EBS

> **Amazon EBS = cloud hard disk for your EC2 instance.**

- You can think of it as the **C: drive or D: drive** of a computer — but in the cloud.

🧠 Real-World Analogy:

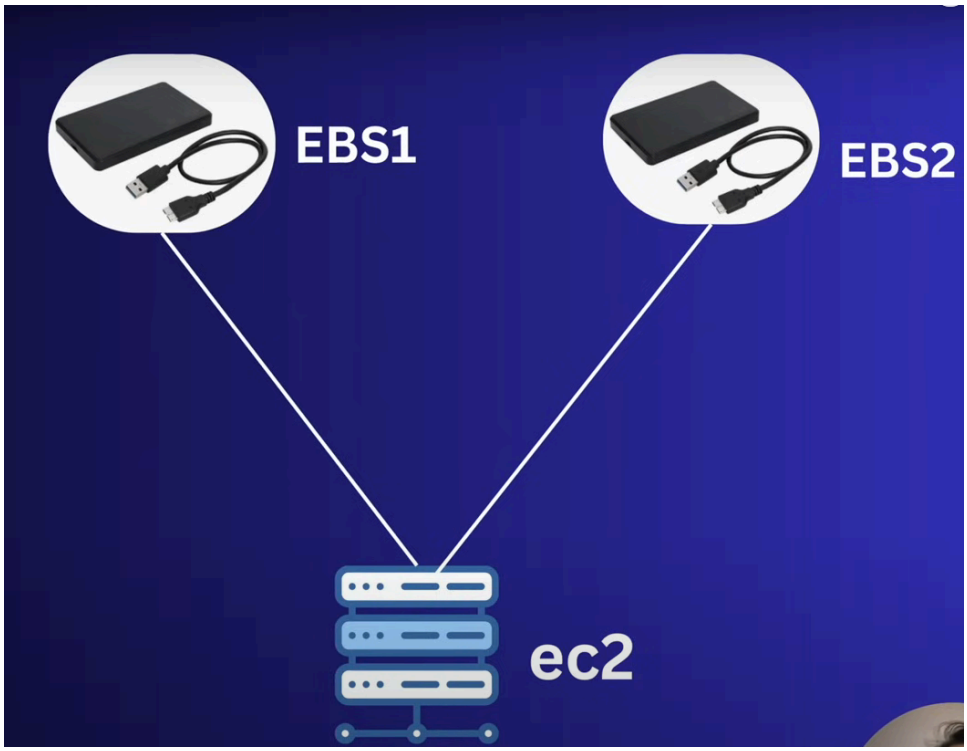
Imagine EC2 is your **cloud laptop**.

- EBS is the **hard drive** where you store files, data, code, logs, model weights, etc.
- You can **stop and start EC2** freely, and your EBS **remembers your data** just like your laptop's SSD.

Without EBS, EC2 would be **stateless** (every reboot = clean slate).

🛠️ What is EBS?

- **Network-attached storage** for EC2 instances (like an external SSD, but in the cloud).
- **Persistent**: Data stays even if you stop/restart your EC2 instance.
- **Scalable**: Resize or change volume types without downtime.



- **We can attach multiple EBS to a single ECE instance.**

✅ Why EBS Is Used?

Use Case	Why EBS Helps
🧠 Machine Learning	Store large datasets or models that persist even if EC2 stops
📝 File Storage	Upload code, logs, CSVs, notebooks
📦 Install software	Everything on the EC2 (Ubuntu, Python, pip) sits on EBS
🔁 Stop and restart EC2	Your data and configuration persist
📂 Multiple volumes	You can attach extra EBS drives for separating data

> 💡 **EBS is Region & AZ spacific.**

🔑 Key Features

Feature	Why It Matters
Volume Types	Choose between SSD (fast) or HDD (cheap) based on your needs.
Snapshots	Backup your data (like a "save point" for disasters).
Encryption	Secure data at rest (AES-256) by default.

💰 EBS Pricing (Simplified)

Type	Use Case	Cost (US East, per GB/month)
gp3 (SSD)	Default for most workloads	$0.08 (~₹6.60)
io1 (High IOPS)	Databases (e.g., MySQL)	$0.125 (~₹10.30)
st1 (HDD)	Big data, logs	$0.045 (~₹3.70)

Note: You pay for **allocated** space, not actual usage (e.g., a 100GB volume costs $8/month even if only 10GB is used).

🔍 How EBS Works

- EBS volumes are **network-attached** block devices.
- You can:
  - 📦 **Create** a volume (like a 50 GB disk)
  - 🔗 **Attach** it to an EC2 instance
  - 🛠️ **Mount** it like a Linux partition
  - 📂 Use it like any local folder (`/mnt/volume1`)
- You can also:
  - Take **snapshots** (backups)
  - **Detach** or delete volumes without deleting EC2
  - **Resize** disks without rebooting

📦 EBS Volume Types (Choose Based on Use Case)

Volume Type	Code	Use Case	Speed	Cost
General Purpose SSD	`gp3`	✅ Default for most cases	Fast	₹
Provisioned IOPS SSD	`io1`, `io2`	High-performance DBs	Super fast	₹₹₹
Throughput Optimized HDD	`st1`	Big data & logs	Good	₹
Cold HDD	`sc1`	Archiving, infrequent access	Slow	₹
Magnetic (deprecated)	`standard`	Older type	❌ Avoid	Cheap

✅ **For most beginner EC2 setups**, use `gp3` **(default SSD)**.

🛠️ Where You See EBS in the AWS Console?

When you launch an EC2 instance:

- Under the **"Storage"** section → AWS creates a default **root volume** (usually 8 or 30 GB EBS volume)
- This is where Ubuntu/Apache/Python goes

You can also:

- Add more EBS volumes during launch
- Attach more later from EC2 → **Volumes → Create Volume → Attach to instance**

File Path of EBS in EC2

- Root volume: `/`
- Additional EBS: must be manually mounted (e.g. `/dev/xvdf` → `/mnt/data`)

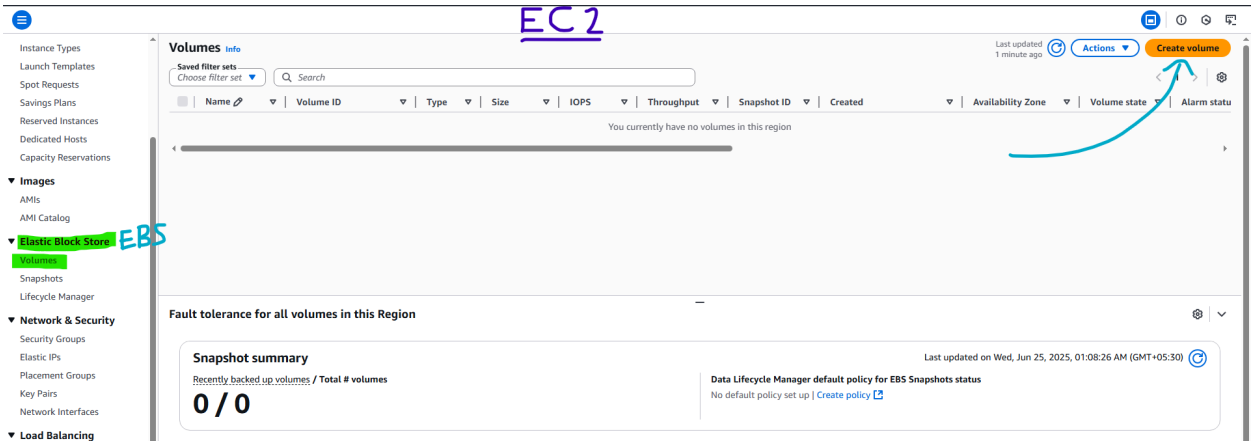
❌ Common Beginner Mistakes with EBS

Mistake	Fix
❌ Leaving volumes unattached	You're charged even if they're not in use — delete them manually
❌ Not deleting volumes when terminating EC2	Tick "Delete on termination" or delete manually
❌ Storing permanent data only on the root volume	Use separate EBS volume for data for easier backup & portability
❌ Not taking snapshots	Take regular backups using EBS Snapshots
❌ Confusing S3 with EBS	EBS = Disk; S3 = Object storage (bucket, not a filesystem)

🚀 How to Use EBS?

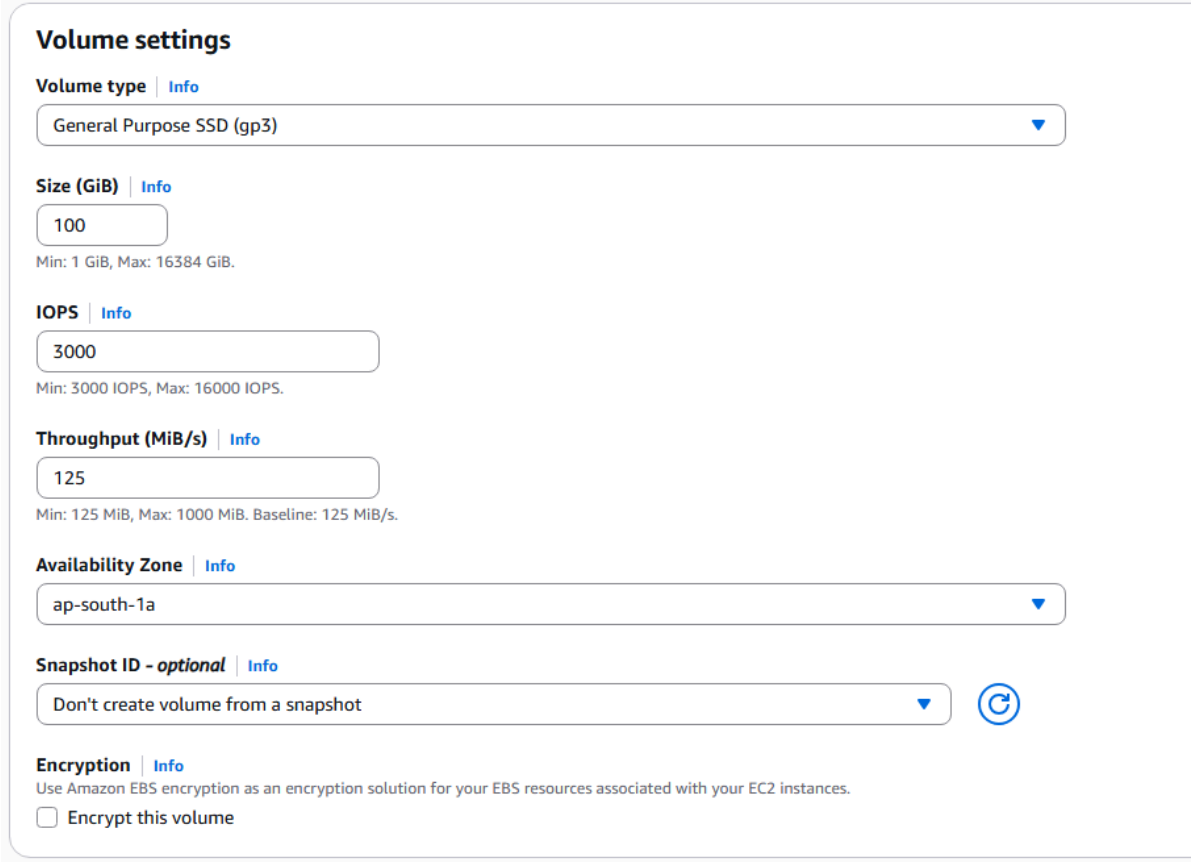
1. Create an EBS Volume

1. Go to **EC2 Console** → **Volumes** → **Create Volume**.



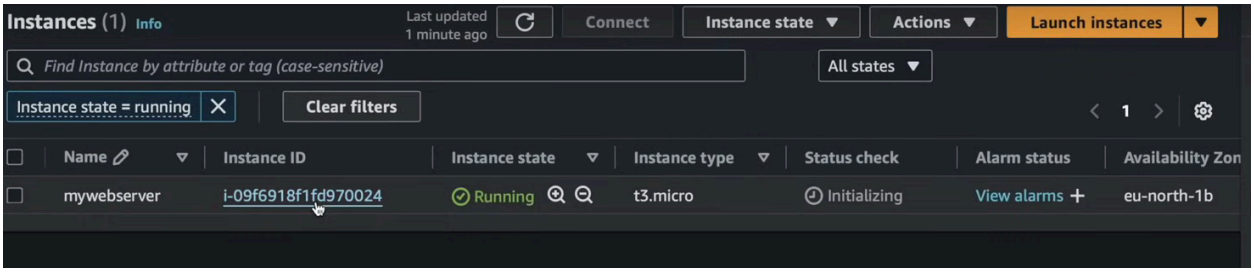
2. Select:

- **Size**: Start with 20GB (free tier eligible).
- **Type**: `gp3` (general purpose).
- **Availability Zone**: Must match your EC2 instance's zone!



