

AWS EBS Service

(Elastic block store)

AWS EBS is a cloud based storage service that provides durable, high-performance block storage for use with Amazon EC2 instances.

It works like a virtual hard drive, allowing you to store and access data even when your EC2 instances are stopped or terminated.

Key Features of Amazon EBS

1. Persistent Storage :- Data persists independently of the EC2 instance lifecycle.
2. Block-Level Storage :- Function like a physical hard drive.
3. Scalable :- Can scale up to 64 TiB per volume.
4. High Performance :- Choose between different performance modes (SSD, HDD).
5. Snapshots :- Create backups (snapshots) of volume to S3.
6. Encryption :- Supports encryption at rest and in-transit using AWS KMS.

EBS volume → Configure Storage → Advance → Encryption with key

26

Availability: Automatically replicated within the same Availability Zone (AZ).

- If block destroy data is recoverable but if zone destroy data is inaccessible so
- Right way to make snapshots to S3 (internally S3 bucket, not access through S3, access through snapshots).

EBS volume types

Types	Best for	Input / Output operation per second	Theory throughput
gp3 (General purpose SSD)	Balance workloads	up to 16000	1000 MB/s
iop2/iop3 Block Express (provisioned IOPS SSD)	mission critical apps like database	up to 256,000	4000 MB/s
st1 (throughput optimized HDD)	Big data, log processing	lower IOPS	High throughput
sc1 (cold HDD)	infrequent accesses, backup	lower cost	lowest performance.

27

Common Usage of EBS

- 1. Boot volume for EC2 instances
 - = Act as the root volume when launching EC2
 - = Store the OS & startup config.
 - = persistent even after instance stop / start.
- 2. Database storage & even if EC2 stopped or restarted
 - = Ideal for hosting structured & NO SQL database.
 - MySQL, PostgreSQL, Oracle, SQL Server.
 - MongoDB, Cassandra, Redis etc.
- 3. File System:-
 - = Mount EBS volumes as file system like:
ext4, XFS, NTFS (Windows)
 - useful for storing application data, logs, media files etc.

File System :- define how data is structured, indexed, accessed, and managed on the disk.

4. Application Storage

- = Support apps that require:
 - High throughput
 - Low latency storage.
- = Use with auto-scaling groups for dynamic workloads

(28)

5. Backup & Disaster Recovery

- Create snapshots that stored in Amazon S3 (indirectly).
- In incremental form (only changed block saved).
- For backup & restore across time.

6. Attachment

- We can attach multiple EBS to a instances.
- Or only allow you to attach a single iO1 & iO2 volume to multiple EC2 instances in same Availability Zone (AZ).

Important points about EBS

- ⇒ Region and AZ specific
- ⇒ Allow encryption & snapshot for backup.
- ⇒ Scalable (volume can be resizeable).
- ⇒ No need to restart the EC2 instance during the process.

Creation of EBS Volume

1. With EC2 instance → At last see create or add volume.
2. Create EBS volume → Search EBS (Elastic Block Store)
→ Volumes → Create Volume → Customizing according to you.

②9

Note: Volume & EC2 instance need to be in same availability zone.

Otherwise you cannot attach directly, need the use of snapshots.

Termination on delete

When you launch a EC2 instance with an EBS volume, there is a setting called :-

"Delete on Termination".

→ If enable (Yes (V)) (By default).

When the EC2 instance is terminated, the EBS volume is also deleted automatically.

→ If disable (No (X))

when the EC2 instance is terminated, the EBS volume persists, and you can reuse it, attach to another instance, or snapshot it.

→ How to Control It →

→ first go to EC2 instance → go to storage → check the Delete on termination → if Yes then make it NO → for best practice.

→ Do it as according to create new instance go to scroll down on create new instance

→ click advance in configure storage.

→ make "no" → for Delete on termination.

(30)

Practical

Create EBS and attaching to EC2 instance

To Step-1 If launch an EC2 instance (if not already done)

1. Go to EC2 dashboard → click "Launch instance"

2. choose:- .AMI

- Instance type
- key pair

3. Add storage, → customize or default.

4. Launch the instance.

Now EC2 is running

Step-2 : Create an EBS Volume

To Navigate to EC2 → Elastic Block Store → Volumes.

