

Experiment – 6

- ❖ **Aim:** To create and understand the Amazon Elastic Block Store (Amazon EBS), a key underlying storage mechanism for Amazon EC2 instances. To create an Amazon EBS volume, attach it to an instance, apply a file system to the volume, and then take a snapshot backup.



Task – 1: Create a New EBS Volume

1. In the AWS Management Console, in the search box next to Services , search for and select EC2.
2. In the left navigation pane, choose Instances.
An Amazon EC2 instance named Lab has already been launched for your lab.
3. Note the Availability Zone of the instance. It will look similar to *us-east-1a*.
4. In the left navigation pane, choose Volumes.

You will see an existing volume that is being used by the Amazon EC2 instance. This volume has a size of 8 GiB, which makes it easy to distinguish from the volume you will create next, which will be 1 GiB in size.

5. Choose **Create volume** then configure:
 - **Volume Type:** *General Purpose SSD (gp2)*
 - **Size (GiB):** 1. **NOTE:** You may be restricted from creating large volumes.
 - **Availability Zone:** Select the same availability zone as your EC2 instance.
 - Choose **Add tag**
 - In the Tag Editor, enter:
 - **Key:** Name
 - **Value:** My Volume

Create volume

Create an Amazon EBS volume to attach to any EC2 instance in the same availability zone.

Volume settings

Volume Type: gp2
General Purpose SSD (gp2)

Size (GB): 1
Max IOPS: 100 / 3000
Throughput (Mbps): 100
Availability Zone: us-east-1a
Snapshot ID (optional): Don't create volume from a snapshot
Encryption: None (Amazon EBS encryption at rest is automatically enabled for your EBS volumes associated with your EC2 instances)
Tags - optional: Add tag

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Name	My Volume

6. Choose Create Volume

Your new volume will appear in the list, and will move from the *Creating* state to the *Available* state. You may need to choose **refresh** to see your new volume.

Successfully created volume vol-0cc43cb06bbe792cf

Volumes (3) info

Last updated less than a minute ago

Actions Create volume

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created
vol-04e8b06c1a5a8475f	gp3	9 GiB	3000	125		snap-0704622...	2025/10/06 16:47 GMT+5:...
vol-03118195f2f5ad06a	gp3	8 GiB	3000	125		snap-0704622...	2025/10/06 16:47 GMT+5:...
My Volume	vol-0cc43cb06bbe792cf	gp2	1 GiB	100	-	-	2025/10/06 16:58 GMT+5:...

Task – 2: Attach the Volume to an Instance

In this task you will attach the new EBS volume to the Amazon EC2 instance.

- Select My Volume.
- In the Actions menu, choose Attach volume.

Volumes (1/3) info

Last updated less than a minute ago

Actions Create volume

Name	Volume ID	Type	Size	IOPS	Throughput	Actions
vol-04e8b06c1a5a8475f	gp3	9 GiB	3000	125		Modify volume Create snapshot Create snapshot lifecycle policy Delete volume Attach volume Detach volume Revert detach volume Manage auto-enabled I/O Manage tags Resiliency testing
vol-03118195f2f5ad06a	gp3	8 GiB	3000	125		
My Volume	vol-0cc43cb06bbe792cf	gp2	1 GiB	100	-	

Volume ID: vol-0cc43cb06bbe792cf (My Volume)