2. Attach to an EC2 Instance

1. Right-click the volume → **Attach Volume**.
2. Select your EC2 instance and assign a device name (e.g., `/dev/sdf`).



3. Format & Mount (Linux Example)

Connect to your EC2 instance via SSH and run:

```
# Check if the volume is detected
lsblk

# Format the volume (only needed once)
sudo mkfs -t xfs /dev/nvme1n1  # Replace with your device name

# Mount the volume
sudo mkdir /data
sudo mount /dev/nvme1n1 /data
```

Now your volume is accessible at `/mnt/data`

You can even set it to mount automatically on reboot by editing `/etc/fstab` (I

EBS vs EC3

Feature	EBS (Elastic Block Store)	S3 (Simple Storage Service)
Think of it as...	A **hard drive** attached to your cloud PC	A **locker** in the cloud for storing files
Tied to EC2?	✅ Yes — it's like your EC2 instance's disk	❌ No — it's independent, global access
File type	Stores files like a computer's disk (block storage)	Stores **objects** like images, CSVs, JSONs (object storage)
Access	Only from the EC2 it’s attached to	From anywhere — even websites, mobile apps, Python scripts
Mountable?	✅ Yes (like `D:` or `/mnt/data`)	❌ No — you don't “mount” S3, you access via API or URL
Use for	OS, databases, software, training data for ML	Backup files, model outputs, datasets, static websites
Data persists?	✅ Yes — even after EC2 is stopped (if configured)	✅ Yes — files stay until deleted manually
Cost per GB	💰 Higher	💰 Lower

Feature	EBS (Elastic Block Store)	S3 (Simple Storage Service)
Example file	Jupyter notebook, ML dataset on EC2	`data.csv`, `model.pt`, `report.pdf`, `photo.jpg`

📦 EBS = Your Laptop's SSD or Hard Disk

- It contains your operating system, software, and local files.
- When you shut down your laptop, your disk stays with it.
- You can't pull it out and plug it into another machine (unless you know what you're doing).

☁️ S3 = A Google Drive or Dropbox

- Upload any file.
- Access it from **any device**.
- Share it with others.
- Doesn't depend on any one machine (like EC2).

Key traits:

- **Accessible from anywhere** (web, apps, EC2).
- **Dirt cheap** ($0.023/GB/month).
- **Immutable**: Files can’t be edited (only replaced).

💾 Use Cases — EBS vs S3

Task	Use EBS?	Use S3?
Install Ubuntu + Apache + Python	✅ Yes	❌ No
Host a static website (HTML files)	❌ No	✅ Yes
Store training datasets for ML	✅ Yes (if running on EC2)	✅ Yes (if accessing via Python or SageMaker)

Task	Use EBS?	Use S3?
Store logs and backups	❌ Not ideal	✅ Perfect use case
Share a PDF or image via public link	❌ Not possible	✅ Use `s3.Object().get()` or public URL
Store large media files	❌ Costly	✅ Cheaper and optimized for it
Attach a drive to EC2	✅ Yes	❌ Not possible (it's not block storage)

🧪 Example: Store a CSV File

Using EBS:

- You SSH into EC2
- Save it like: `/home/ec2-user/data.csv`
- Only EC2 can access it

Using S3:

- You upload to a bucket: `my-bucket/data.csv`
- You can access it:
  - From EC2
  - From a Streamlit app
  - From a local Python script using `boto3`
  - Even from a mobile app

💰 Cost Difference (Very Important)

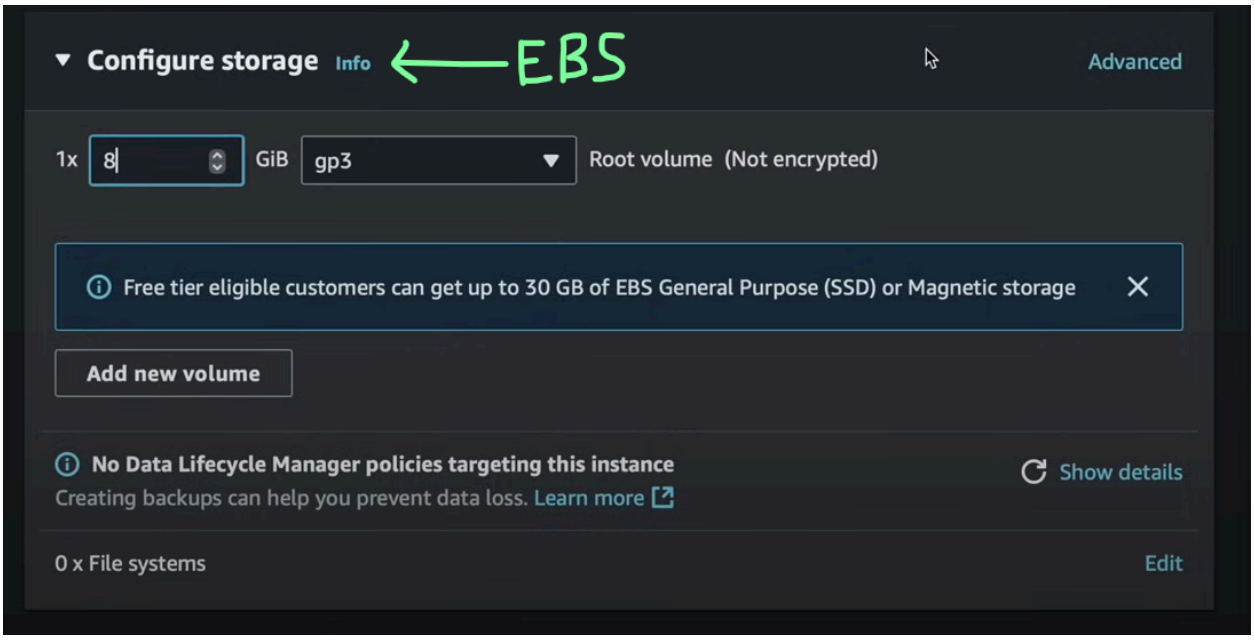
Feature	EBS	S3
Charged while idle?	✅ Yes — per GB even when EC2 is stopped	✅ Yes — but **cheaper per GB**
Minimum cost	Higher (e.g. ₹8–₹10 per GB/month)	Lower (e.g. ₹2–₹4 per GB/month)
Charges for access?	❌ No	✅ Only when you download large amounts (GET/PUT/transfer fees)

📌 Summary — When to Use What?

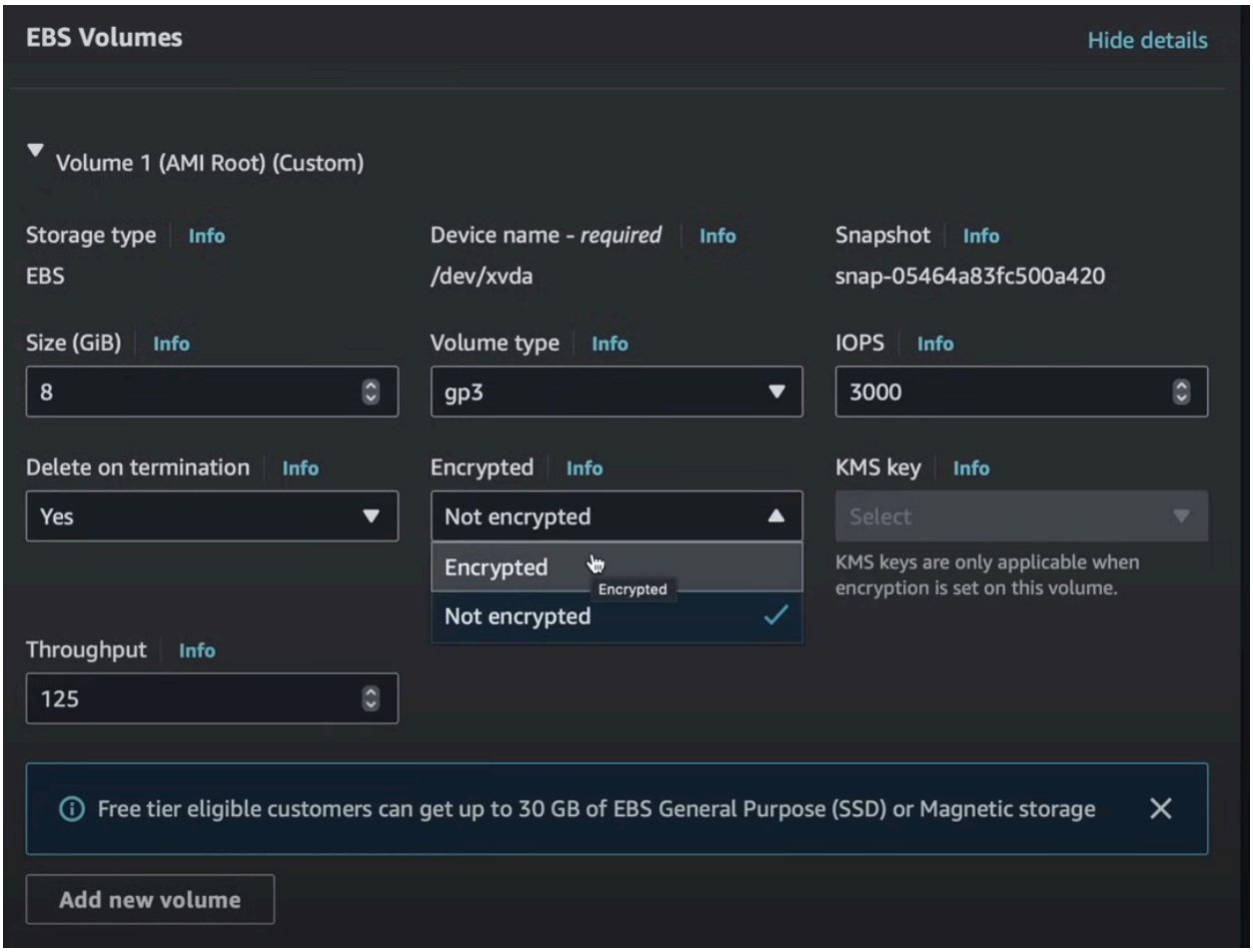
If you want to...	Use...
Store files temporarily or work with OS or installed packages	**EBS**
Save model weights, datasets, or user uploads in a central place	**S3**
Run ML training inside EC2 and save intermediate files	**EBS or attached volume**
Share files between apps, EC2s, or people	**S3**
Backup your EC2 volume	**EBS Snapshot or S3**
Use Python to upload and fetch files	**S3** (with `boto3`)

