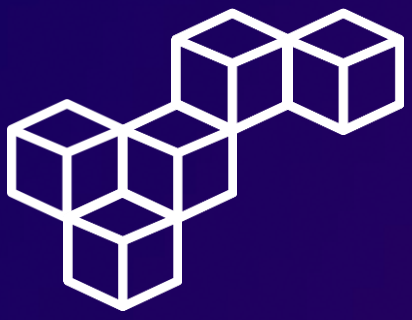
AWS re:Start LAB

Working with Amazon EBS

WEEK 10

aws re/start



Overview

Amazon Elastic Block Store (Amazon EBS) is a scalable, high-performance block storage service designed for use with Amazon EC2 instances. It provides persistent storage volumes that can be attached to EC2 instances, enabling data to persist independently of the instance's lifecycle. EBS volumes offer a range of performance options, including SSD-backed volumes for high-performance applications and HDD-backed volumes for large, throughput-intensive workloads. This flexibility allows users to optimize storage performance and cost according to their specific needs.

EBS volumes can be easily managed and scaled up or down, providing a seamless experience for handling growing storage requirements. Snapshots of EBS volumes can be created and stored in Amazon S3, providing a reliable backup solution and facilitating easy data recovery. Additionally, EBS supports encryption, ensuring that data at rest is secure. With its integration with other AWS services and its robust performance, Amazon EBS is a critical component for managing storage in cloud-based applications.

Topics covered

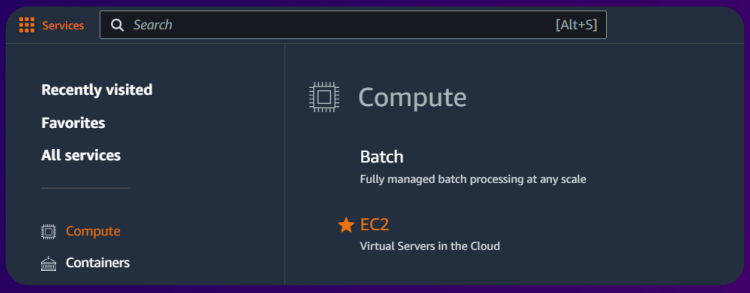
- Create an EBS volume.
- Attach and mount an EBS volume to an EC2 instance.
- Create a snapshot of an EBS volume.
- Create an EBS volume from a snapshot.

Task 1

Creating a new EBS volume

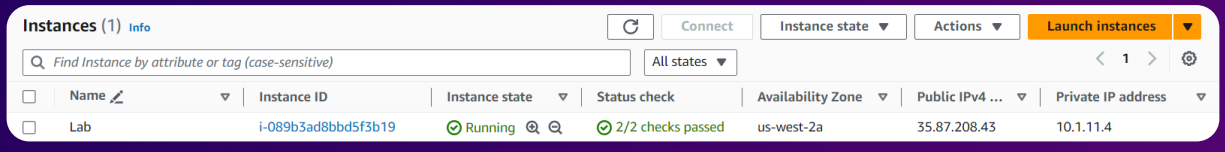
Step 1: Access the EC2 Management Console

Open the AWS Management Console, and select EC2.



Step 2: Review instances

Navigate to the **Instances** section, an EC2 instance named **Lab** has already been launched for your lab. Make note of the Availability Zone for the **Lab** instance.



