

Elastic Block Store

Amazon Elastic Block Store (EBS) is a fundamental component of Amazon Web Services (AWS), providing scalable and highly available block-level storage volumes for EC2 instances. It provides persistent block-level storage volumes for Amazon EC2 instances. These volumes behave as raw, unformatted block devices and can be mounted on instances. They persist independently from the instance's lifecycle and allow the creation of file systems or usage as block devices (similar to hard drives). EBS volumes are dynamically configurable and are recommended for data requiring quick access and long-term persistence. They are ideal for primary storage in file systems, databases, or applications needing fine-grained updates and raw block-level storage access. EBS suits both random read/write database applications and throughput-intensive tasks involving continuous reads and writes. Amazon EBS provides a reliable storage solution for critical workloads, ensuring data persistence, backup, and disaster recovery capabilities. Here are the key aspects of Amazon EBS:

- **Durability and Redundancy:** EBS volumes automatically replicate within the same Availability Zone (AZ) to protect against component failure. For additional fault tolerance, you can create snapshots of EBS volumes and copy them across different AZs or regions.
 - **Elasticity:** EBS volumes can be easily resized, allowing you to adjust your storage capacity as your requirements change. This flexibility eliminates the need for over-provisioning, ensuring cost efficiency.
 - **Performance:** EBS provides different volume types optimized for various use cases, offering varying levels of I/O performance. From general-purpose SSD volumes to high-performance provisioned IOPS SSD volumes, EBS enables you to choose the right storage type for your applications.
- | | |
|--|--|
| Snapshots: EBS allows you to create | point-in-time snapshots of volumes. These snapshots are incremental backups, capturing only the changed data since the last snapshot. Snapshots are stored in Amazon S3, making them highly durable and accessible across different regions. |
|--|--|

Types of Amazon EBS Volumes

- Amazon EBS offers multiple volume types, each tailored to specific use cases and performance requirements:
- **General Purpose SSD (gp2):** General-purpose SSD volumes provide a balance of price and performance. They are suitable for a wide range of workloads, including small to medium-sized databases and development/test environments.
 - **Provisioned IOPS SSD (io1):** Provisioned IOPS SSD volumes deliver high-performance storage for I/O-intensive applications, such as large relational databases or NoSQL databases. You can provision a specific level of IOPS based on your application requirements.
 - **Cold HDD (sc1):** Cold HDD volumes offer low-cost storage optimized for infrequent access workloads, such as data warehouses and backup storage. While they provide lower throughput than SSD volumes, they are cost-effective for storing large amounts of data.
 - **Throughput Optimized HDD (st1):** Throughput Optimized HDD volumes deliver low-cost storage optimized for frequently accessed, throughput-intensive workloads. They are ideal for big data and data warehouse applications that require high throughput.
 - **Magnetic (standard):** Magnetic volumes, also known as standard volumes, provide the lowest cost per gigabyte of all EBS volume types. They are suitable for workloads with low I/O performance requirements, such as archival storage or development environments.

Instance Storage

- Instance storage, also known as ephemeral storage, refers to the temporary block storage directly attached to an Amazon EC2 instance. This storage is physically connected to the host machine where the instance runs. Instance storage provides high I/O performance and low latency, making it ideal for applications requiring fast data access, such as cache storage or temporary processing.
- **Volatility:** Instance storage is volatile, meaning the data stored on it is lost if the associated instance is stopped, terminated, or if the underlying hardware fails. It is intended for temporary storage and is not suitable for critical, long-term data.
 - **Auto-Delete:** Instance storage is automatically deleted when the associated EC2 instance is terminated, and it cannot be detached or reattached to another instance.
 - **Use Cases:** Instance storage is suitable for stateless applications, temporary storage of intermediate data, and high-performance computing tasks where data can be recomputed if lost.

Amazon Elastic Block Store (EBS)

- Amazon EBS, on the other hand, provides persistent block-level storage volumes that can be attached to Amazon EC2 instances. Unlike instance storage, EBS volumes are network-attached and persist independently from the life of an instance. EBS volumes are suitable for data that requires long-term persistence, durability, and the ability to survive instance failures.
- **Durability:** EBS volumes are highly durable and can be configured for replication within an Availability Zone (AZ).

Snapshots of EBS volumes are stored in Amazon S3, providing additional data protection.

- **Persistence:** EBS volumes persist even if the associated EC2 instance is stopped or terminated. They can be detached from one instance and attached to another, allowing data mobility.
- **Use Cases:** EBS volumes are ideal for databases, file systems, application data, and any workload that requires reliable, long-term storage. They are suitable for mission-critical applications where data durability and persistence are essential.

Differences between EBS and Instance Storage

- **Durability and Persistence:** EBS provides durable and persistent storage, while instance storage is volatile and temporary.
- **Data Loss:** Data stored in instance storage is lost if the instance is terminated. EBS data persists even if the instance is terminated or stopped.
- **Use Cases:** Instance storage is suitable for temporary, stateless applications, whereas EBS is designed for long-term data storage and applications requiring data persistence.
- **Availability:** EBS volumes can be accessed from multiple instances, providing data accessibility even if one instance fails. Instance storage is specific to a single instance and cannot be shared or accessed by other instances.