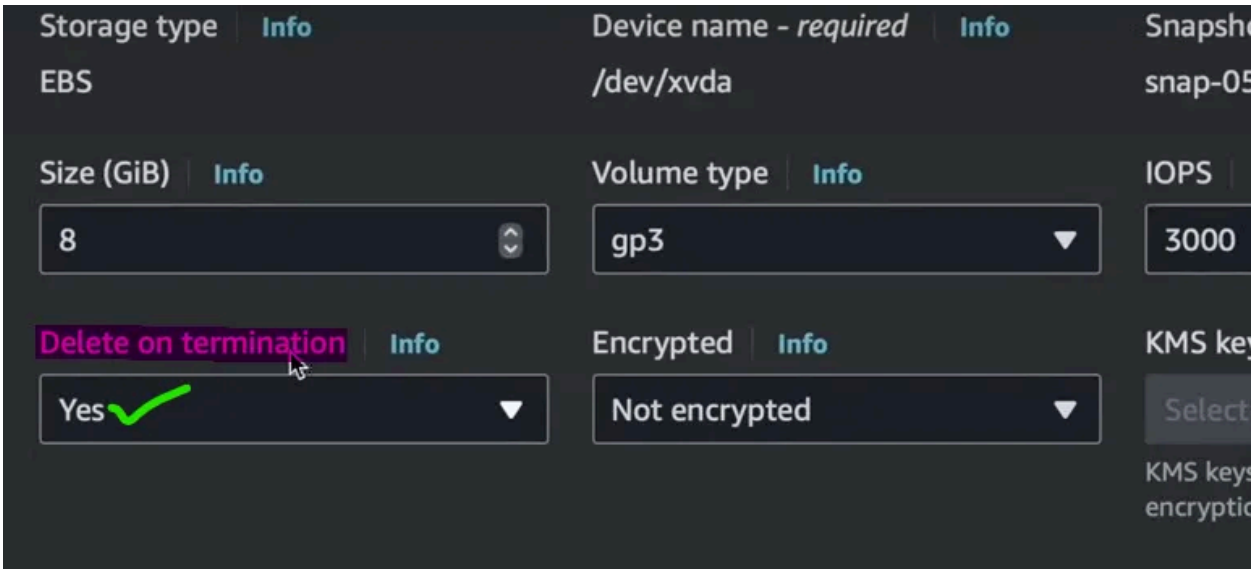
✨Practical Implementation

> 💡 **EBS can be configured while creating an EC2 instance.**



Click Advance 👇



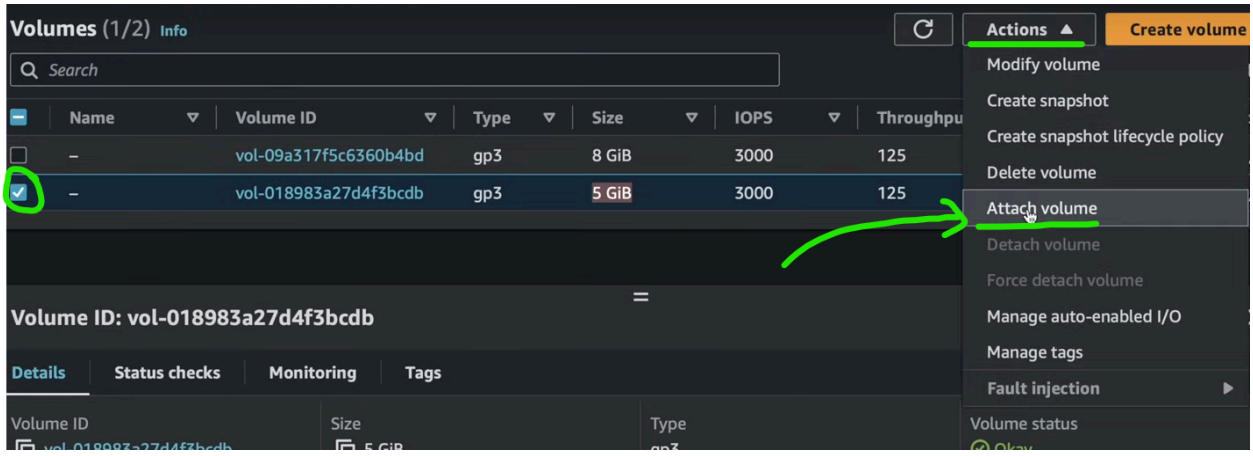


Keep delete on termination→ **Yes**

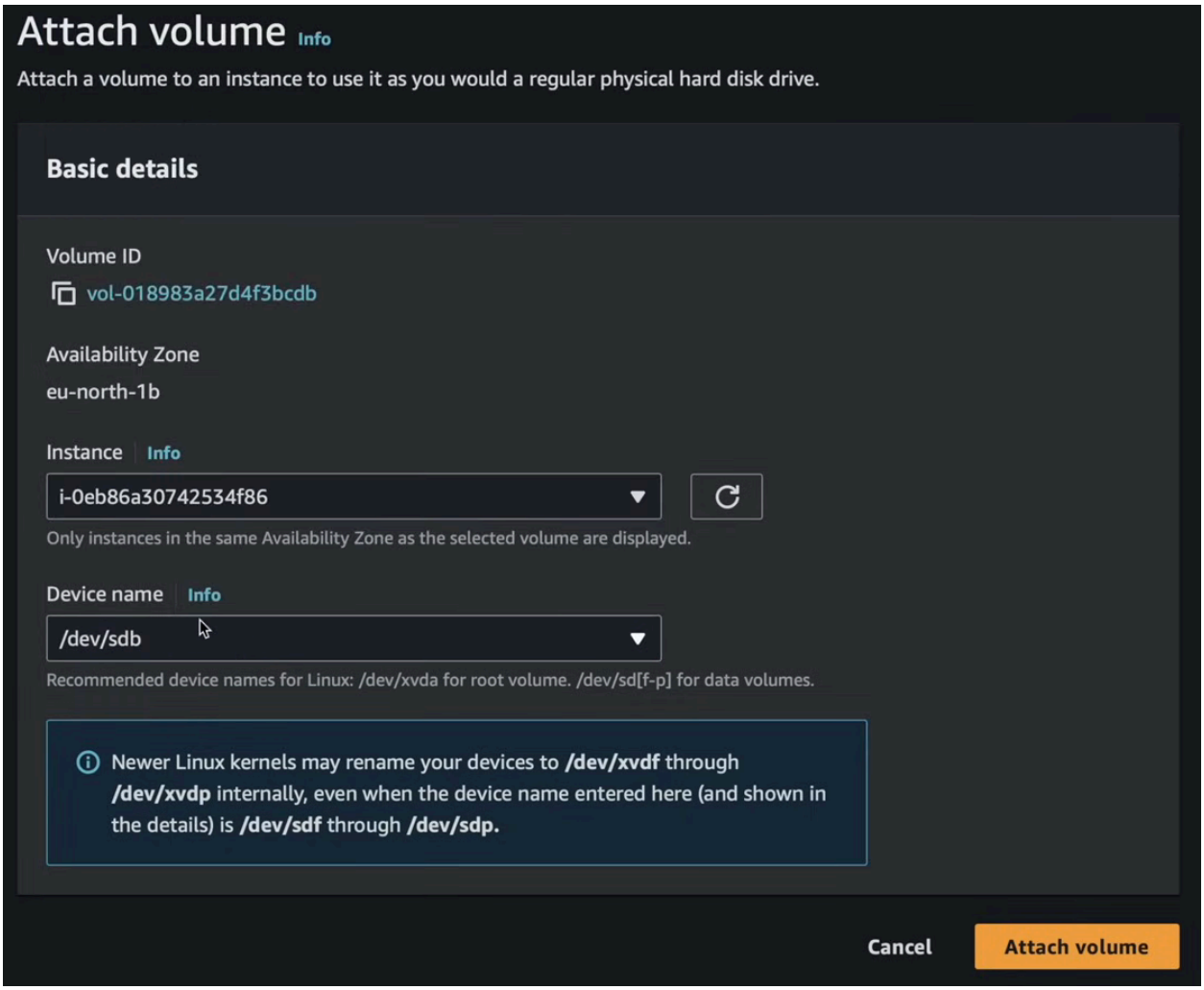
- 👆This will delete the Volume if you delete the EC2 instance.

Attach volume to an EC2 instance

1. Select the Volume
2. Click Actions dropdown → Attach Volume



3. Select the Instance & give Device Name



> 💡 **Avoid naming conflicts: don't assign `/dev/sda1`, `/dev/xvda` (reserved for root volume)**

What is `/dev/` in Linux?

In Linux, everything — even hardware — is treated like a **file**.

All the hardware devices are listed under the `/dev` directory.

Example	Meaning
`/dev/xvda`	Virtual hard disk A

Example	Meaning
`/dev/xvdf`	Virtual hard disk F
`/dev/null`	A black hole (data sink)
`/dev/sda`	SCSI disk A (common in physical systems)

📦 What Is `/dev/xvdX`?

Name	Meaning
`/dev/`	Device folder in Linux
`xv`	Short for **Xen Virtual** device (older virtualization tech)
`d`	Stands for **disk**
`f`, `g`, `h`, etc.	Just **letters for naming multiple disks**

These are **virtual disk device names** used by AWS EC2 and Xen hypervisor.

Name	Meaning
`xvda`	Root volume (OS disk)
`xvdf`, `xvdg`, etc.	Additional volumes (your attached EBS)

- `xvd` stands for **Xen Virtual Disk**
- The **last letter** (`a`, `b`, `f`, `g`, etc.) represents **disk number**

So:

- `/dev/xvda` → first disk (usually the root volume)
- `/dev/xvdf` → second disk
- `/dev/xvdg` → third disk
- and so on...

✅ These are **not partitions**, but **entire volumes/disks**.

🧩 Why Does EC2 Say `/dev/sdf` But You See `/dev/xvdf`?

> In EC2, if you attach a volume and type /dev/sdf in the console,

> the Linux VM will see it as:
>
> ✅ `/dev/xvdf`

That's because:

- **EC2 automatically renames** `/dev/sdX` → `/dev/xvdX`
- It's totally normal — and safe

✅ Which Device Names Can I Use?

You can choose **any of these device names** when attaching EBS volumes:

```
/dev/sdf  → Linux sees as /dev/xvdf
/dev/sdg  → /dev/xvdg
/dev/sdh  → /dev/xvdh
...
```

Avoid:

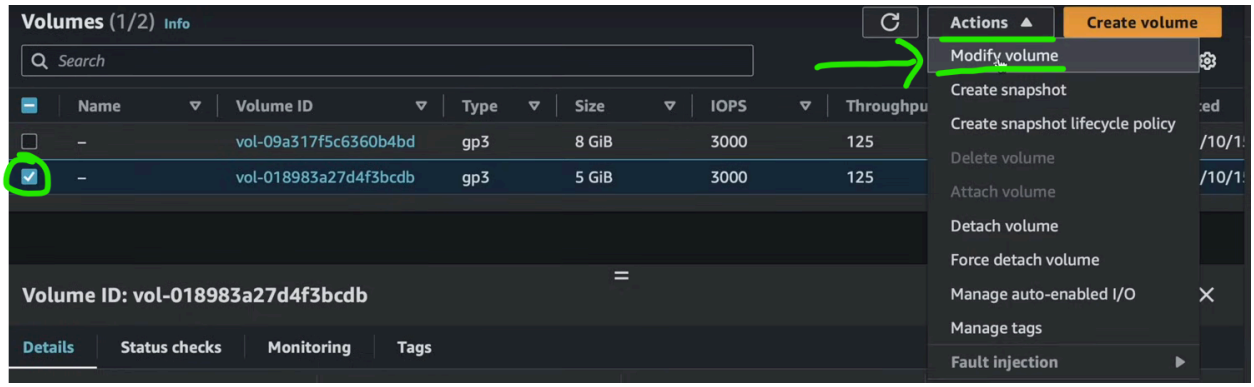
- `/dev/xvda` → reserved for root OS
- Duplicates (you cannot attach two volumes to `/dev/xvdf`)