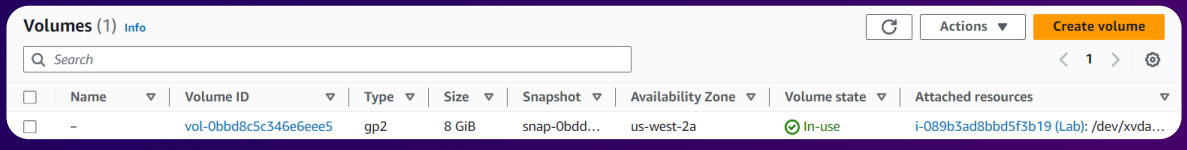
aws re/start

Task 1

Creating a new EBS volume

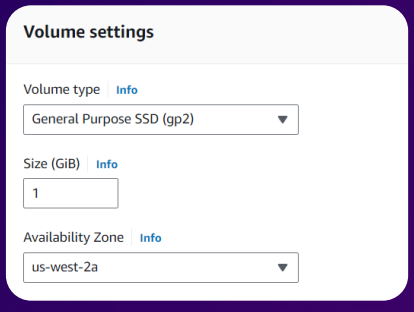
Step 3: Create volume

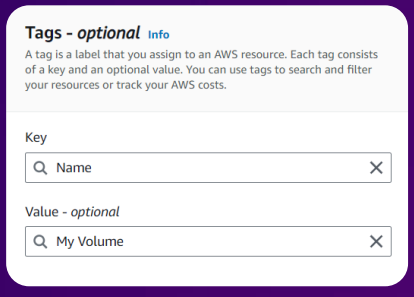
Navigate to the **Volumes** section, you see an existing (8 GiB) volume that the EC2 instance is using. Choose Create volume.



Step 4: Volume settings

In the **Volume settings** section, configure the following settings.





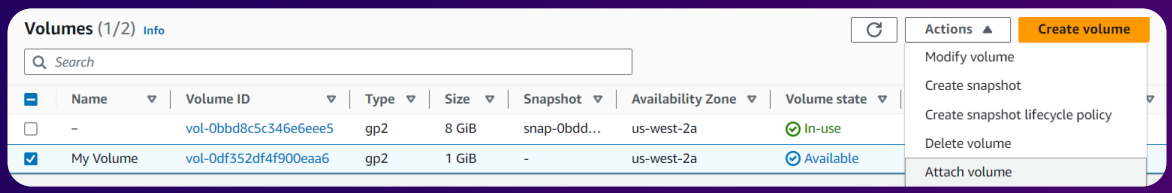
aws re/start

Task 2

Attaching the volume to an EC2 instance

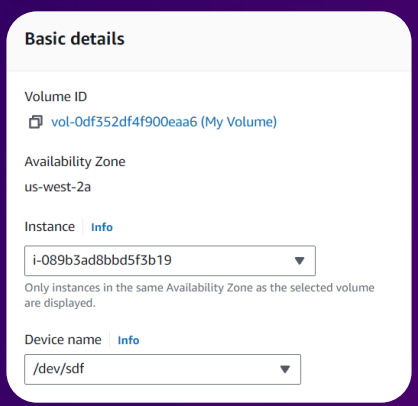
Step 1: Review Volume state

Select the newly created volume **My Volume** when the Volume state changes to Available. Then, from the Actions menu, choose Attach volume.



Step 2: Attach volume

From the Instance dropdown list, choose the **Lab** instance. The Device name field is set to /dev/sdf, make note of this device identifier. Choose Attach volume. The Volume state of your new volume should now be In-use.



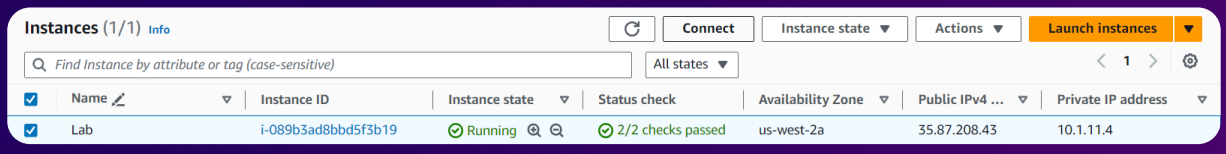
aws re/start

Task 3

Connecting to the Lab EC2 instance

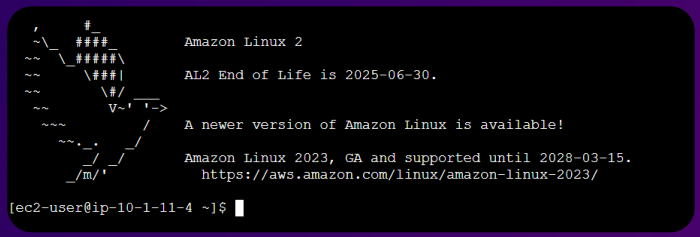
Step 1: Connect to the Lab instance

Navigate to the **Instances** section, select the **Lab** instance, and connect to the instance using EC2 Instance Connect.