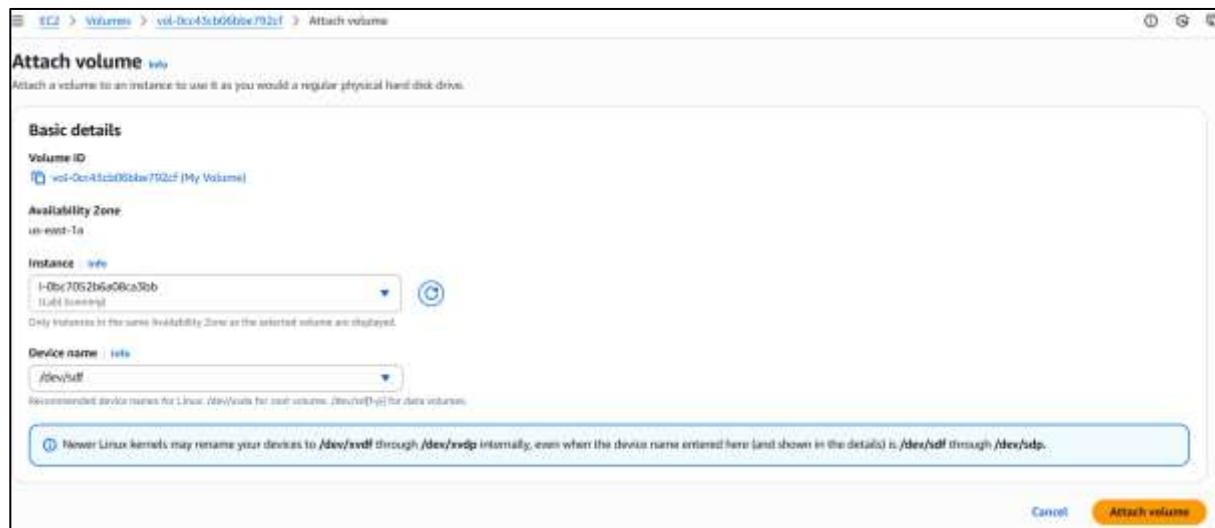
Actions: Details, Status checks, Monitoring, Tags

9. Choose the Instance field, then select the Lab instance.

Note that the Device name is set to `/dev/sdf`. Notice also the message displayed that "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`."

10. Choose Attach volume

The volume state is now *In-use*.

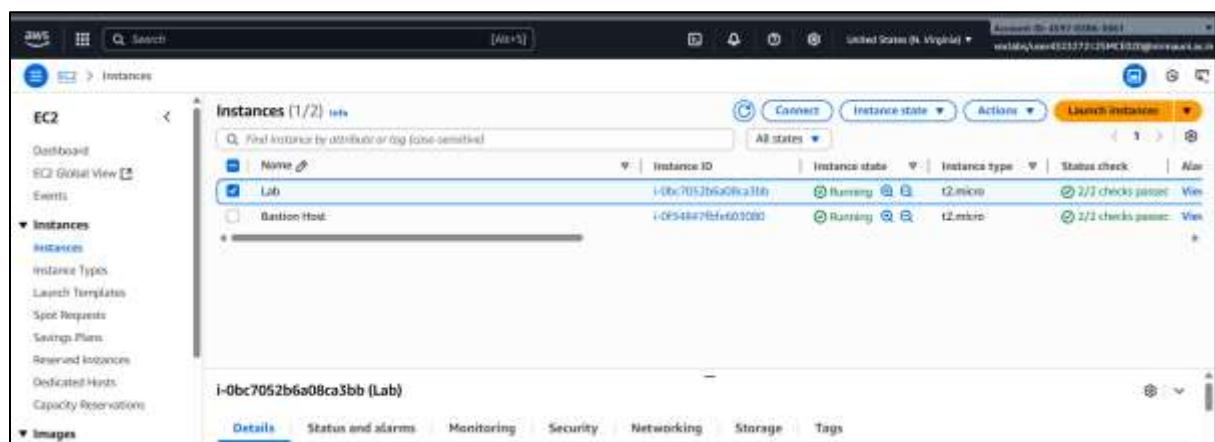


Task – 3: Connect to Your Amazon EC2 Instance

In this task, you will connect to the EC2 instance using EC2 Instance Connect which provides access to a terminal in the browser.