💡 **Always increment the last letter to attach multiple volumes safely:**

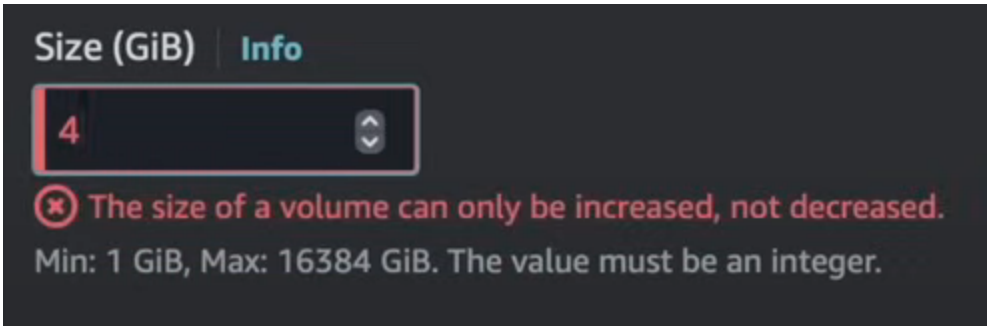
```
xvdf, xvdg, xvdh, xvdi...
```

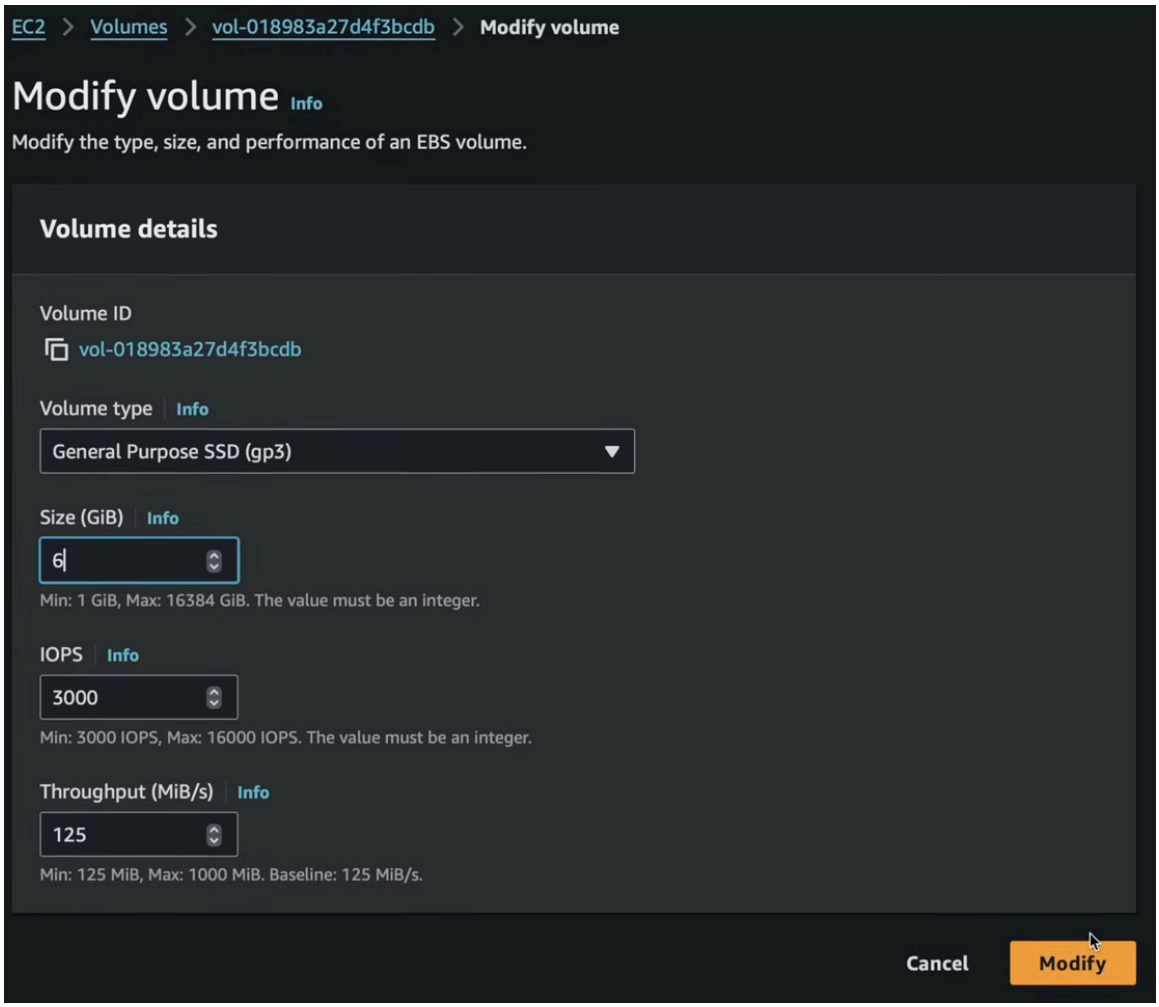
Modify Volume

- Select → Actions → Modify Volume



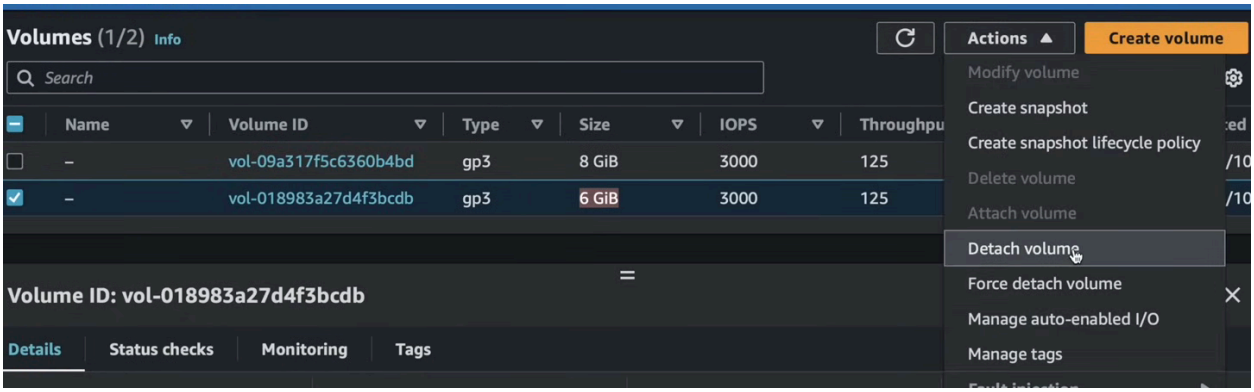
💡 **Size can only be increased.**





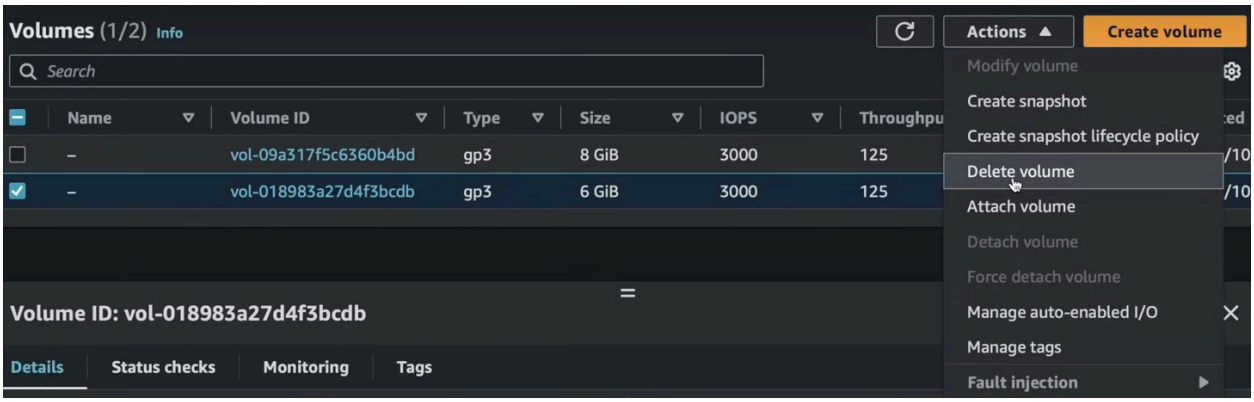
Detach & Delete

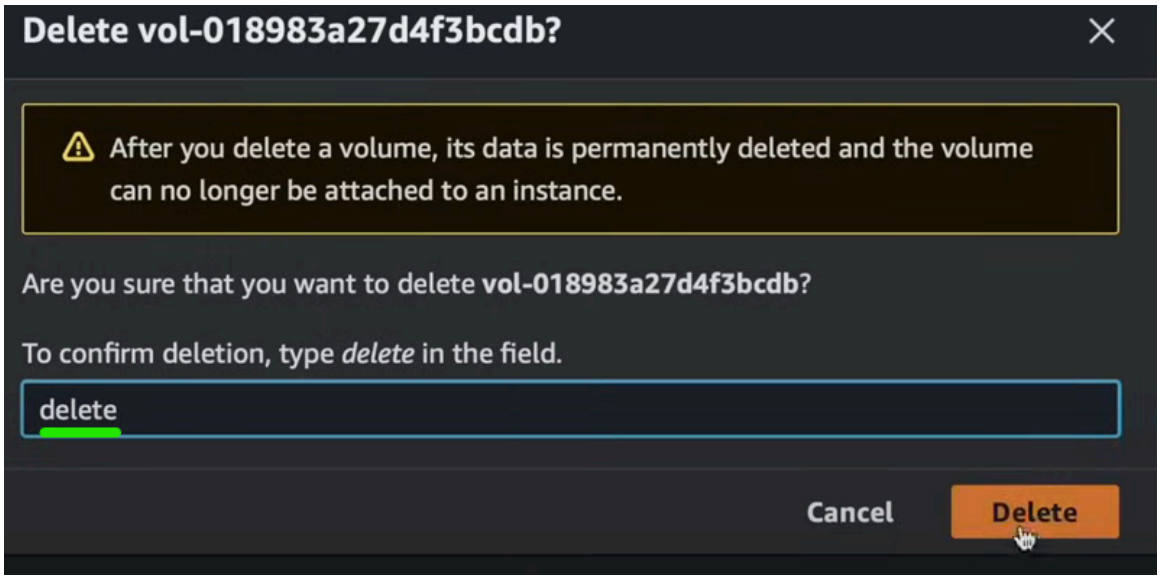
- Select → Actions → Detach volume



- Now the delete volume button will be available

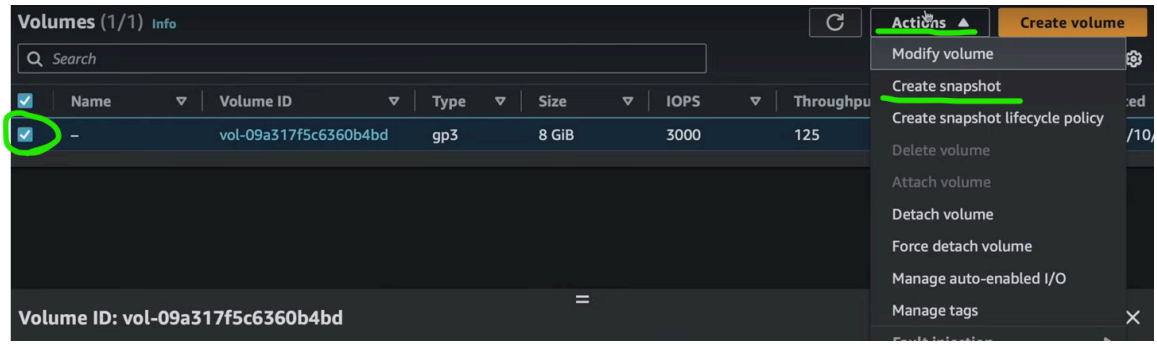
- Select → Actions → Delete volume

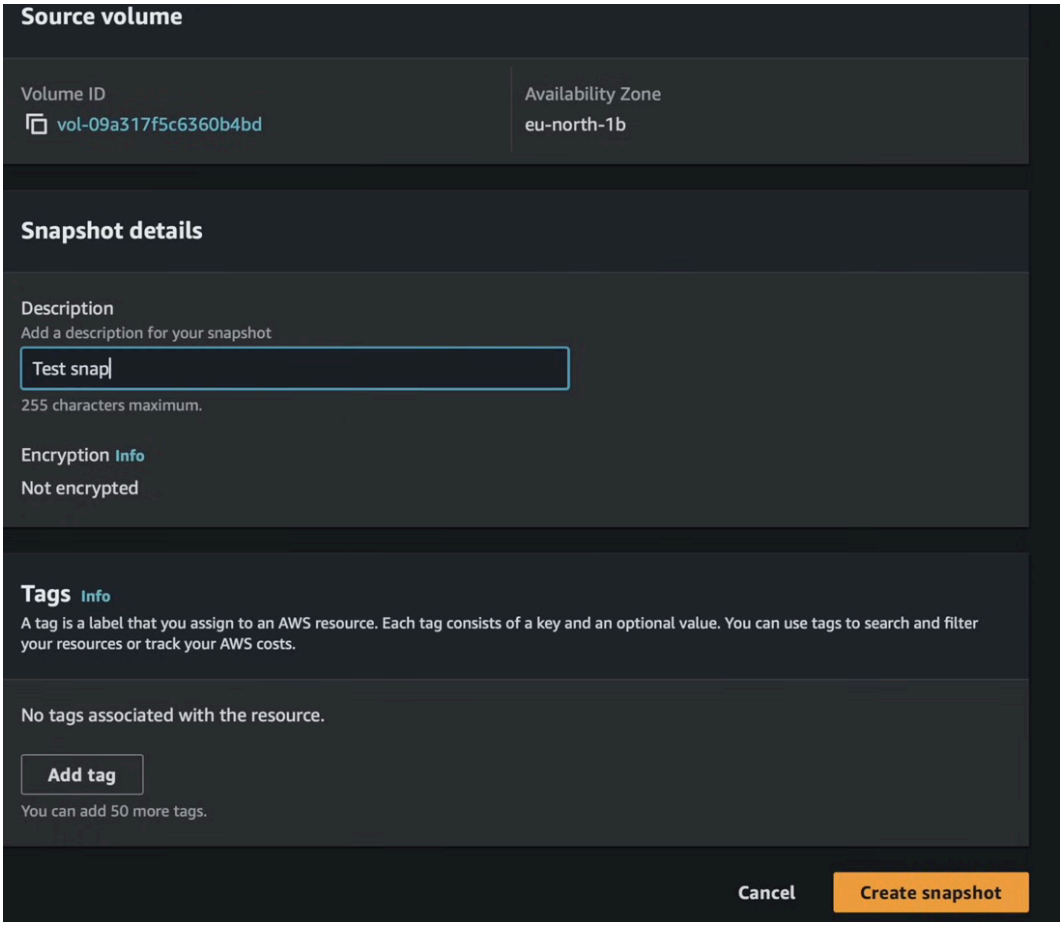




📸EBS Snapshot

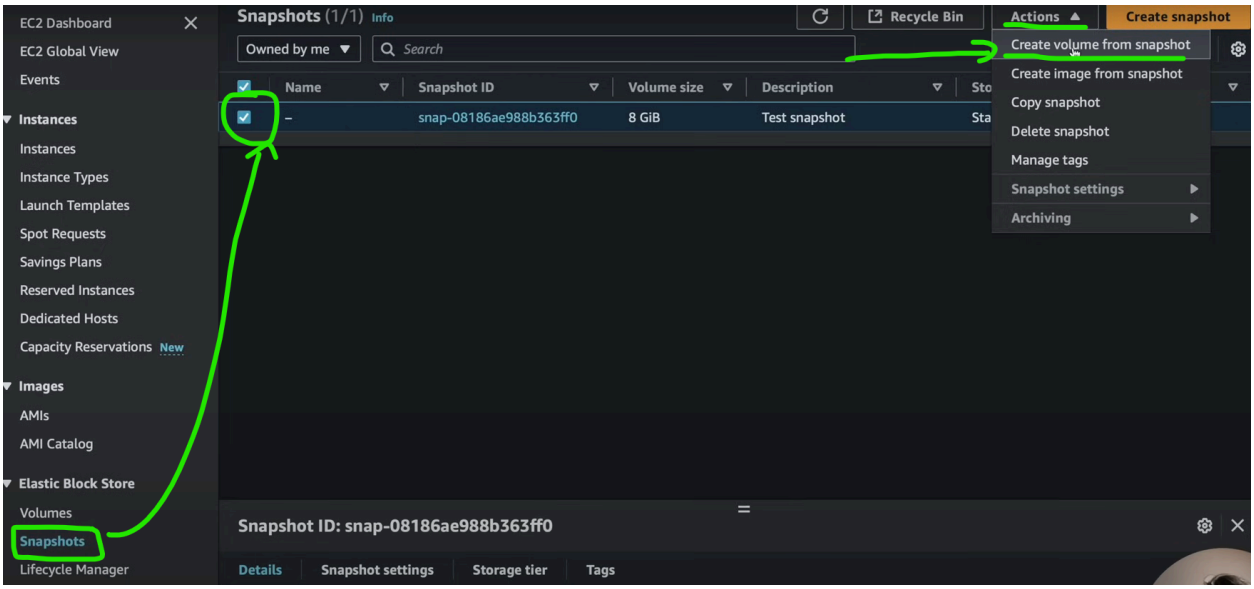
- Select → Actions → Create snapshot

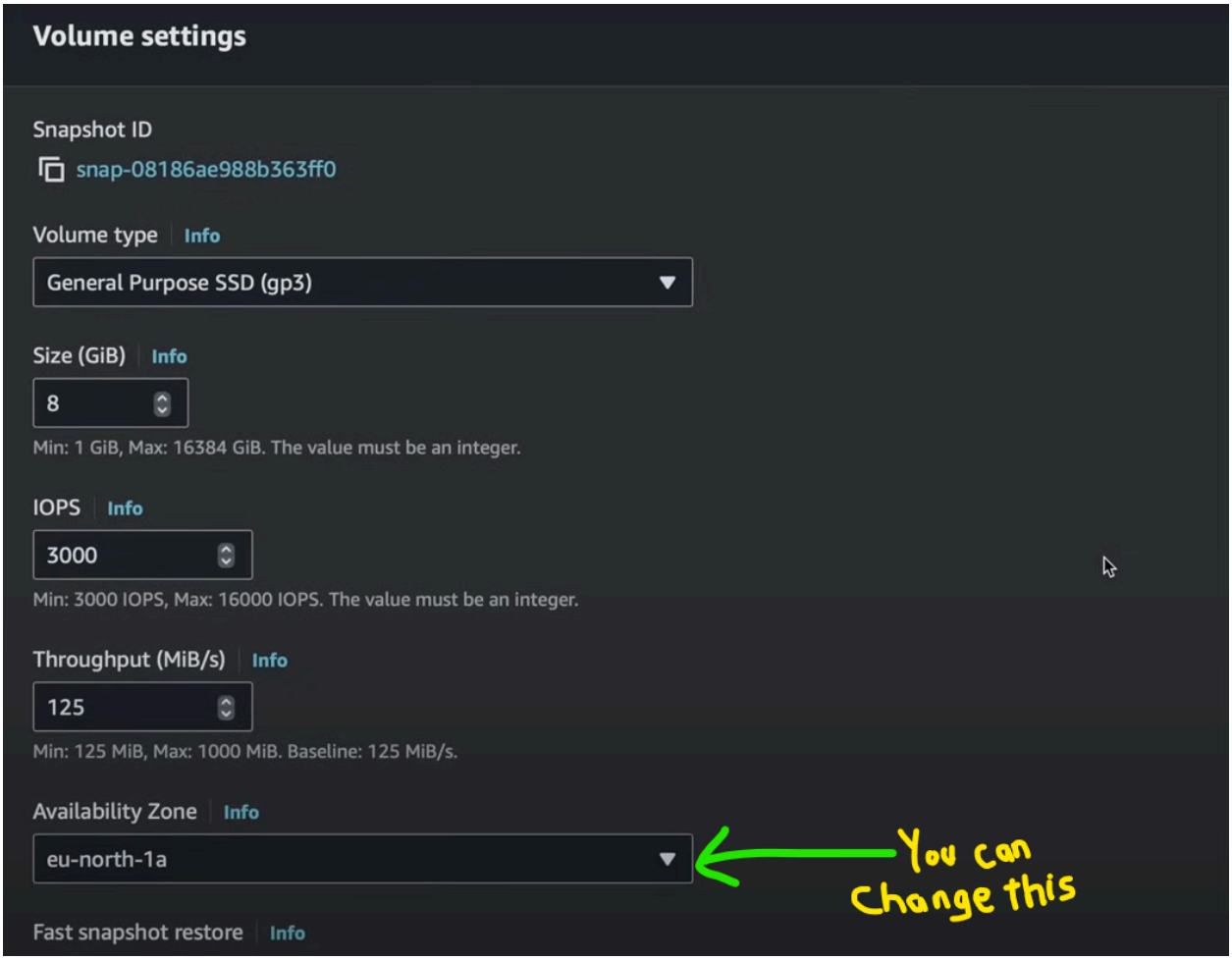


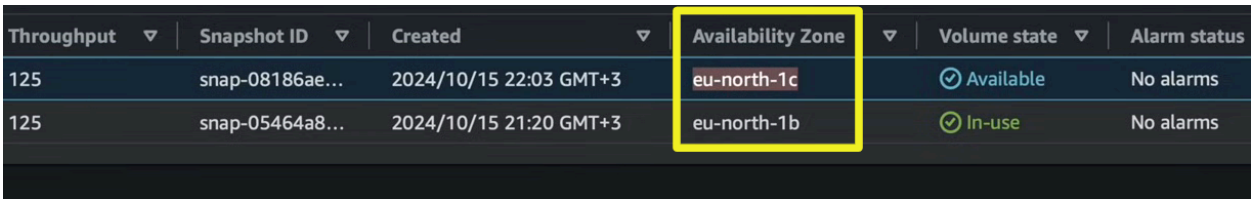


Create Volume from snapshot:

- Go to snapshot
- Select snapshot→ Actions→ Create volume from snapshot


EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Snapshots (1/1) Info
Recycle Bin
Actions
Create snapshot
Owned by me
Search
Name
Snapshot ID
Volume size
Description
snap-08186ae988b363ff0
8 GiB
Test snapshot
Create volume from snapshot
Create image from snapshot
Copy snapshot
Delete snapshot
Manage tags
Snapshot settings
Archiving
Snapshot ID: snap-08186ae988b363ff0
Details
Snapshot settings
Storage tier
Tags

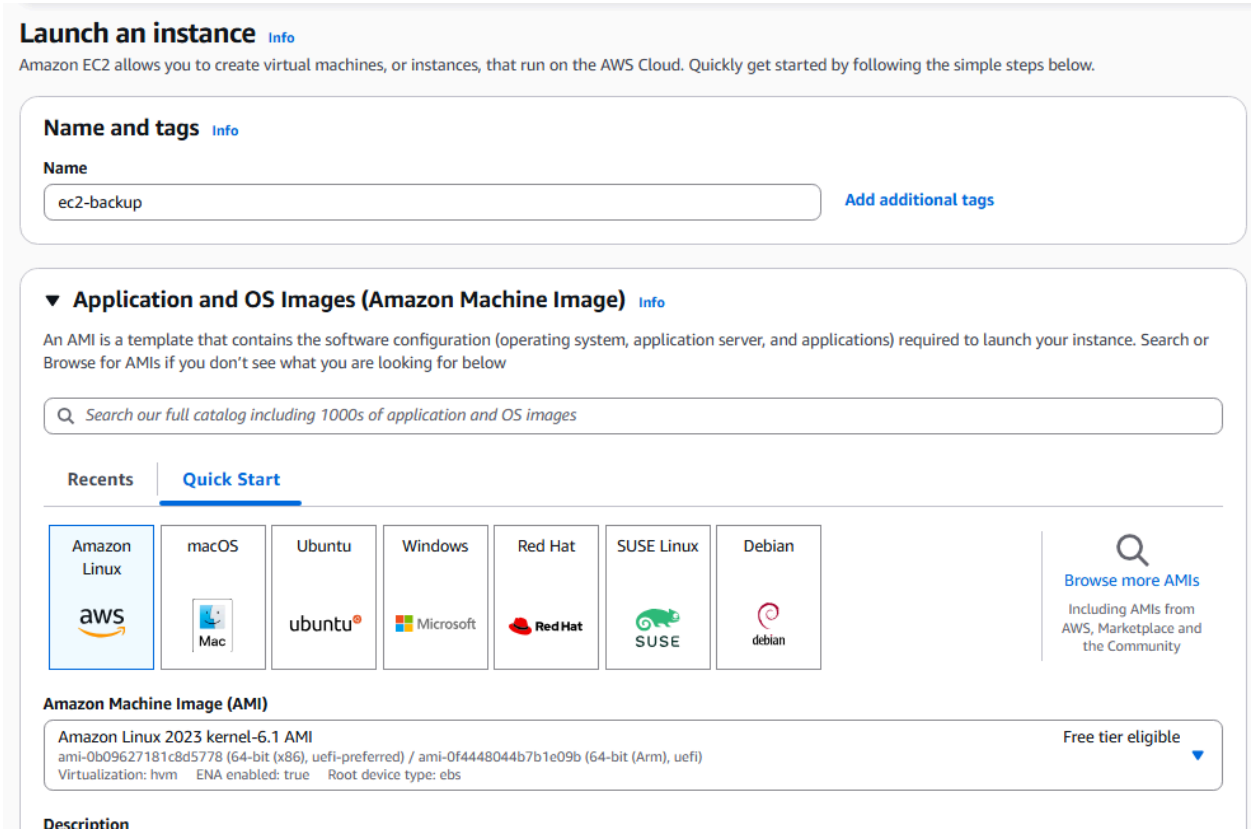

Volume settings
Snapshot ID
snap-08186ae988b363ff0
Volume type Info
General Purpose SSD (gp3)
Size (GiB) Info
8
Min: 1 GiB, Max: 16384 GiB. The value must be an integer.
IOPS Info
3000
Min: 3000 IOPS, Max: 16000 IOPS. The value must be an integer.
Throughput (MiB/s) Info
125
Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.
Availability Zone Info
eu-north-1a
You can Change this
Fast snapshot restore Info