Step 2: Review connection

A new browser tab opens with the EC2 Instance Connect terminal window.



aws re/start

Task 4

Creating and configuring the file system

Step 1: Review the available storage

To view the storage that is available on your instance, in the EC2 Instance Connect terminal, run the df -h command. The output shows the original 8 GB disk volume. Your new volume is not yet shown.

```
[ec2-user@ip-10-1-11-4 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        465M     0  465M   0% /dev
tmpfs           473M     0  473M   0% /dev/shm
tmpfs           473M  408K  472M   1% /run
tmpfs           473M     0  473M   0% /sys/fs/cgroup
/dev/nvme0n1p1  8.0G  1.6G  6.5G  20% /
tmpfs            95M     0   95M   0% /run/user/1000
[ec2-user@ip-10-1-11-4 ~]$
```

Step 2: Create an ext3 file system

To create an ext3 file system on the new volume, run the following mkfs command.

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mkfs -t ext3 /dev/sdf
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
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-10-1-11-4 ~]$
```

aws re/start

Task 4

Creating and configuring the file system

Step 3: Create a directory

To create a directory to mount the new storage volume, run the following mkdir command.

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mkdir /mnt/data-store
[ec2-user@ip-10-1-11-4 ~]$
```

Step 4: Mount the volume

To mount the new volume, run the following mount command. The echo command line ensures that the volume is mounted even after the instance is restarted. To view the configuration file to see the setting on the last line, run the following cat command.

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mount /dev/sdf /mnt/data-store
[ec2-user@ip-10-1-11-4 ~]$ 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-10-1-11-4 ~]$ cat /etc/fstab
#
UUID=e1c8828e-1de8-4a09-abb1-a211a3d69be0     /           xfs    defaults,noatime  1   1
/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2
[ec2-user@ip-10-1-11-4 ~]$
```

Task 4

Creating and configuring the file system

Step 5: Review the available storage

To view the available storage again, run the df -h command. The output now contains an additional line: /dev/nvme1n1.