2. click "Create volume".

3. fill in:

- Size: eq 8 GiB
- Availability zone → must same as instance
- Volume type → purposes.
- Leave the rest as default.

4. Click create volume.

Step-3 Attached EBS Volume to EC2 instance

To After creating, select the volume → Click "Action" → "Attach Volume"

③

2. Select the EC2 instance ID.
3. Device name (Linux) : /dev/xvdf (default if fine).

4. Click Attach.

Now volume is attached, but not yet mounted.

Step 4 :- Mount the Volume:-

1. Click on selected EC2 instance id in instances.
2. Click on "connect" → EC2 instance connect
→ "Connect" → here you go on terminal.

3. Check if volume is detected.

- lsblk → type in terminal
4. Format the disk (only first time to make proper read/write)

sudo mkfs -t ext4 /dev/xvdf

5. Create a mount directly.

sudo mkdir -p /mnt/ebs

6. Mount it

sudo mount /dev/xvdf /mnt/ebs

7. (optional) Check it

df -h Now the volume is mounted & accessible.

→ You can read/write data.

But → If the instance reboots, this mount is lost
- we will have to manually mount it again →
To this solved by permanent mount.

Permanent mount
↳ Get the UUID of the volume

sudo blkid /dev/xvdf.

Let say the output is

/dev/xvdf : UUID = "abc123-456def" TYPE = "ext4"

2. Add to /etc/fstab

sudo nano /etc/fstab

Add this at the bottom (replace UUID by your is different).

UUID=abc123-456def /mnt/mydata ext4

defaults, nofail 0 2

3. final check.

sudo mount -a.

{ if type != "ext4"
 else
 sudo mount -o nouuid /dev/xvdf /mnt/ebs
 on mounting }

Step-5 Unmount

↳ sudo umount /mnt/ebs

or

sudo umount -l /mnt/ebs

df -h for checking.

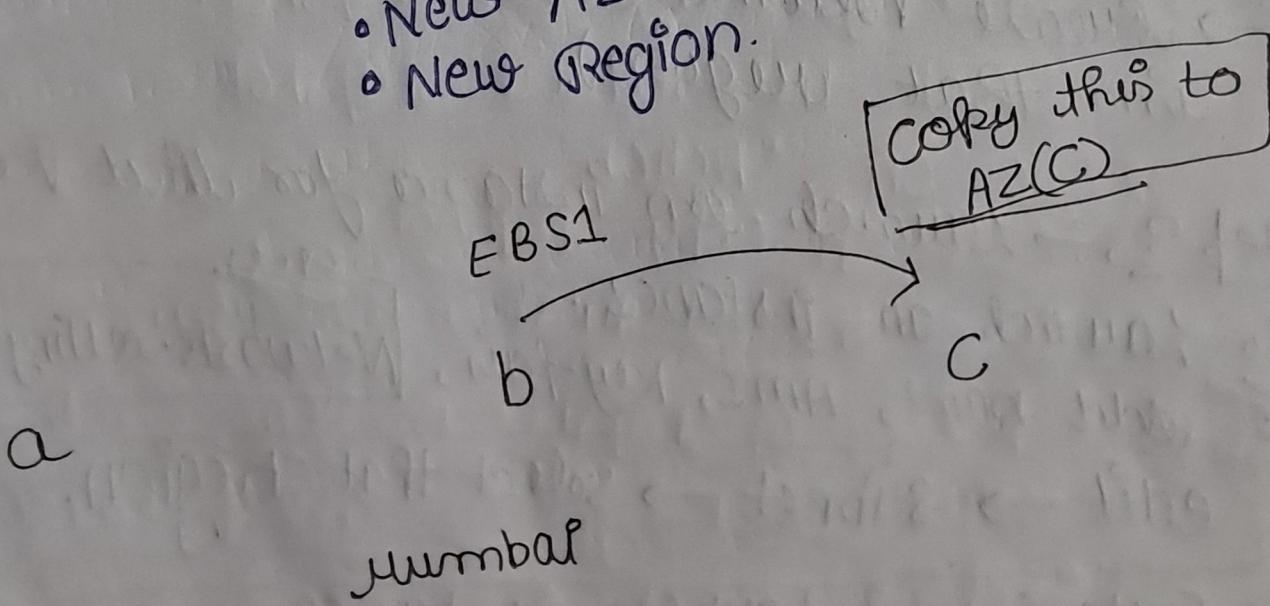
(33)

Modify Size EBS Volume

1. Go to Volumes.
2. Click on check which need to modify.
3. Action → Modify Volume → modify it.
4. Click modify → modified.

Delete & Detach EBS

1. Go to Volumes.
 2. Select which you need to detached.
 3. Click "Action" → Detach volume.
 4. Detached.
 5. If want delete again click "Action".
 6. Delete volume → deleted.
- # EBS Snapshot → what if we want to copy our data to
 • New AZ
 • New Region.



⁽³⁴⁾
Now → practical make Snapshot and save & mount in another AZ. (in same Region)

Step-1 Create Snapshot

1. Select volume for which snapshot require → click "Action" → Create Snapshot → add description → Create → created.

2. Go to → EBS → Snapshot at the left bar.
→ select it by checked → it need to be complete as status.

3. click → "Action" → Create volume from Snapshot.

Step-2 Create Volume in required region

1. After click on Create volume from Snapshot → change the availability zone → Create Volume. (Now new Volume Created in that region).

Step-3 Launch an instance for that Region.

1. Launch an instance.

2. Add name, AMI, key-pair, Network-setting → edit → subnet → select that Region.

3. Customize & Launch instance.

- ③
- ### Step-4 Connect to terminal & Mount
- 1 Go to that New instance.
 - 2 Select it and click on "Connect". → connect.
 - 3 terminal is opened.
- (i) ls → nothing is shown
- (ii) lsblk → show that volume, that is already built with instance.

Step-5; Attach snapshot volume to instance of that region.

- 1 Go to that volume.
- 2 Click → "Action" → Attached Volume.
- 3 Attached.

Step-6 Mount using terminal

- (i) lsblk → volume attached
- (ii) check the new volume name that add recent by adding.
- (iii) read that name of that with $x(G)$ → how this or show GRB data. (eg: nvme1n1p1)
- (iv) sudo file -s /dev/nvme1n1p1
display → xfs filesystem data → means filesystem attached data.
- v2 sudo mkdir /mnt/mybackup

③

(vi) sudo mount /dev/nvme1n1p1 /mnt/mybackup
or

sudo mount -o nouuid /dev/nvme1n1p1 /mnt
/backup

(vii) check mount → df -h (mounted)

Step-7 Unmount

(i) sudo umount /mnt/mybackup
or

sudo unmount -l /mnt/mybackup

(ii) df -h (checking)

Copy Snapshot from One Region to
Another Region

~~Step-1~~ - Select from left bar Snapshot.

→ Select the require snapshot

→ Click On "Action"

→ Click On "Copy Snapshot"

→ Select the destination region

~~Step-2~~ - A copy of snapshot is form on that
region and follow all previous
Steps

(37)

Lifecycle Manager EBS

Amazon Data Lifecycle Manager → Automate the creation, retention, copy & deletion of snapshots and AMIs.

- There is no charge using Lifecycle Manager.
- this is just a policy.
- Schedule automatic EBS snapshots,
- And delete old snapshots,

Step 1 → first make key-value of that volume.
→ click "Volume"
→ Go to Tags
→ click "Manage Tags"
→ Add tag → As key = ENV
value = DAILY

Step 2 → Go to Lifecycle Manager → Select policy type

→ Create lifecycle custom policy

→ Add key pair value of volume.

→ Add description

→ Then make customize or other default

→ Click "Next".

Step 3 → Customize all things like, frequency, Retention type, time, & copy cross region.