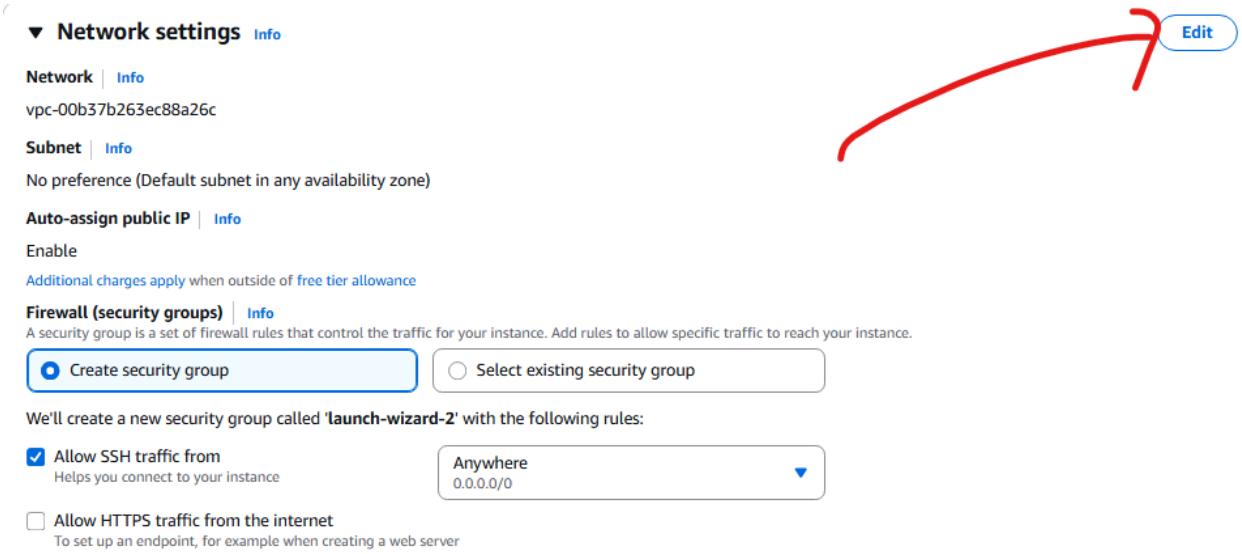
Throughput	Snapshot ID	Created	Availability Zone	Volume state	Alarm status
125	snap-08186ae...	2024/10/15 22:03 GMT+3	eu-north-1c	Available	No alarms
125	snap-05464a8...	2024/10/15 21:20 GMT+3	eu-north-1b	In-use	No alarms

Now attach the newly created Volume in 1c to an EC2 instance in 1c

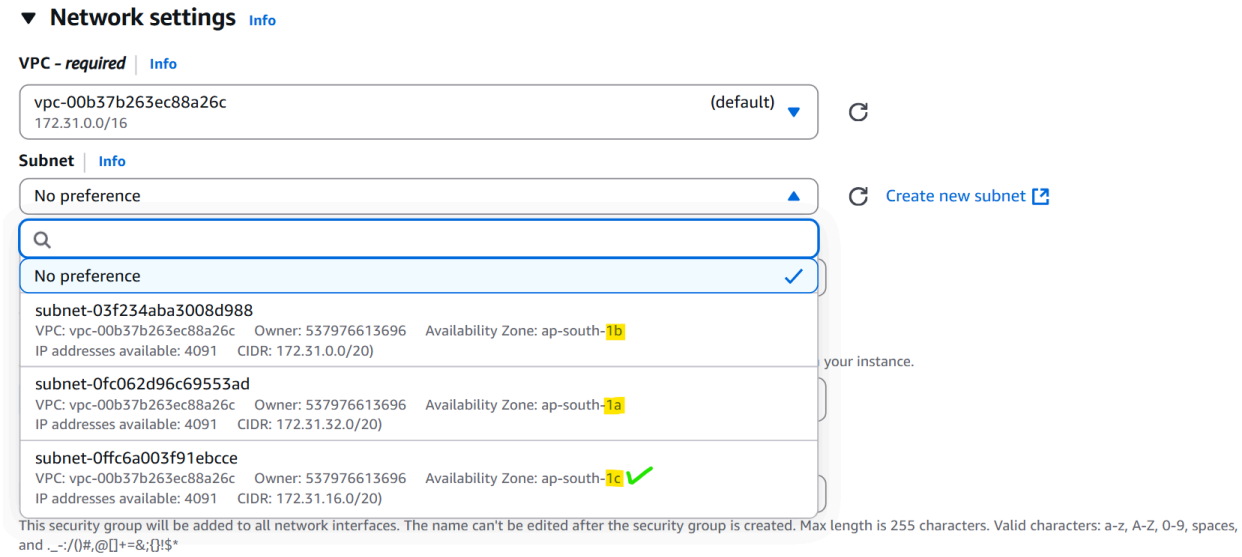
- **Launch Instance**



Edit Network settings to change the zone

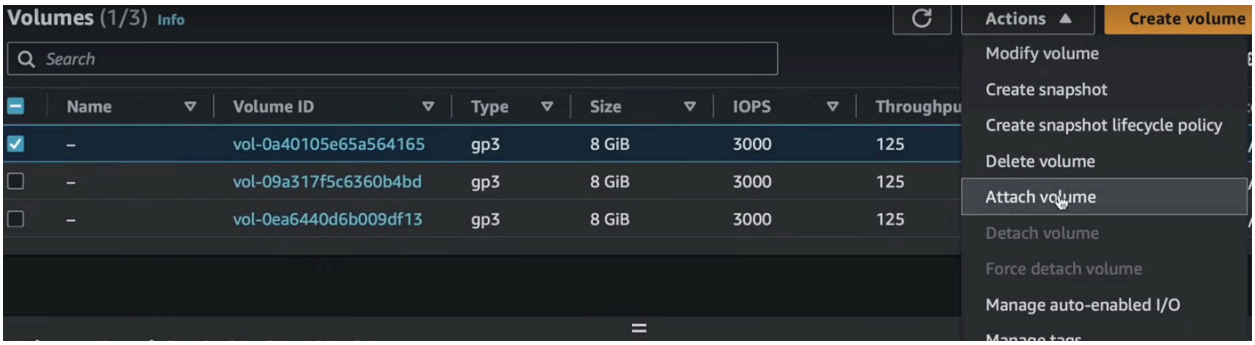


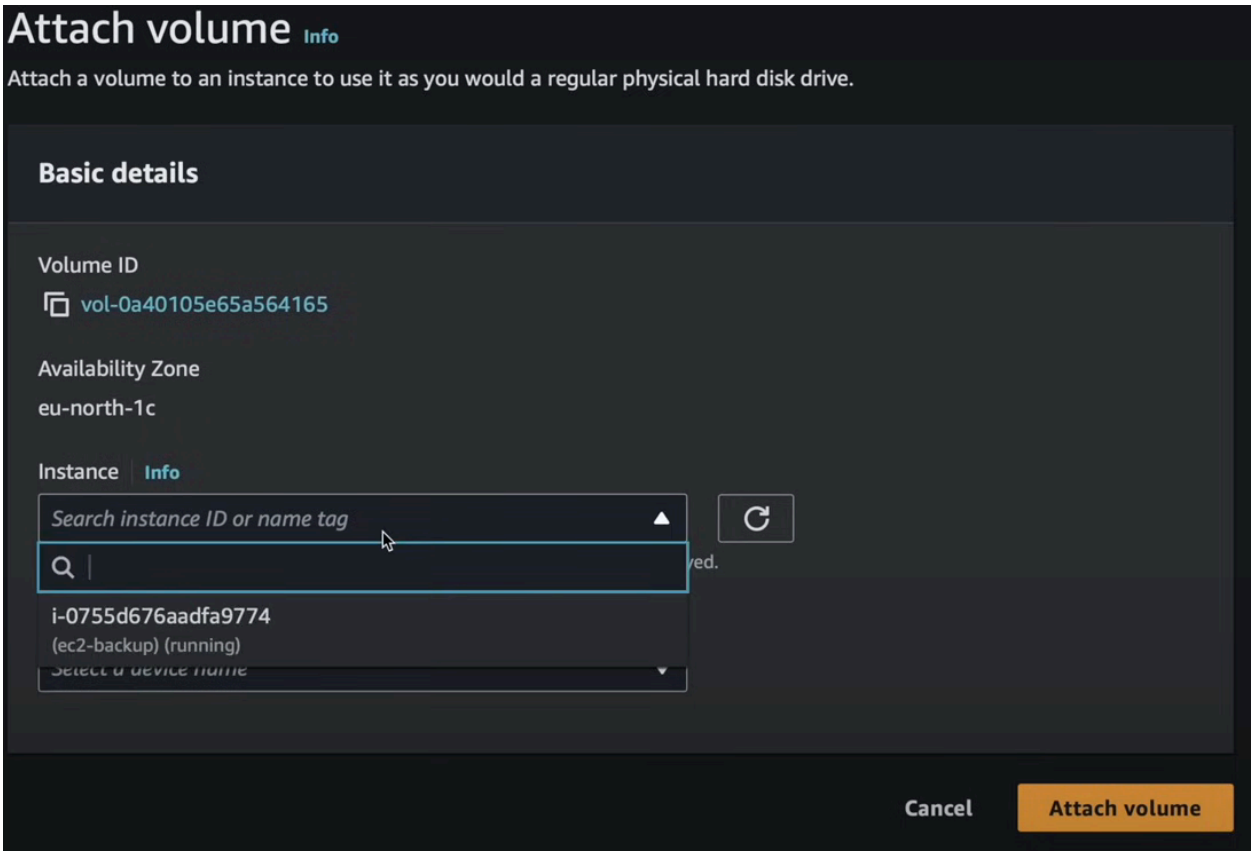
- Select 1c from dropdown in **subnets**



Attach backup to newly created instance

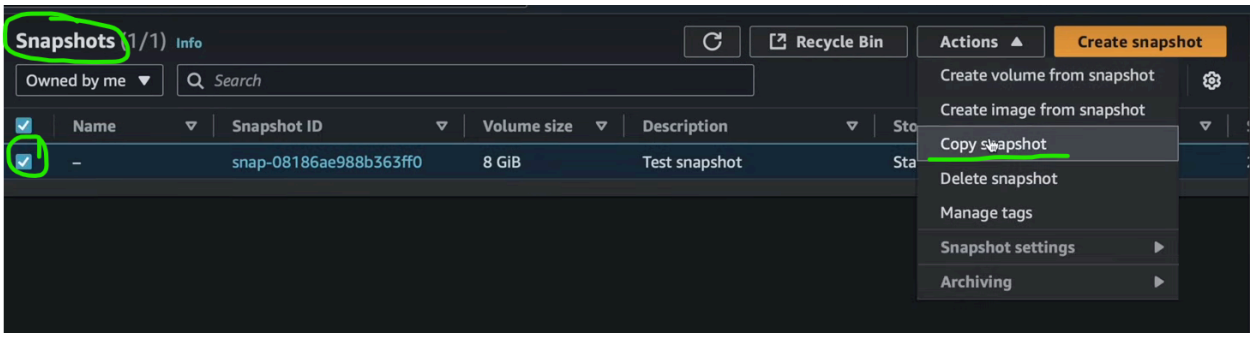
- Select → Actions → Attach volume

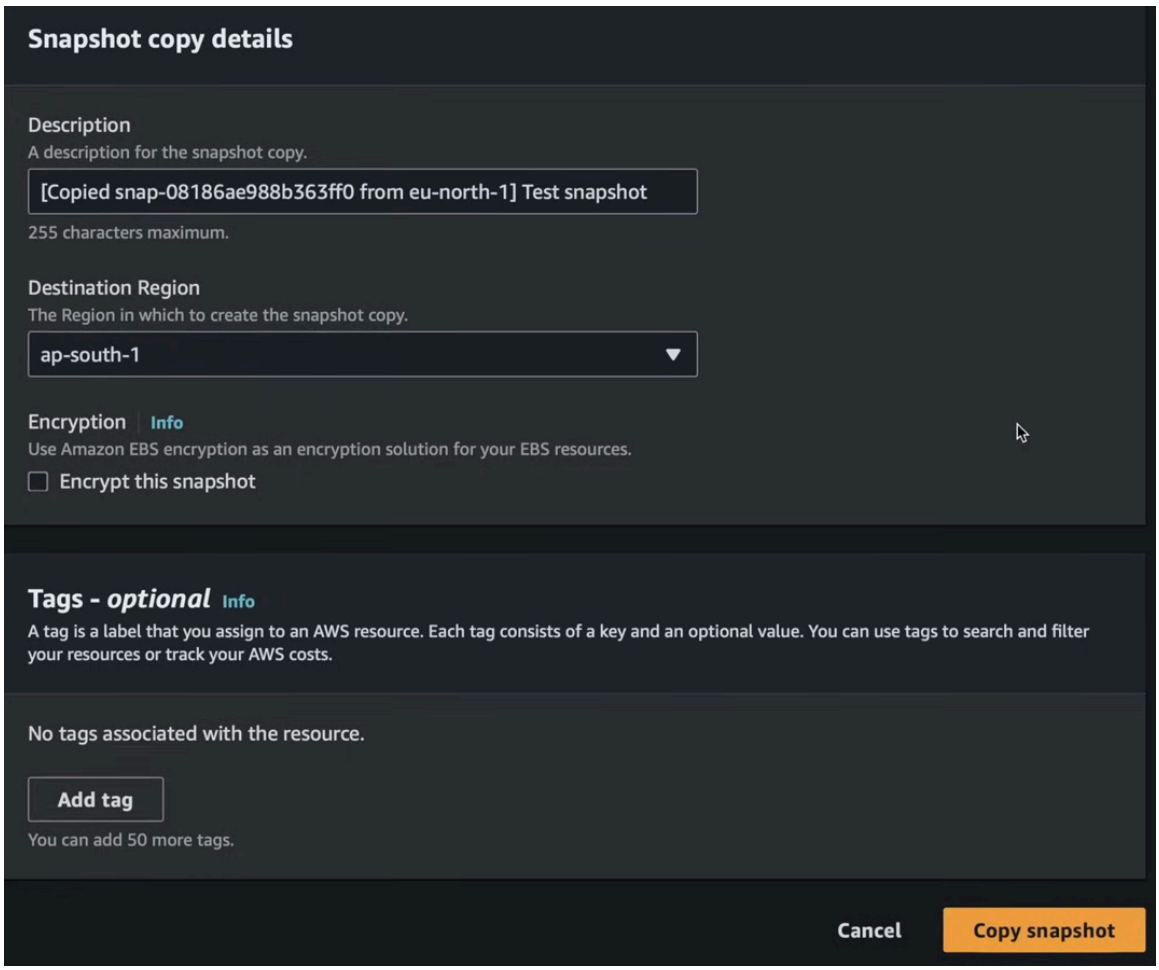




Only **1c** instance will be available.