```
[ec2-user@ip-10-1-11-4 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        465M     0  465M   0% /dev
tmpfs           473M     0  473M   0% /dev/shm
tmpfs           473M  408K  472M   1% /run
tmpfs           473M     0  473M   0% /sys/fs/cgroup
/dev/nvme0n1p1  8.0G  1.6G  6.5G  20% /
tmpfs            95M     0   95M   0% /run/user/1000
/dev/nvme1n1    975M   60K  924M   1% /mnt/data-store
[ec2-user@ip-10-1-11-4 ~]$
```

Step 6: Create a file on the mounted volume

To create a file and add some text on the mounted volume, run the following sh command. To verify that the text has been written to your volume, run the following cat command.

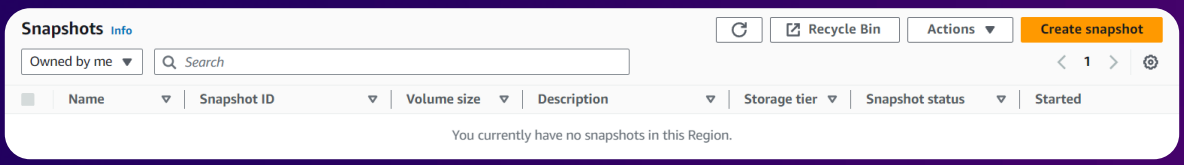
```
[ec2-user@ip-10-1-11-4 ~]$ sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
[ec2-user@ip-10-1-11-4 ~]$ cat /mnt/data-store/file.txt
some text has been written
[ec2-user@ip-10-1-11-4 ~]$
```

Task 5

Creating an Amazon EBS snapshot

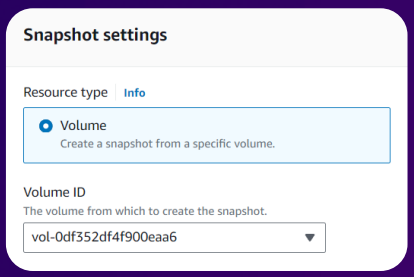
Step 1: Create snapshot

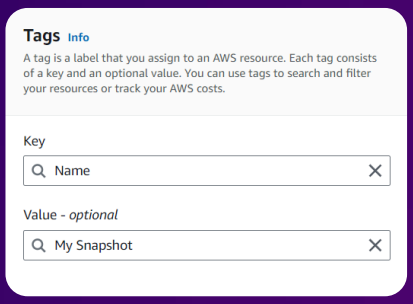
Navigate to the **Snapshots** section, and select Create snapshot.



Step 2: Snapshot settings

In the **Snapshot settings** section, configure the following settings.





aws re/start

Task 5

Creating an Amazon EBS snapshot

Step 3: Delete file

In your EC2 Instance Connect terminal window, to delete the file that you created on your volume, run the following rm command.