EBS Operations

These are the steps to create EBS volume

1. Go to EC2 Console: Visit the Amazon EC2 website.
2. Navigate to Volumes: In the menu, click on **Volumes**.
3. Create a Volume: Select **Create volume**.
4. Choose Type: Decide the type of volume you want. Different types have different features.
5. Specify Size: Enter the size of the volume in gigabytes (GiB).
6. (Optional: IOPS and Throughput: Depending on the type, specify the maximum input/output operations per second (IOPS) or throughput in MiB/s.)
7. Select Availability Zone: Choose the specific zone where the volume will be located. It must match the zone of your instance.
8. Encryption: Decide whether to enable encryption. If you choose to encrypt, select the encryption method and key.
9. Tags (Optional): You can add custom tags to the volume for better organization and identification.
10. Create the Volume: Click **Create volume**.
11. To use the volume, attach it to an instance.

Attach EBS Volume to EC2 Instance

1. Go to EC2 Console: Visit the Amazon EC2 website.
2. Navigate to Volumes: Click on **Volumes** in the menu.
3. Select the Volume: Choose the volume you want to attach, then click **Actions** and select **Attach volume**.
4. Choose an Instance: Enter the ID of the instance or select it from the list. The volume and the instance must be in the same Availability Zone.
5. Device Name: Enter a name for the volume. This name is used by Amazon EC2, although it might get a different name when mounted on the instance.
6. Attach the Volume: Click **Attach volume**.

After attaching an Amazon EBS volume to your instance, it appears as a block device. You can format and mount it like any other volume. Data written to this volume is transparent to applications. Snapshots can be taken for backup or to create new volumes. Volumes larger than 2 TiB require a GPT partitioning scheme. Windows instance users can find specific instructions in the Amazon EC2 User Guide for Windows Instances.

Enable Amazon EBS Volume for Linux Use

To make a new Amazon EBS volume ready for use on a Linux system:

Connect to Your Instance: Use SSH to connect to your Linux instance.

Check Device Name and File System:

```
lsblk
```

Output:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf 202:80 0 10G 0 disk
```

Check Existing File System (Optional):

```
sudo file -s /dev/xvdf
```

Output (If no file system exists):

```
/dev/xvdf: data
```

Create File System (If Needed):

```
sudo mkfs -t xfs /dev/xvdf
```

Output (If mkfs.xfs is not found, first install XFS tools using `sudo yum install xfsprogs`).

Create Mount Point:

```
sudo mkdir /data
```

Mount the Volume:

```
sudo mount /dev/xvdf /data
```

Now, the Amazon EBS volume is mounted at the `/data` directory and ready for use.

Enable Amazon EBS Volume for Windows Use

Steps to Make an EBS Volume Accessible Using DiskPart in Windows:

- Login to Your Windows Instance:
- Access your Windows instance via Remote Desktop.
- Determine Disk Number:
- Open Start and select Windows PowerShell.
- Use the `Get-Disk` Cmdlet to list available disks.
- Note the Number corresponding to the disk you want to make accessible.
- Create a DiskPart Script:
- Open File Explorer and go to a directory (e.g., C:\) to store the script file.
- Right-click an empty space, select New > Text Document.
- Name the file `diskpart.txt`.
- Edit the Script:
- Open `diskpart.txt` and add the necessary commands, adjusting disk number, partition type, volume label, and drive letter if needed.
- Be cautious not to reformat a volume with existing data.
- Example Commands:

```
select disk 1
attributes disk clear readonly
online disk noerr
convert mbr
create partition primary
format quick fs=ntfs label="volume_label"
assign letter="drive_letter"
```

- Run DiskPart Script:
- Open a command prompt, and navigate to the folder containing the script.
- Execute the following command to make the volume accessible:

```
C:\> diskpart /s diskpart.txt
```

Unmount and detach a volume

- Unmount the Volume: From your Linux instance, execute the following command to unmount the specified device (/dev/sdh).

```
[ec2-user ~]$ sudo umount -d /dev/sdh
```

- Detach the Volume from the Instance:
- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, click on **Volumes**.
- Select the volume you want to detach.
- Choose **Actions**, then click **Detach Volume**.
- Confirm the action by clicking **Detach**.

Key Points on Detaching Amazon EBS Volumes:

1. Detaching and Data Preservation:
2. Detaching an Amazon Elastic Block Store (Amazon EBS) volume from an instance is necessary to attach it elsewhere or delete it.
3. Importantly, detaching a volume does not impact the data stored on it.
4. Explicit Detachment and Instance Termination:
5. You can detach an EBS volume explicitly or as part of terminating an instance.
6. When the instance is running, it's crucial to unmount the volume before detaching it.
7. Special Consideration for Root Devices:
8. If an EBS volume serves as the root device of an instance, the instance must be stopped before detaching the volume.
9. Reattaching Volumes:
10. It's possible to reattach a detached volume without unmounting it.
11. However, the reattached volume might not retain the same mount point.
12. If there were ongoing writes to the volume during detachment, data synchronization might be required.
13. Storage Charges and Deletion:
14. Even after detaching, you continue incurring charges for volume storage if it exceeds the AWS Free Tier limit.
15. To stop further charges, you must delete the volume. Refer to **Delete an Amazon EBS volume** for detailed instructions.

Delete an Amazon EBS volume

When you're done using an Amazon EBS storage, you can remove it. Once removed, all the data is gone, and you can't use it with any Instance anymore. But, before you remove it, you can save a copy of all the data (called a snapshot). Later, if you need the storage again, you can use this snapshot to make a new one.

To delete an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume to delete and choose **Actions, Delete volume**.
4. If **Delete volume** is greyed out, the volume is attached to an instance. You must detach the volume from the instance before it can be deleted.
5. In the confirmation dialog box, choose **Delete**.