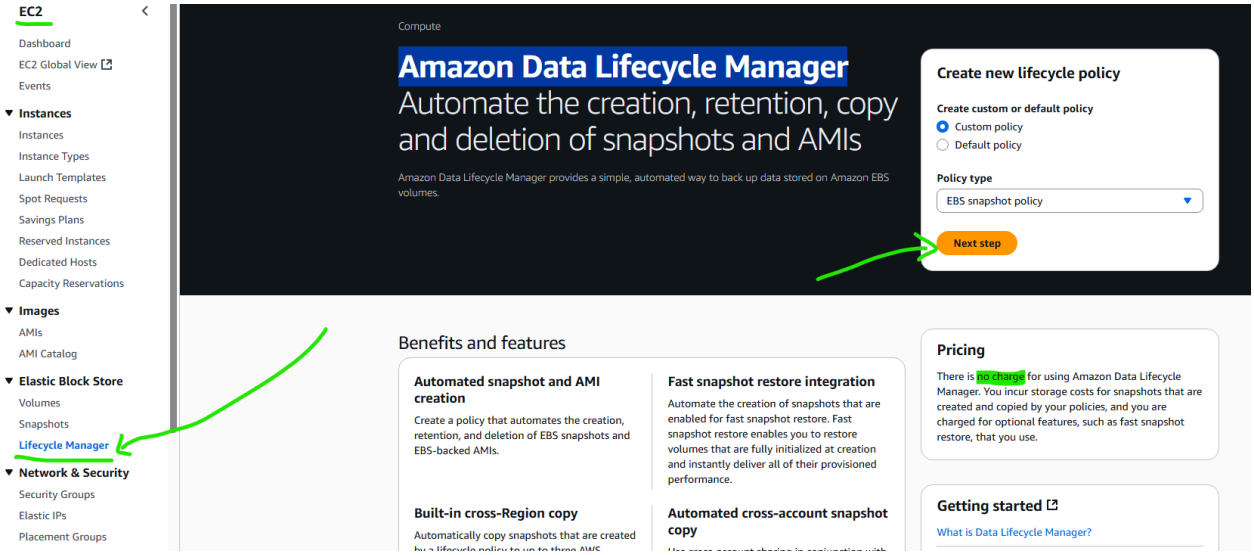
Copy Snapshot to another region



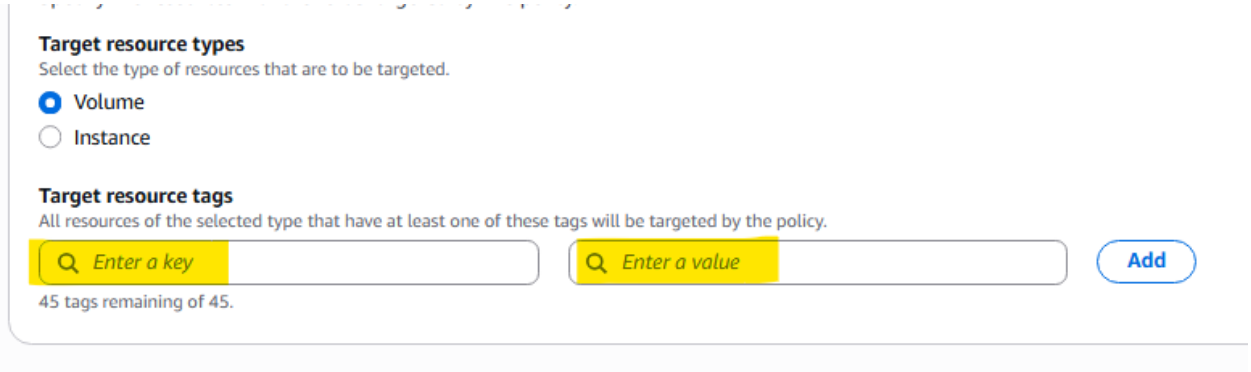


🔁Amazon Data Lifecycle Manager

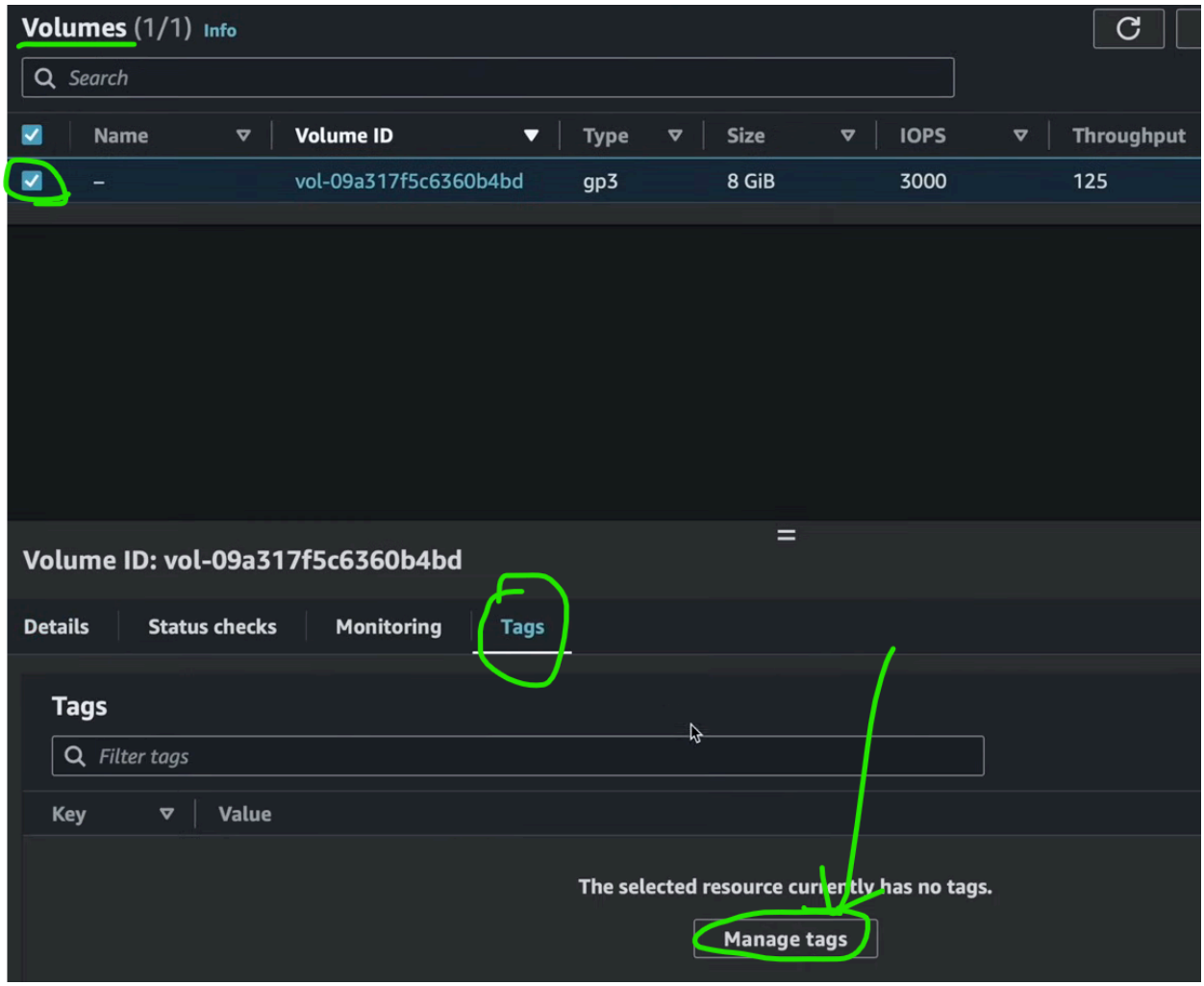
- Useful to take auto-backup

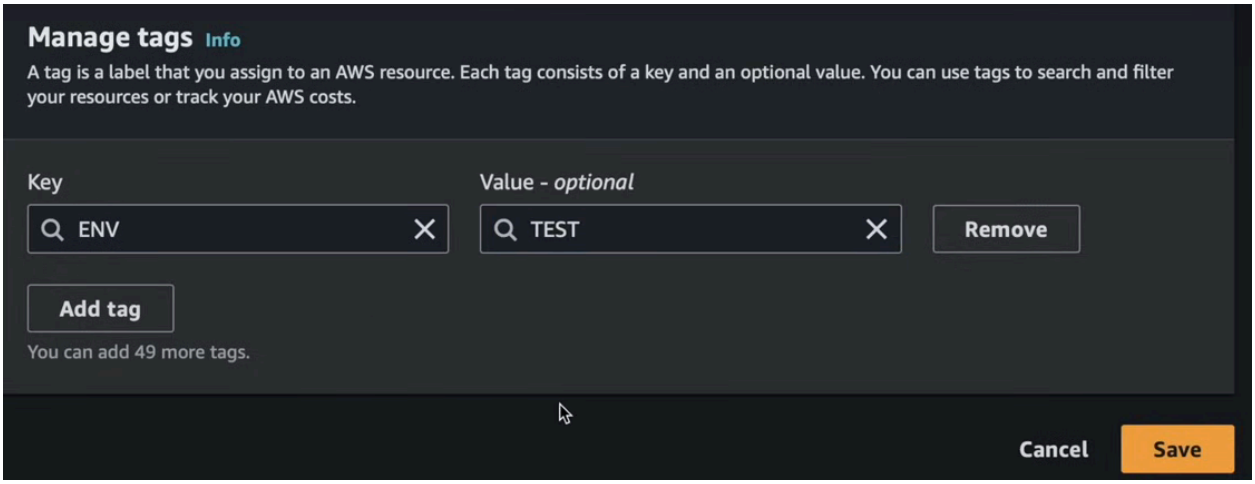


- It’s free

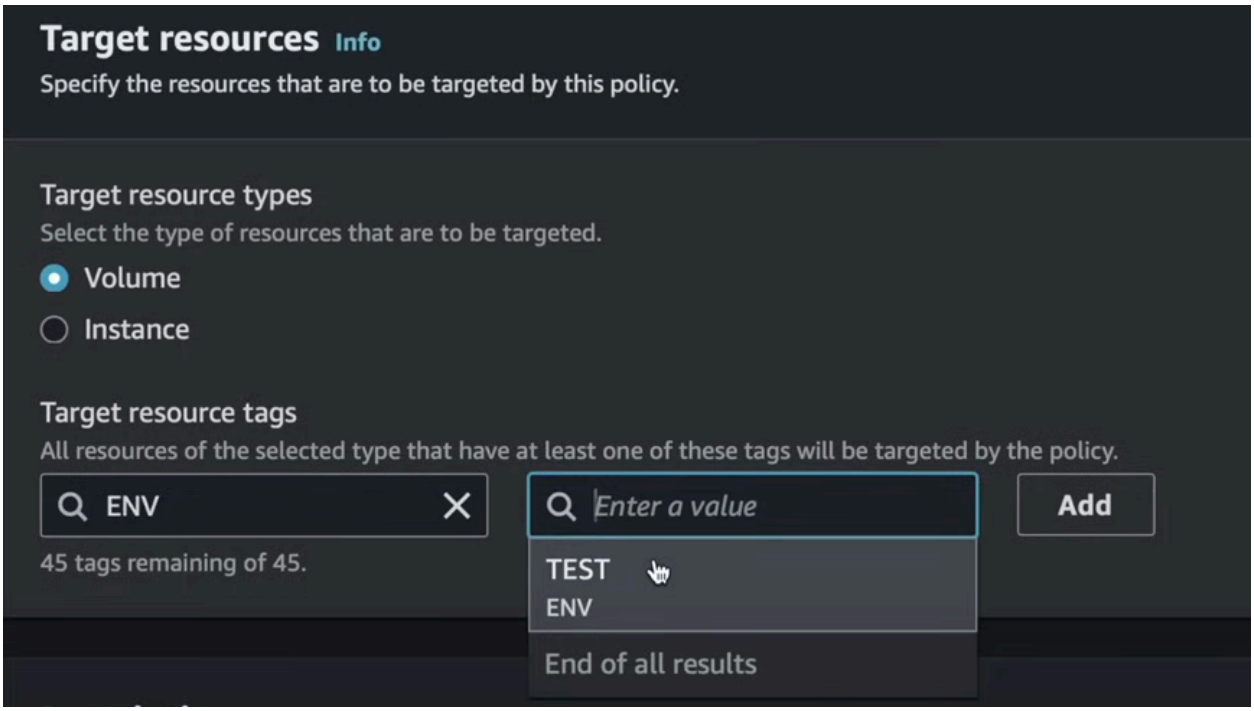


- You can get key & value by adding volume → tags

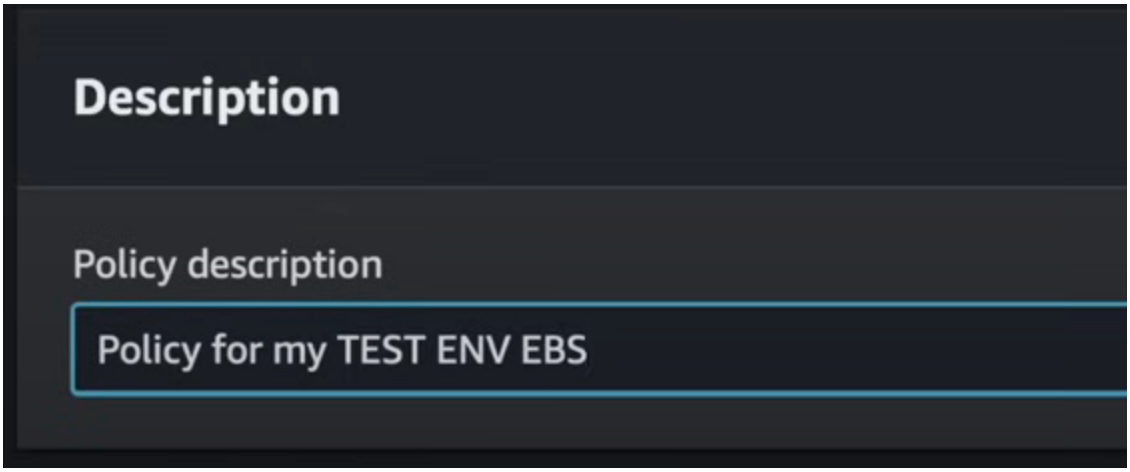




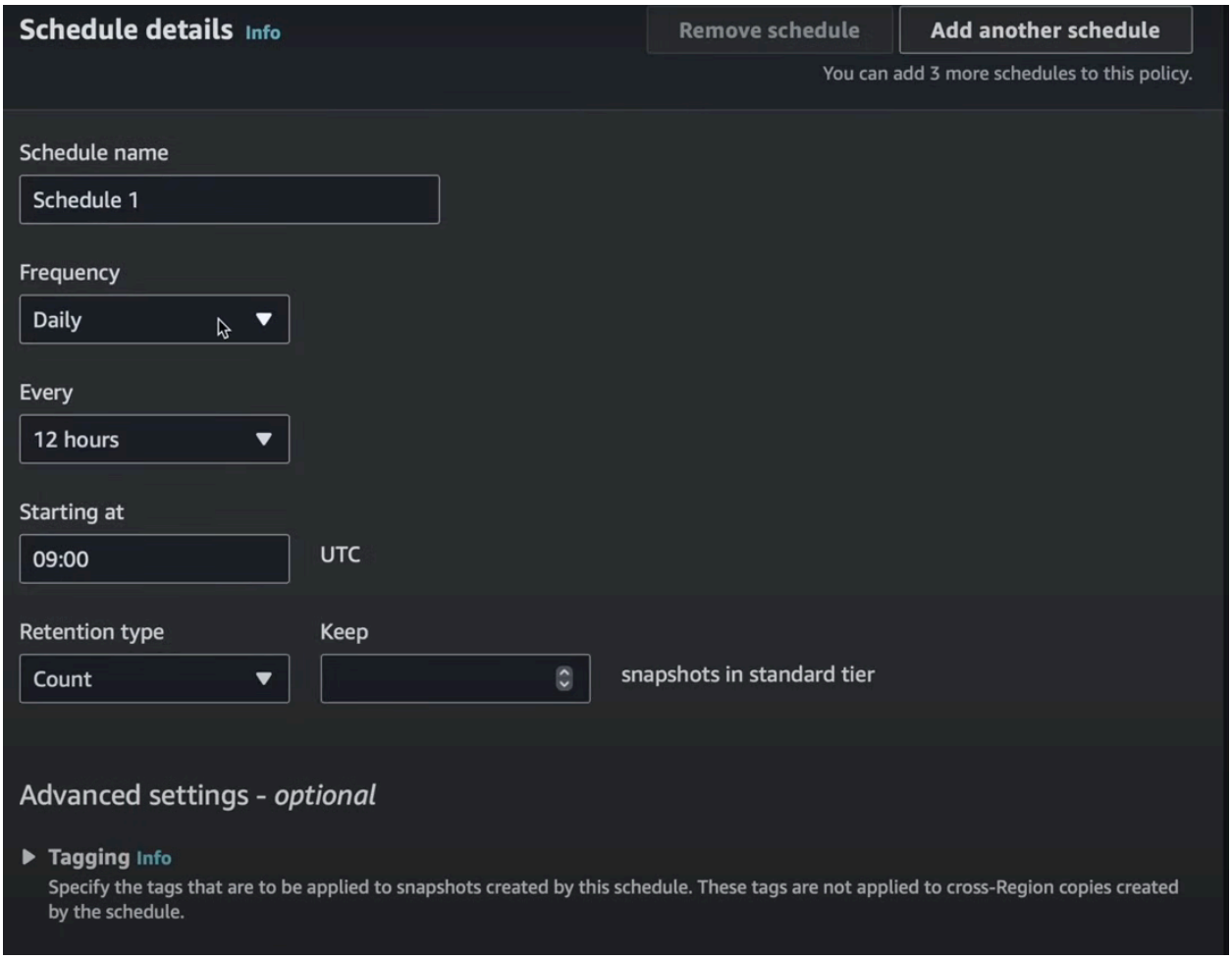
Add the key & value in Life cycle manager



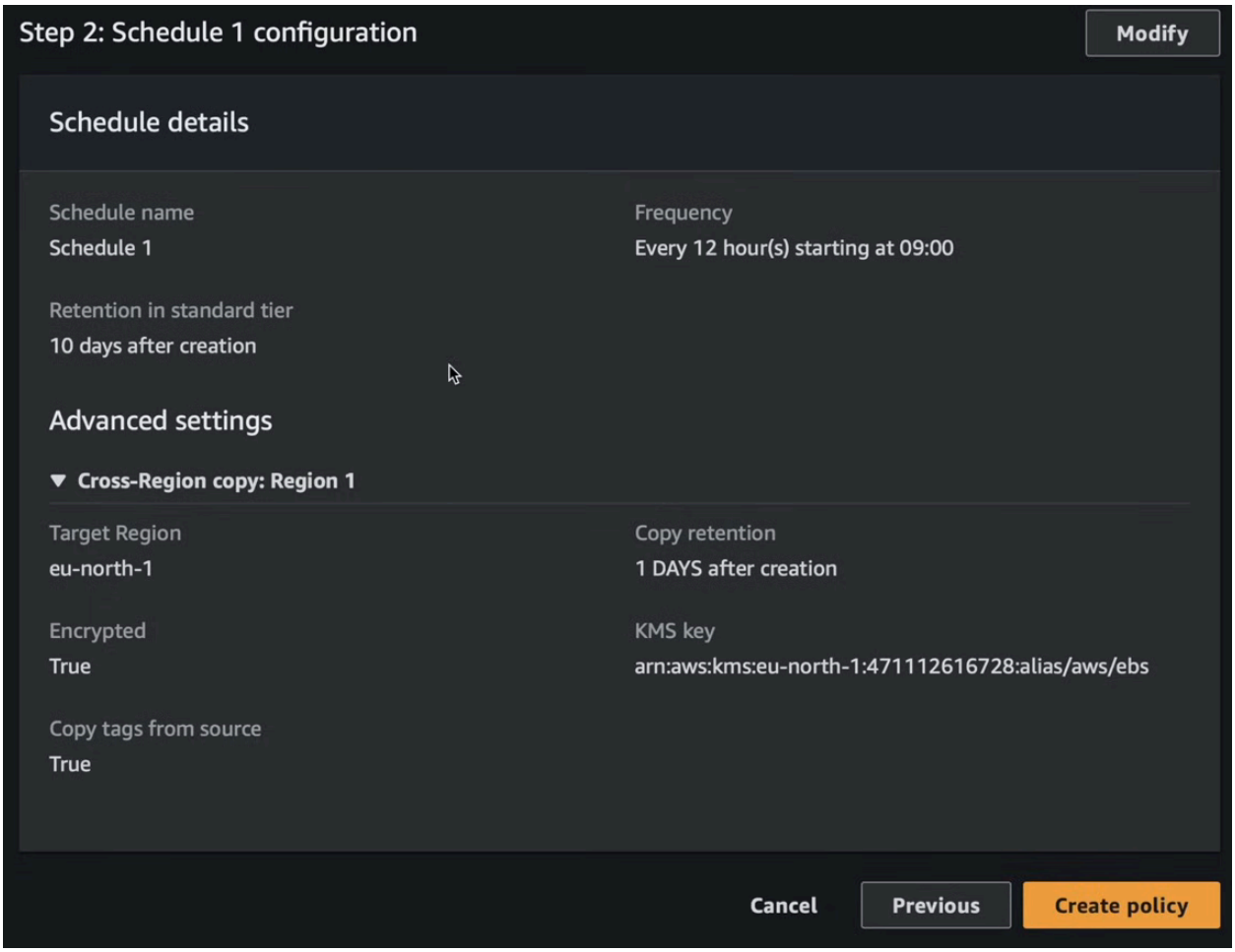
Add description:



Keep everything default → Next


Schedule details Info
Remove schedule
Add another schedule
You can add 3 more schedules to this policy.
Schedule name
Schedule 1
Frequency
Daily
Every
12 hours
Starting at
09:00
UTC
Retention type
Count
Keep
snapshots in standard tier
Advanced settings - optional
Tagging Info
Specify the tags that are to be applied to snapshots created by this schedule. These tags are not applied to cross-Region copies created by the schedule.

- **Next**



Recycle Bin

- Recovers the deleted snapshot