```
[ec2-user@ip-10-1-11-4 ~]$ sudo rm /mnt/data-store/file.txt
[ec2-user@ip-10-1-11-4 ~]$
```

Step 4: Review file deletion

To verify that the file has been deleted, run the following ls command. The message ls: cannot access /mnt/data-store/file.txt: No such file or directory displays. Your file has been deleted.

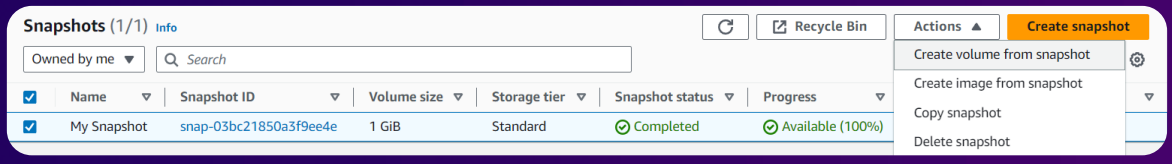
```
[ec2-user@ip-10-1-11-4 ~]$ ls /mnt/data-store/file.txt
ls: cannot access /mnt/data-store/file.txt: No such file or directory
[ec2-user@ip-10-1-11-4 ~]$
```

Task 6

Restoring the Amazon EBS snapshot

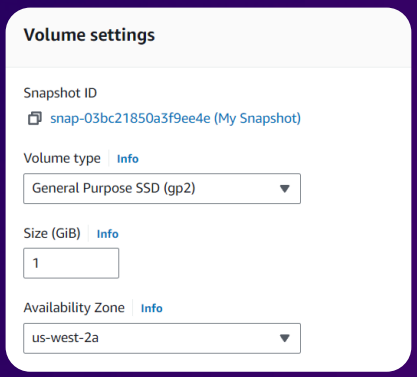
Step 1: Create volume from snapshot

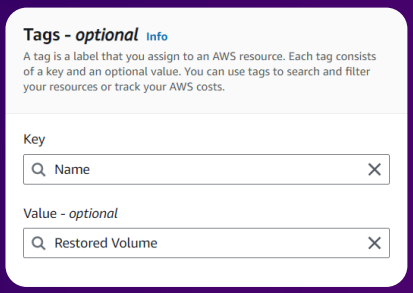
Navigate to the **Snapshots** section, select **My Snapshot**. From the Actions menu, choose Create volume from snapshot.



Step 2: Volume settings

In the **Volume settings** section, configure the following settings.





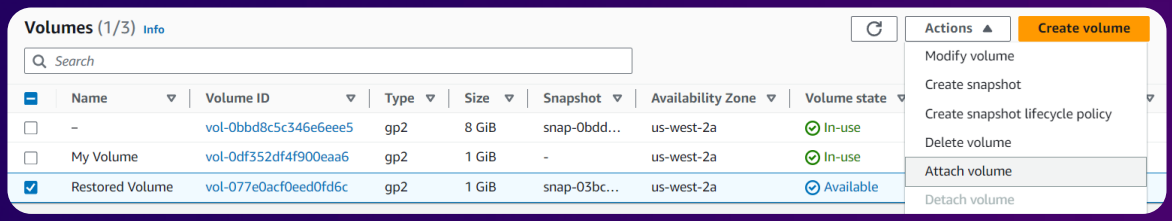
aws re/start

Task 6

Restoring the Amazon EBS snapshot

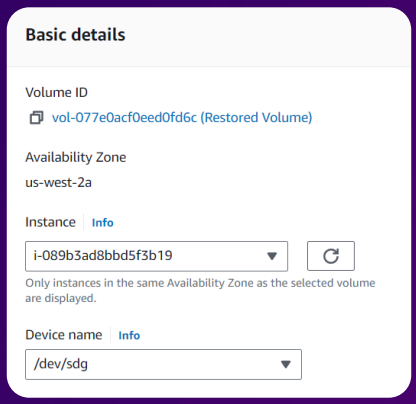
Step 3: Review Volume state

In the **Volumes** section, select the newly created volume **Restored Volume** when the Volume state changes to Available. Then, from the Actions menu, choose Attach volume.



Step 4: Attach volume

From the Instance dropdown list, choose the **Lab** instance. The Device name field is set to /dev/sdg, make note of this device identifier. Choose Attach volume. The Volume state of your new volume should now be In-use.



aws re/start

Task 6

Restoring the Amazon EBS snapshot

Step 5: Create a new directory

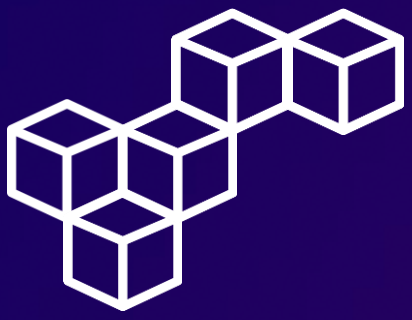
To create a directory for mounting the new storage volume, in the EC2 Instance Connect terminal, run the following mkdir command.

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mkdir /mnt/data-store2
[ec2-user@ip-10-1-11-4 ~]$
```

Step 6: Mount the new volume

To mount the new volume, run the following mount command. To verify that the volume that you mounted has the file that you created earlier, run the following ls command. You should see the file.txt file.

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mount /dev/sdg /mnt/data-store2
[ec2-user@ip-10-1-11-4 ~]$ ls /mnt/data-store2/file.txt
/mnt/data-store2/file.txt
[ec2-user@ip-10-1-11-4 ~]$
```



Conclusions

Creating an EBS Volume
Creating an EBS volume provides scalable, persistent storage for EC2 instances.

Attaching a Volume to an EC2 Instance
Attaching an EBS volume to an EC2 instance is crucial for expanding storage capacity and enhancing data durability.

Creating and Configuring the File System
Creating and configuring the file system ensures the EBS volume is ready for data storage and access by the instance.

The mkfs and mount commands
Utilize mkfs to format the EBS volume with a specific file system, and mount to attach it to a directory in the EC2 instance's file system.

Creating an Amazon EBS Snapshot
Capture a point-in-time backup of an EBS volume with Amazon EBS snapshots.

Restoring an Amazon EBS Snapshot
Easily restore an EBS volume to a previous state by using an Amazon EBS snapshot.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru