Guided Lab: Managing Instance Volumes Using Elastic Block Store (EBS)

Description

In AWS, you can effortlessly attach and detach EBS volumes from one EC2 instance to another. This flexibility is great for things such as upgrading instances or moving data around without much hassle. It's a handy feature for scaling your storage needs or handling system maintenance with minimal disruption.

Throughout this hands-on lab, you will gain experience in creating an EC2 instance with an additional EBS volume. You will learn to attach and detach an EBS to/from a specific EC2 instance and learn the process of volume deletion.

Prerequisites

To ensure successful completion of this lab, you must have prior experience in creating EC2 instances and be familiar with their essential components. If you feel that your knowledge in this area is insufficient, we highly recommend taking the following labs to gain the necessary understanding:

- Creating an Amazon EC2 instance (Linux)
- Creating an Amazon Machine Image (AMI) from an EBS-backed EC2 instance
- Familiarity with basic Linux commands is beneficial but not required
- Connecting to your EC2 instance using EC2 Instance Connect

Objectives

In this lab, you will:

- Create an EC2 instance with an additional EBS volume.
- Create a new EBS volume.
- Attach and Detach an EBS volume

Subscribe to access AWS
PlayCloud Labs

Lab Steps

Creating an EC2 instance

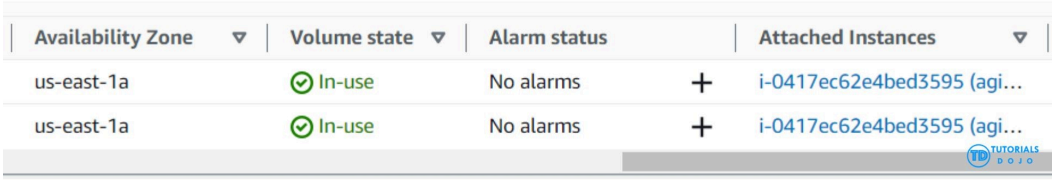
Create two EC2 instances using the following configurations:

1. **Name**: agila1 & agila2
2. **Instance type**: t2.micro
3. **AMI**: Linux
4. **Key pair**: Create a new Key Pair
   - **Key Pair name**: agila_keys
   - **Key Pair Type:** RSA
   - **Private key file format:** .pem
   - Click **Create key pair**

Review your instance configurations and click the "**Launch Instance**" button.

Creating a new EBS volume

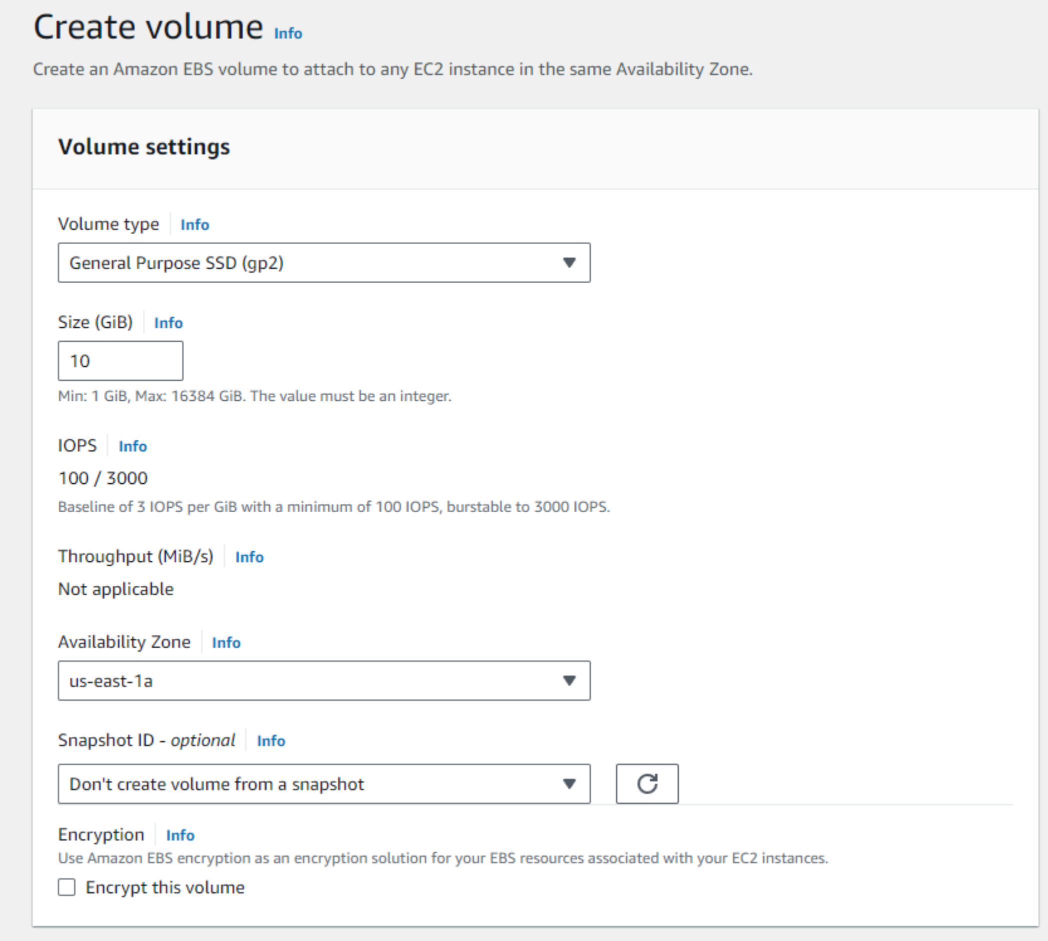
1. On the **Volumes** listing page, make sure to take note of the **Availability Zone**:
Note: Create the volume in the same availability zone. If you do not do this, you will not be able to attach the volume.



Availability Zone	Volume state	Alarm status		Attached Instances
us-east-1a	✓ In-use	No alarms	+	i-0417ec62e4bed3595 (agi…
us-east-1a	✓ In-use	No alarms	+	i-0417ec62e4bed3595 (agi…

2. Create a new volume, click on **Create volume**. This will take you to a **Volume setting** page and set the following values before clicking **Create volume.**

- **Volume Type**: **General purpose SSD (GP2)**
- **Size**: *10* GiB
- **Availability Zone**: us-east-1a (Depends on where the AZ of your EBS Volumes are)



Create volume Info

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

Volume settings

Volume type | Info
General Purpose SSD (gp2)

Size (GiB) | Info
10
Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS | Info
100 / 3000
Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS.

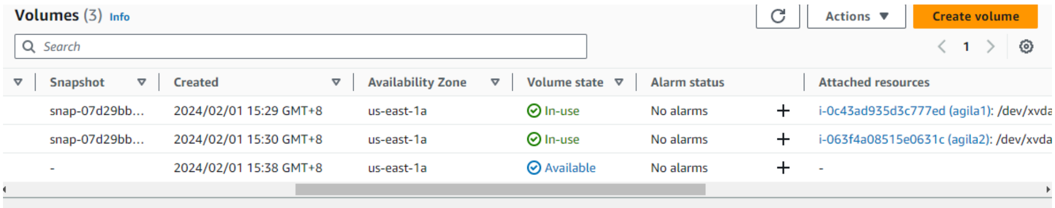
Throughput (MiB/s) | Info
Not applicable

Availability Zone | Info
us-east-1a

Snapshot ID - *optional* | Info
Don't create volume from a snapshot

Encryption | Info
Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.
□ Encrypt this volume

3. Wait until the **Volume state** is **Available** (refresh the page every 10/15 seconds):



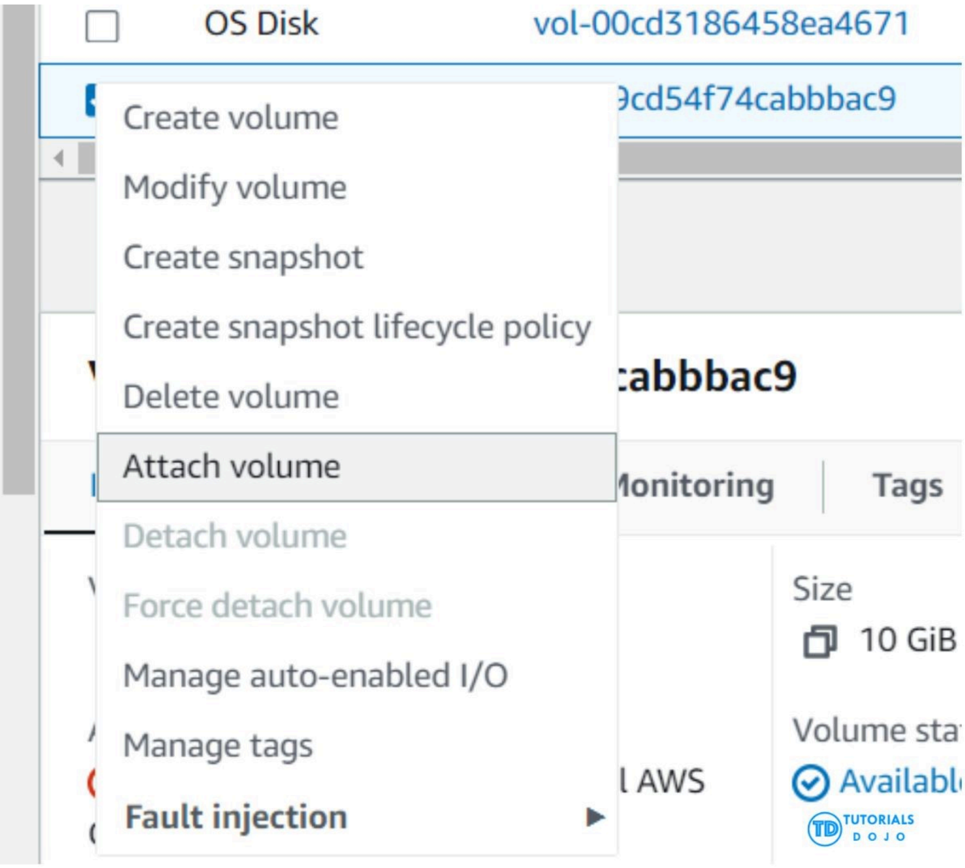
Volumes (3) Info

	Snapshot	Created	Availability Zone	Volume state	Alarm status		Attached resources
	snap-07d29bb…	2024/02/01 15:29 GMT+8	us-east-1a	✓ In-use	No alarms	+	i-0c43ad935d3c777ed (agila1): /dev/xvda
	snap-07d29bb…	2024/02/01 15:30 GMT+8	us-east-1a	✓ In-use	No alarms	+	i-063f4a08515e0631c (agila2): /dev/xvda
	-	2024/02/01 15:38 GMT+8	us-east-1a	✓ Available	No alarms	+	-

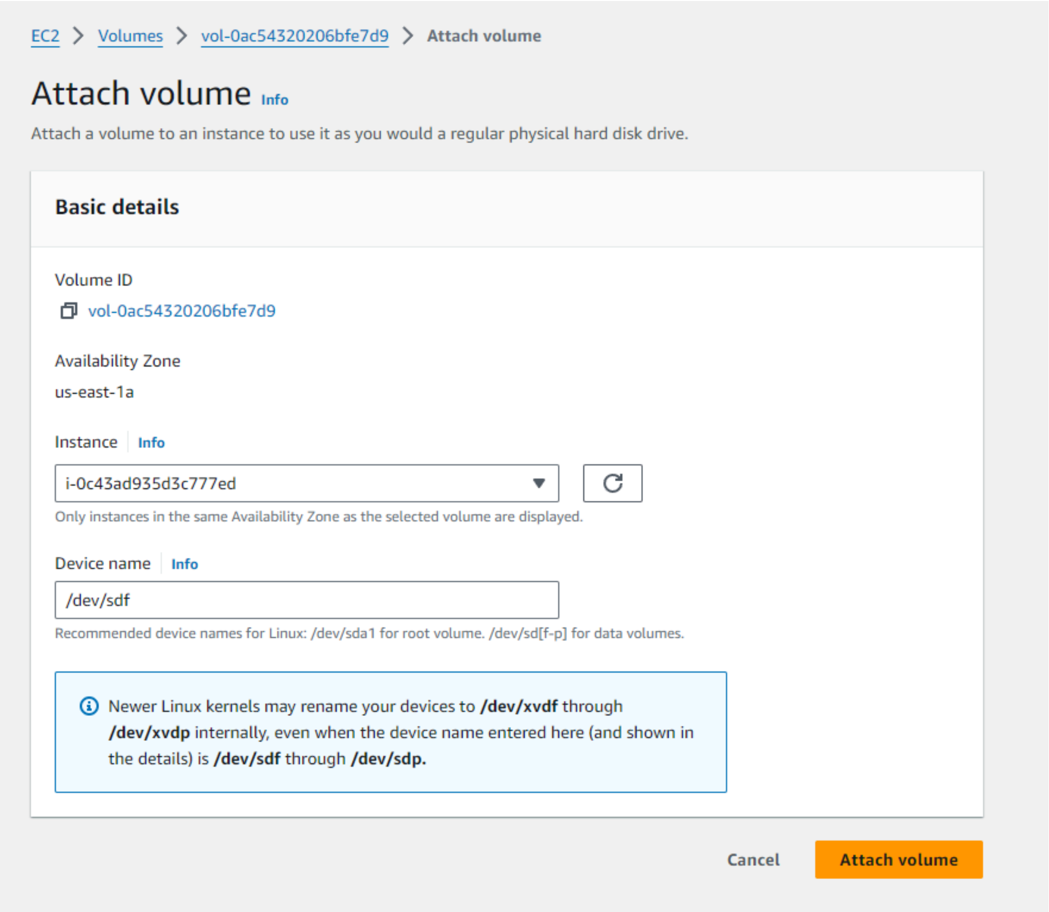
Creating a change on the EBS Volumes and reattaching it to an EC2 instance

To attach the newly created EBS Volume, follow the steps below:

1. Right-click on the newly created EBS volume and select Attach Volume.



2. Select the **instance ID** of the **agila1 instance**. Then, click on the **Attach volume** button.



EC2 > Volumes > vol-0ac54320206bfe7d9 > Attach volume

Attach volume Info

Attach a volume to an instance to use it as you would a regular physical hard disk drive.

Basic details

Volume ID
vol-0ac54320206bfe7d9

Availability Zone
us-east-1a

Instance | Info
i-0c43ad935d3c777ed
Only instances in the same Availability Zone as the selected volume are displayed.

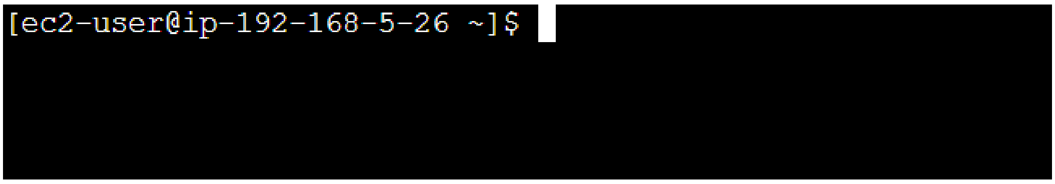
Device name | Info
/dev/sdf
Recommended device names for Linux: /dev/sda1 for root volume. /dev/sd[f-p] for data volumes.

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.**

Note: The **device name** may be automatically renamed by newer Linux kernels, even when it is initially entered as **/dev/sdf**.

Format and mount an attached volume

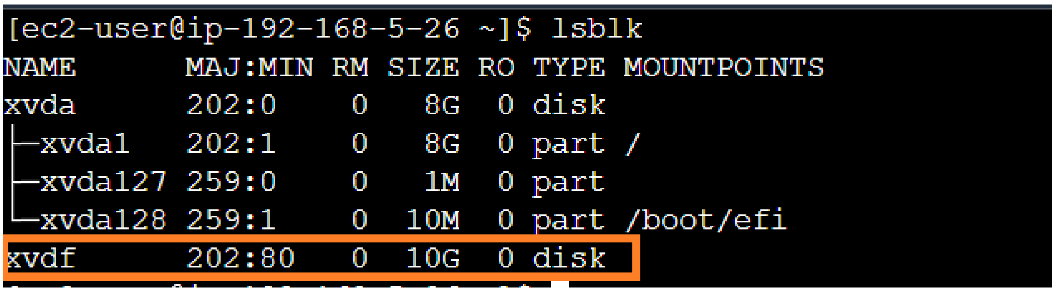
1. Connect to the **agila1 instance** using EC2 Instance Connect.



```
[ec2-user@ip-192-168-5-26 ~]$
```

2. Check the available disk devices and their mount points using this command.

```
lsblk
```

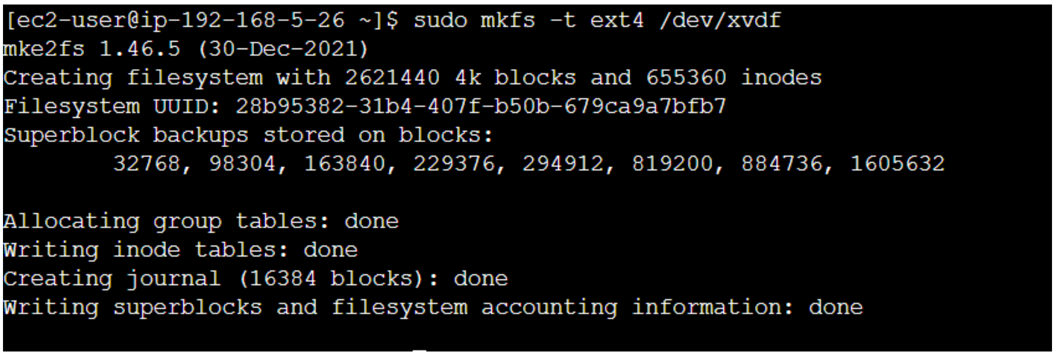


```
[ec2-user@ip-192-168-5-26 ~]$ lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda        202:0    0   8G  0 disk
├─xvda1     202:1    0   8G  0 part /
├─xvda127 259:0      0   1M  0 part
└─xvda128 259:1      0  10M  0 part /boot/efi
xvdf        202:80   0  10G  0 disk
```

The `/dev/xvda` device is the root EBS volume of the instance, which has three partitions named `xvda1` , `xvda127` , and `xvda128` . The EBS volume on `/dev/xvdf` is the new 10GB volume that we created. We have to format it first and then mount it for it to be usable.

3. Now, format the Data Volume with an ext4 file system by using this script.

```
sudo mkfs -t ext4 /dev/xvdf
```



```
[ec2-user@ip-192-168-5-26 ~]$ sudo mkfs -t ext4 /dev/xvdf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: 28b95382-31b4-407f-b50b-679ca9a7bfb7
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

4. Next, create a mount point in the Data Volume. Use the **`mkdir`** command. The mount point is where the volume is located in the file system tree and where you read and write files after you mount the volume. In this lab, create a directory named `/playcloud` .

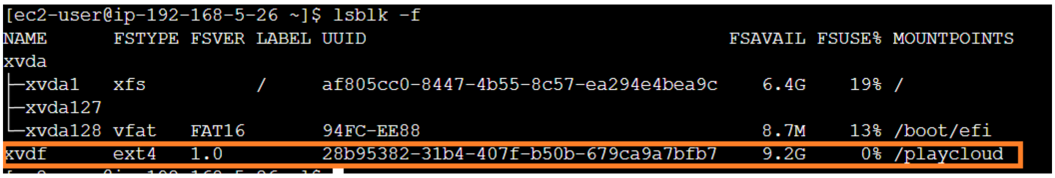
```
sudo mkdir /playcloud
```

5. Mount the volume or partition to the `/playcloud` mount point.

```
sudo mount /dev/xvdf /playcloud
```

6. Verify if the Data Volume is successfully mounted. Use `lsblk -f` command to view your available disk devices and their mount points.

```
lsblk -f
```

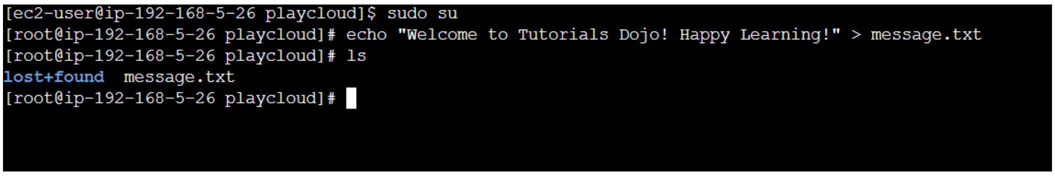


```
[ec2-user@ip-192-168-5-26 ~]$ lsblk -f
NAME       FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
xvda
├─xvda1    xfs          /     af805cc0-8447-4b55-8c57-ea294e4bea9c    6.4G    19% /
├─xvda127
└─xvda128 vfat   FAT16        94FC-EE88                               8.7M    13% /boot/efi
xvdf       ext4   1.0          28b95382-31b4-407f-b50b-679ca9a7bfb7    9.2G     0% /playcloud
```

After mounting the Data volume, we will make changes to it and reattach it to the **agila2 instance**. Finally, we will check if the changes persist after reattachment.

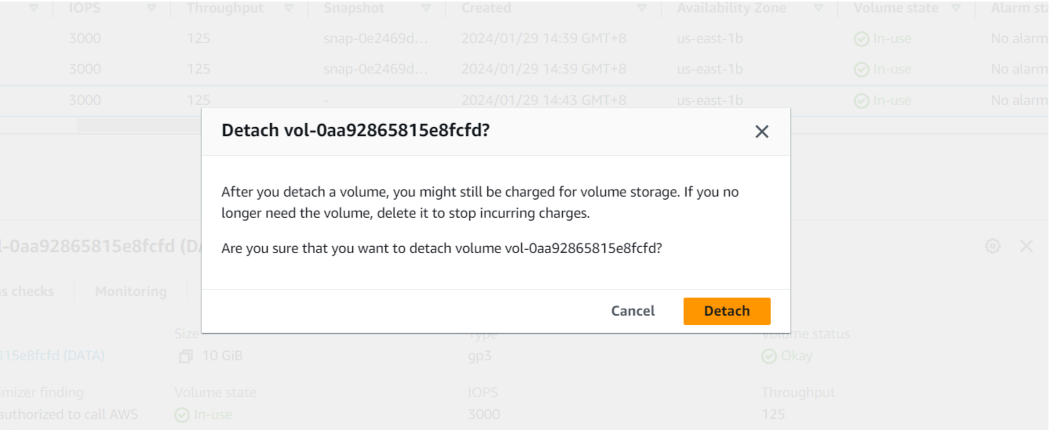
7. Now, go to the `/playcloud` directory and create a file inside.

```
#Switch your user account
sudo su

#Create a file txt
echo "Welcome Tutorials Dojo! Happy Learning!" >
message.txt

#Check the if the file is created
ls
```



```
[ec2-user@ip-192-168-5-26 playcloud]$ sudo su
[root@ip-192-168-5-26 playcloud]# echo "Welcome to Tutorials Dojo! Happy Learning!" > message.txt
[root@ip-192-168-5-26 playcloud]# ls
lost+found  message.txt
[root@ip-192-168-5-26 playcloud]# 
```

8. Go back to the Volumes listing page. Right-click on the **Data Volume** → select the **Detach volume** action and click **Detach** in the confirmation dialogue box.

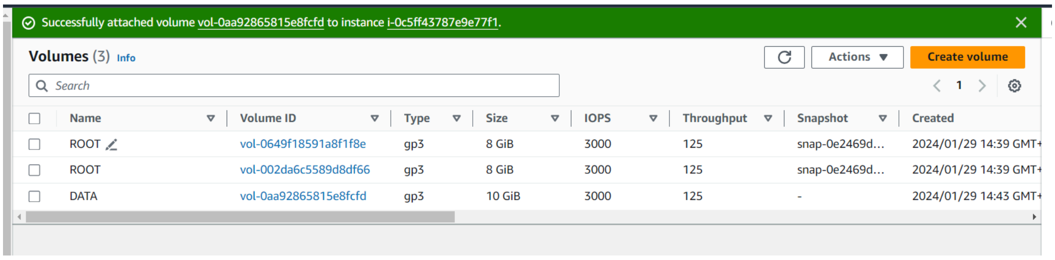


Detach vol-0aa92865815e8fcfd?

After you detach a volume, you might still be charged for volume storage. If you no longer need the volume, delete it to stop incurring charges.

Are you sure that you want to detach volume vol-0aa92865815e8fcfd?

9. Now, attach the **Data Volume** to the **agila2 instance**.

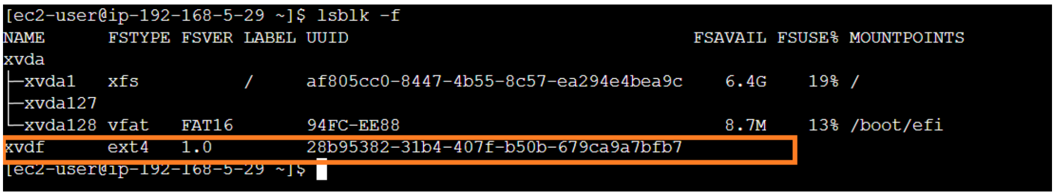


Successfully attached volume vol-0aa92865815e8fcfd to instance i-0c5ff43787e9e77f1.

Volumes (3) Info

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created
ROOT	vol-0649f18591a8f1f8e	gp3	8 GiB	3000	125	snap-0e2469d...	2024/01/29 14:39 GMT+
ROOT	vol-002da6c5589d8df66	gp3	8 GiB	3000	125	snap-0e2469d...	2024/01/29 14:39 GMT+
DATA	vol-0aa92865815e8fcfd	gp3	10 GiB	3000	125	-	2024/01/29 14:43 GMT+

10. Next, connect to your **agila2 instance** using EC2 Instance Connect. Check the available disk device and their mount points.



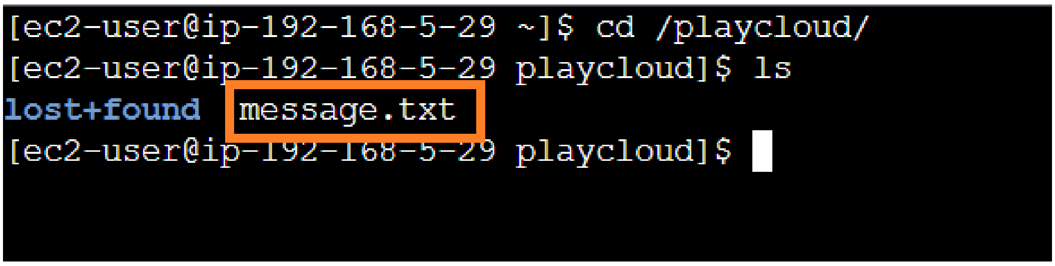
```
[ec2-user@ip-192-168-5-29 ~]$ lsblk -f
NAME       FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
xvda
├─xvda1    xfs          /     af805cc0-8447-4b55-8c57-ea294e4bea9c    6.4G    19% /
├─xvda127
└─xvda128  vfat   FAT16       94FC-EE88                               8.7M    13% /boot/efi
xvdf       ext4   1.0         28b95382-31b4-407f-b50b-679ca9a7bfb7
[ec2-user@ip-192-168-5-29 ~]$
```

11. Since we have already formatted the Data Volume, we only need to create a directory and mount it again to the **agila2 instance**.

```
#Add a directory
sudo mkdir /playcloud

#Mounting the EBS Volume
sudo mount /dev/xvdf /playcloud
```

12. Verify if the file added earlier exists.

```
cd /playcloud
ls
```



```
[ec2-user@ip-192-168-5-29 ~]$ cd /playcloud/
[ec2-user@ip-192-168-5-29 playcloud]$ ls
lost+found  message.txt
[ec2-user@ip-192-168-5-29 playcloud]$
```

That's it! Congratulations on completing the basics of managing your Instance Volumes using Elastic Block Store (EBS)! As you continue your journey, remember that EBS offers additional features and optimizations to enhance your storage experience. Happy learning!