@ser1 ~]# df -Th  
Filesystem      Type      Size  Used Avail Use% Mounted on  
devtmpfs        devtmpfs  4.0M   0    4.0M   0% /dev  
tmpfs           tmpfs     475M   0    475M   0% /dev/shm  
tmpfs           tmpfs     190M   2.8M 188M   2% /run  
/dev/xvda1      xfs       8.0G   1.5G  6.5G  19% /  
tmpfs           tmpfs     475M   0    475M   0% /tmp  
tmpfs           tmpfs     95M    0    95M    0% /run/user/1000  
[root@ser1 ~]#  
[root@ser1 ~]#  
[root@ser1 ~]# lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS  
xvda        202:0    0   8G  0 disk  
├─xvda1     202:1    0   8G  0 part /  
├─xvda127   259:0    0   1M  0 part  
├─xvda128   259:1    0  10M  0 part  
└─xvdf1     202:80   0   1G  0 disk
```

This will list all block devices attached to your instance.

### Step 7: Mount the Volume

Assuming the new volume is, for example, /dev/xvdf, you'll want to create a file system on it and mount it:

**mkfs.ext4 /dev/xvdf**

**mkdir /madhu**

**mount -t ext4 /dev/xvdf /madhu**

```
root@ser1:~  
[root@ser1 ~]#  
[root@ser1 ~]# mkfs.ext4 /dev/xvdf  
mke2fs 1.46.5 (30-Dec-2021)  
Creating filesystem with 262144 4k blocks and 65536 inodes  
Filesystem UUID: dbeb4cb6-30c4-4eec-9c5d-fed8df4ed6bc  
Superblock backups stored on blocks:  
    32768, 98304, 163840, 229376  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (8192 blocks): done  
Writing superblocks and filesystem accounting information: done  
  
[root@ser1 ~]#  
[root@ser1 ~]# mkdir /madhu  
[root@ser1 ~]#  
[root@ser1 ~]#  
[root@ser1 ~]#  
  
root@ser1:~  
[root@ser1 ~]#  
[root@ser1 ~]# mount -t ext4 /dev/xvdf /madhu  
[root@ser1 ~]#  
[root@ser1 ~]# df -Th  
Filesystem      Type      Size  Used Avail Use% Mounted on  
devtmpfs        devtmpfs  4.0M   0    4.0M   0% /dev  
tmpfs           tmpfs     475M   0    475M   0% /dev/shm  
tmpfs           tmpfs     190M   2.8M 188M   2% /run  
/dev/xvda1      xfs       8.0G   1.5G  6.5G  19% /  
tmpfs           tmpfs     475M   0    475M   0% /tmp  
tmpfs           tmpfs     95M    0    95M   0% /run/user/1000  
/dev/xvdf       ext4      974M   24K   907M   1% /madhu  
[root@ser1 ~]#
```

### Step 8: Verify

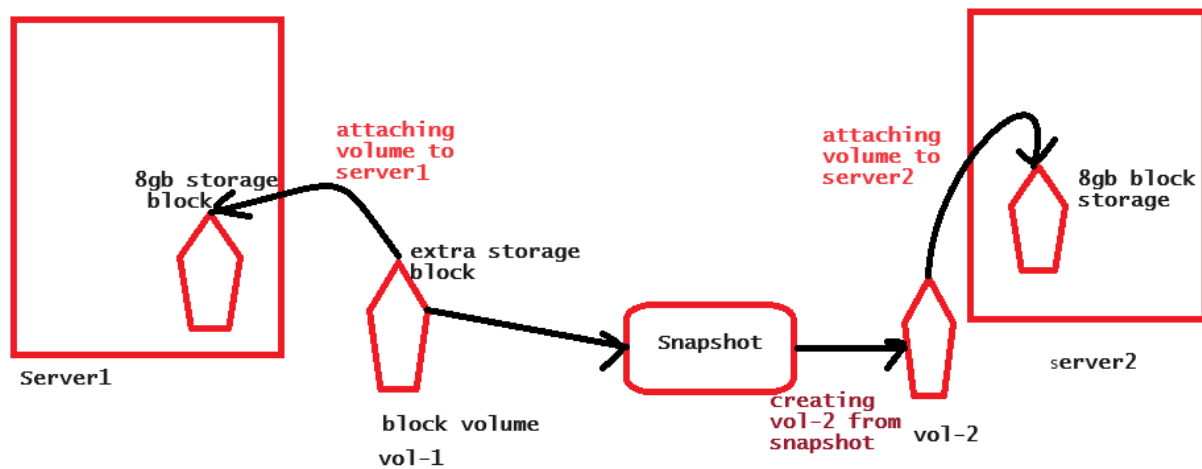
Verify that the volume is mounted correctly

**df -Th**

You've created and attached a volume to your EC2 instance.



## Creating a snapshot from volume1 of server1 and creating volume from snapshot and attach to the volume to server2



### Step 1: Create Snapshot from Volume1 on Server1

- In the EC2 dashboard, under "Elastic Block Store," select "Volumes."
- Identify Volume1 attached to Server1 and select it.
- Click on "Actions" and then "Create Snapshot."
- Give your snapshot a meaningful name and description, then click "Create Snapshot."

### Step 2: Create a Volume from the Snapshot

- Click on "Snapshots" in the left navigation pane.
- Select the snapshot you just created.
- Click on "Actions" and then "Create Volume."
- Specify the volume size and availability zone (make sure it's the same as Server2's availability zone).
- Click "Create Volume."

### Step 3: Navigate to Volumes in the Same Region

- Still in the EC2 dashboard, under "Elastic Block Store," select "Volumes."

### Step 4: Attach the New Volume to Server2

- Once the new volume is created, go back to "Volumes" in the EC2 dashboard.

- Select the newly created volume.
- Click "Actions" and then "Attach Volume."
- Choose Server2 as the instance to attach the volume to and click "Attach."

#### Step 5: Connect to Server2 and Mount the Volume

- Connect to Server2 and identify the new volume using `lsblk`.

```
root@ser2:~
[root@ser2 ~]#
[root@ser2 ~]#
[root@ser2 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda         202:0    0   8G  0 disk
├─xvda1      202:1    0   8G  0 part /
├─xvda127    259:0    0   1M  0 part
└─xvda128    259:1    0  10M  0 part
[root@ser2 ~]#
```

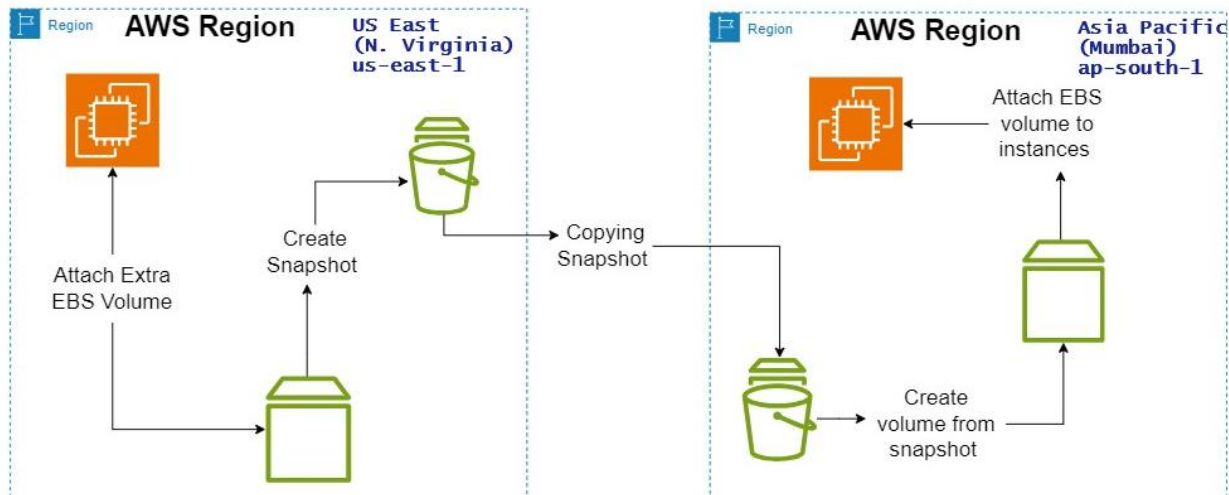
- Create a file system and mount the volume as needed.
- **`mkdir /kiran`**
- **`mount -t ext4 /dev/xvdf /kiran`**

#### Step 6: Verify

- Verify that the volume is correctly mounted using **`df -Th`**.

```
[root@ser2 ~]# mkdir /kiran
[root@ser2 ~]#
[root@ser2 ~]# mount -t ext4 /dev/xvdf /kiran
[root@ser2 ~]# df -Th
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  4.0M   0    4.0M  0% /dev
tmpfs           tmpfs     475M   0    475M  0% /dev/shm
tmpfs           tmpfs     190M  2.8M  188M  2% /run
/dev/xvda1      xfs       8.0G  1.5G  6.5G  19% /
tmpfs           tmpfs     475M   0    475M  0% /tmp
tmpfs           tmpfs     95M    0    95M  0% /run/user/1000
/dev/xvdf       ext4      974M  28K  907M  1% /kiran
[root@ser2 ~]# |
```

## Creating a volume in another region using a snapshot from a different region



### Step 1: Navigate to Snapshots in the Source Region

- In the AWS Management Console, go to the EC2 dashboard.
- Under "Elastic Block Store," click on "Snapshots."
- Select the snapshot from the source region that you want to use.

### Step 2: Copy Snapshot to the Destination Region

- Click on "Actions" and then "Copy Snapshot."
- Choose the destination region where you want to create the volume.
- Configure other settings as needed and click "Copy."

**New snapshot settings**

**Description**  
A description for the snapshot copy.

sn1

255 characters maximum.

**Destination Region**  
The Region in which to create the snapshot copy.

ap-south-1

**Encryption Info**  
Use Amazon EBS encryption as an encryption solution for your EBS resources.

☐ Encrypt this snapshot

**Step 3:** Navigate to Volumes in the Destination Region

- Switch to the destination region in the AWS Management Console.
- Under "Elastic Block Store," click on "Volumes."

**Step 4:** Create Volume from Copied Snapshot

- Click on "Create Volume."
- In the "Snapshot" dropdown, select the snapshot that was copied from the source region.
- Specify the volume size and other settings as needed.
- Click "Create Volume."

**Step 5:** Attach the Volume to an Instance

- Once the volume is created, select it from the list.
- Click on "Actions" and then "Attach Volume."
- Choose the EC2 instance in the destination region to attach the volume to.
- Click "Attach."

**Step 6:** Connect to the EC2 Instance and Mount the Volume

- Connect to your EC2 instance in the destination region.
- Identify the newly attached volume using `lsblk`.
- Create a file system on the volume and mount it as needed.
- Verify that the volume is correctly mounted using `df -h`.

**!!!!Caution!!!!**

**Make sure delete all volumes and snapshots which created by you in all regions**

**Delete Volumes:**

1. In the AWS Management Console, go to the EC2 dashboard.
2. Under "Elastic Block Store," click on "Volumes."
3. Select the volumes you want to delete.
4. Click on "Actions" and then "Delete Volume."

5. Confirm the deletion when prompted.

#### **Delete Snapshots:**

1. In the AWS Management Console, go to the EC2 dashboard.
2. Under "Elastic Block Store," click on "Snapshots."
3. Select the snapshots you want to delete.
4. Click on "Actions" and then "Delete Snapshot."
5. Confirm the deletion when prompted.

**Make sure to double-check and confirm the deletion to avoid any accidental removal of important resources. Once you've deleted the volumes and snapshots, your AWS environment will be clean from the temporary resources you created.**