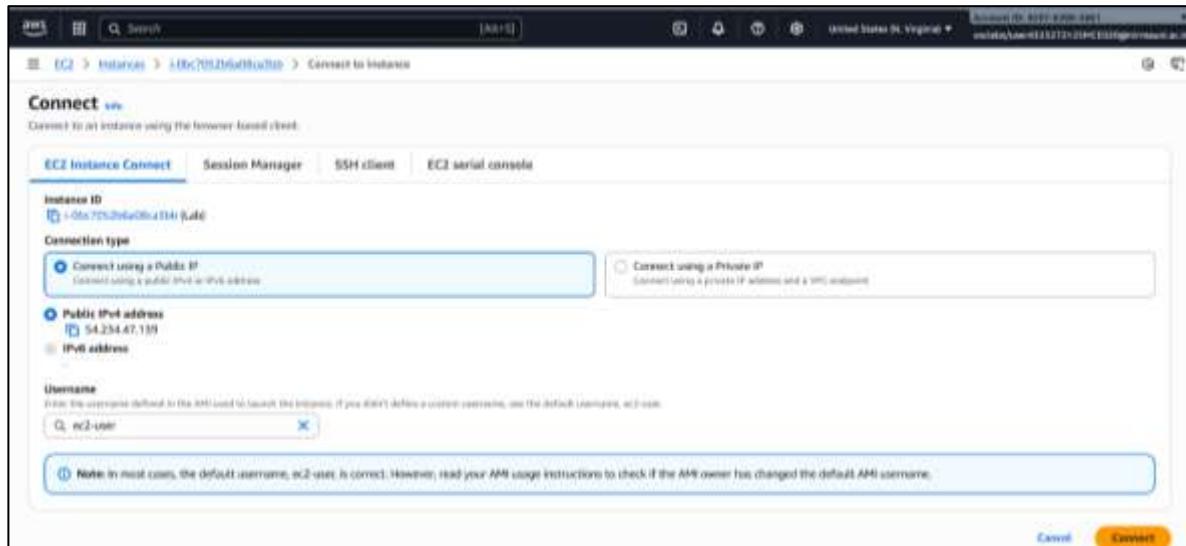
11. In the AWS Management Console, in the search box next to Services , search for and select EC2.

12. Choose Instances.



13. Select the Lab instance, and then choose Connect.

14. On the EC2 Instance Connect tab, choose Connect.



An EC2 Instance Connect terminal session opens and displays a \$ prompt.

This screenshot shows a terminal window from the EC2 Instance Connect interface. The title bar of the terminal window reads 'i-0bc7052b6a08ca3bb (Lab)'. The window itself displays a Linux terminal session. The user is connected to an Amazon Linux 2023 instance. The terminal shows the user's path (~/.aws/credentials), the URL for the Amazon Linux 2023 AMI ('https://aws.amazon.com/linux/amazon-linux-2023'), and the command 'ls' which lists the contents of the current directory. At the bottom of the terminal window, there is a status bar with the text 'Public IPs: 54.234.47.139 Private IPs: 10.1.11.124'.

Task – 4: Create and Configure Your File System

In this task, you will add the new volume to a Linux instance as an ext3 file system under the /mnt/data-store mount point.

15. View the storage available on your instance:

Run the following command:

```
df -h
```

The output shows that the original 8GB /dev/xvda1 disk volume mounted at / which indicates that it is the root volume. It hosts the Linux operating system of the EC2 instance.

The 1GB other volume that you attached to the Lab instance is not listed, because you have not yet created a file system on it or mounted the disk. Those actions are necessary so that Linux operating system can make use of the new storage space. You will take those actions next.

```
[ec2-user@ip-10-1-11-124 ~]$ 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  448K 190M  1% /run
/dev/xvda1       8.0G  1.6G  6.4G  21% /
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-10-1-11-124 ~]$
```

16. Create an ext3 file system on the new volume:

```
sudo mkfs -t ext3 /dev/sdf
```

The output should indicate that a new file system was created on the attached volume.

```
[ec2-user@ip-10-1-11-124 ~]$ sudo mkfs -t ext3 /dev/sdf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 18944166-3c8a-4617-8343-557359dc1504
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

[ec2-user@ip-10-1-11-124 ~]$
```

17. Create a directory for mounting the new storage volume:

```
sudo mkdir /mnt/data-store
```

18. Mount the new volume:

```
sudo mount /dev/sdf /mnt/data-store
```

To configure the Linux instance to mount this volume whenever the instance is started, you will need to add a line to */etc/fstab*. Run the command below to accomplish that:

```
echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab
```

```
[ec2-user@ip-10-1-11-124 ~]$ echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab  
/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2  
[ec2-user@ip-10-1-11-124 ~]$ cat /etc/fstab  
#UUID=985c5998-2242-4734-85d8-c0f972ff40e9 / xfs defaults,noatime 1 1  
#UUID=3021-32B7 /boot/efi vfat defaults,noatime,nid=0,gid=0,umask=0077,x-systemd.automount 0 2  
/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2  
[ec2-user@ip-10-1-11-124 ~]$
```

19. View the configuration file to see the setting on the last line:

```
cat /etc/fstab
```

20. View the available storage again:

```
df -h
```

Notice the last line. The output now lists */dev/xvdf* which is the new mounted volume.

```
[ec2-user@ip-10-1-11-124 ~]$ 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  448K  190M  1% /run  
/dev/xvda1      8.0G  1.6G  6.4G 21% /  
tmpfs          475M   0   475M  0% /tmp  
/dev/xvda128     10M  1.3M  8.7M 13% /boot/efi  
/dev/xvdf      975M  60K  924M  1% /mnt/data-store  
tmpfs          95M   0   95M  0% /run/user/1000  
[ec2-user@ip-10-1-11-124 ~]$
```

21. On your mounted volume, create a file and add some text to it.

```
sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
```

22. Verify that the text has been written to your volume.

```
cat /mnt/data-store/file.txt
```

```
[ec2-user@ip-10-1-11-124 ~]$ sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"  
[ec2-user@ip-10-1-11-124 ~]$ cat /mnt/data-store/file.txt  
some text has been written  
[ec2-user@ip-10-1-11-124 ~]$
```

Task 5: Create an Amazon EBS Snapshot

In this task, you will create a snapshot of your EBS volume.

You can create any number of point-in-time, consistent snapshots from Amazon EBS volumes at any time. Amazon EBS snapshots are stored in Amazon S3 with

high durability. New Amazon EBS volumes can be created out of snapshots for cloning or restoring backups. Amazon EBS snapshots can also be easily shared among AWS users or copied over AWS regions.

23. In the EC2 Console, choose Volumes and select My Volume.

24. In the Actions menu, select Create snapshot.

The screenshot shows the AWS EC2 Volumes console. A volume named 'My Volume' is selected in the list. The 'Actions' menu is open, and 'Create snapshot' is highlighted. Other options in the menu include 'Modify volume', 'Create snapshot lifecycle policy', 'Delete volume', 'Attach volume', 'Detach volume', 'Force detach volume', 'Manage auto-enabled I/O', 'Manage tags', and 'Resilience testing'.

25. Choose Add tag then configure:

- **Key: Name**
- **Value: My Snapshot**
- **Choose Create snapshot**

26. In the left navigation pane, choose Snapshots.

Your snapshot is displayed. The status will first have a state of *Pending*, which means that the snapshot is being created. It will then change to a state of *Completed*.

Note: Only used storage blocks are copied to snapshots, so empty blocks do not occupy any snapshot storage space.

The screenshot shows the AWS EC2 Snapshots console. A volume named 'vol-0cc43cb06bbe792cf' is selected. The 'Create snapshot' dialog is open, showing 'Snapshot details' and 'Tags'. Under 'Snapshot details', there is a 'Description' field with placeholder text 'Add a description for your snapshot...' and an 'Encryption info' section indicating 'Not encrypted'. Under 'Tags', there is a table with one tag: 'Key: name' and 'Value: My Snapshot'. At the bottom right of the dialog are 'Cancel' and 'Create snapshot' buttons.

27. In your EC2 Instance Connect session, delete the file that you created on your volume.

```
sudo rm /mnt/data-store/file.txt
```

28. Verify that the file has been deleted.

```
ls /mnt/data-store/
```

Your file has been deleted.

```
[ec2-user@ip-10-1-11-124 ~]$ sudo rm /mnt/data-store/file.txt
[ec2-user@ip-10-1-11-124 ~]$ ls /mnt/data-store/
lost+found
[ec2-user@ip-10-1-11-124 ~]$
```

Task 6: Restore the Amazon EBS Snapshot

If you ever wish to retrieve data stored in a snapshot, you can Restore the snapshot to a new EBS volume.

Create a Volume Using Your Snapshot

29. In the EC2 console, select My Snapshot.

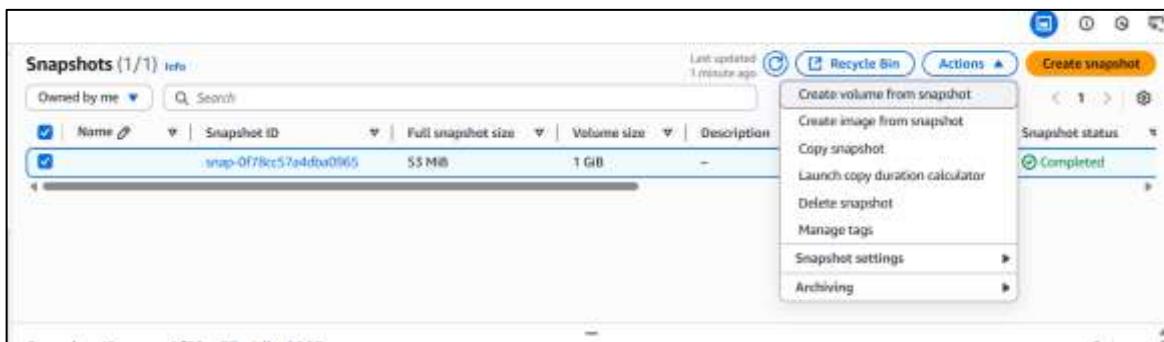
30. In the Actions menu, select Create volume from snapshot.

31. For Availability Zone, select the same availability zone that you used earlier.

32. Choose Add tag then configure:

- o **Key: Name**
- o **Value: Restored Volume**
- o **Choose Create volume**

Note: When restoring a snapshot to a new volume, you can also modify the configuration, such as changing the volume type, size or Availability Zone.



Attach the Restored Volume to Your EC2 Instance

33. In the left navigation pane, choose Volumes.

34. Select Restored Volume.

35. In the Actions menu, select Attach volume.

36. Choose the Instance field, then select the Lab instance that appears.

Note that the Device field is set to `/dev/sdg`. You will use this device identifier in a later task.

37. Choose Attach volume

The volume state is now *in-use*.

The screenshot shows the AWS Volumes (1/4) page. A table lists four volumes:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created
	vol-04e8b86c1a5a84751	gp3	9 GiB	3000	125	snap-0704622...	2025/10/06 16:47 GMT+5...
	vol-03118195f2f5ed06a	gp3	8 GiB	3000	125	snap-0704622...	2025/10/06 16:47 GMT+5...
My Volume	vol-0cc43cb06bbe792cf	gp2	1 GiB	100	-	-	2025/10/06 16:58 GMT+5...
Restored Vol...	vol-029512f318ee31cf2	gp3	1 GiB	3000	125	snap-0f78cc5...	2025/10/06 17:26 GMT+5...

A tooltip at the bottom says "Select volume: Restored Volume".

The screenshot shows the "Attach volume" dialog. It has sections for "Basic details" and "INSTANCE".

Basic details

- Volume ID: vol-029512f318ee31cf2 (Restored Volume)
- Availability Zone: us-east-1a
- INSTANCE: i-0bc7052b6a06ca11b (Last known)
- Device name: /dev/sdg

Notes:
Only instances in the same Availability Zone as the selected volume are displayed.
Recommended device names for Linux: /dev/xvdf for swap volumes, /dev/xvdf1 for data volumes.
Newer Linux kernels may rename your devices to /dev/nvme0 through /dev/nvme1 internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel Attach volume

Mount the Restored Volume

40. Create a directory for mounting the new storage volume:

```
sudo mkdir /mnt/data-store2
```

41. Mount the new volume:

```
sudo mount /dev/sdg /mnt/data-store2
```

42. Verify that volume you mounted has the file that you created earlier.

```
ls /mnt/data-store2/
```

You should see file.txt.

```
[ec2-user@ip-10-1-11-124 ~]$ sudo mkdir /mnt/data-store2
[ec2-user@ip-10-1-11-124 ~]$ sudo mount /dev/sdg /mnt/data-store2
[ec2-user@ip-10-1-11-124 ~]$ ls /mnt/data-store2/
file.txt  lost+found
```

❖ **Conclusion:**

This practical demonstrates the use of Amazon Elastic Block Store (EBS) for managing persistent storage on EC2 instances. It involves creating an EBS volume, attaching and mounting it to an EC2 instance, and storing data on the volume. It further illustrates how to create a snapshot of the volume to securely back up data and how to restore that snapshot to a new volume, ensuring data can be recovered when needed. Overall, the lab highlights EBS's capabilities for reliable, durable, and flexible storage management, emphasizing its usefulness for data persistence, backup, and restoration in cloud environments.