

Module 2: EC2 and EFS Assignment

Tasks To Be Performed:

1. Create an EFS and connect it to 3 different EC2 instances. Make sure that all instances have different operating systems. For instance, Ubuntu, RedHat Linux and Amazon Linux 2.

Step 1 : Launch 3 EC2 instances with different Operating Systems.

- Ubuntu EC2 instance, RedHat Linux EC2 instance, Amazon Linux 2 EC2 instance:

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input checked="" type="checkbox"/>	Amazon linux	i-025e53a6fcd003d0b	Running	t2.micro	2/2 checks passed
<input checked="" type="checkbox"/>	Redhat Linux	i-049edf03017d7194a	Running	t2.micro	2/2 checks passed
<input checked="" type="checkbox"/>	Ubuntu Instance	i-02f6c9c2baac1b801	Running	t2.micro	2/2 checks passed

Step 2: Create an EFS and connect to all 3 EC2 instances

- Go to Elastic filesystem (EFS) and Create filesystem (fill the details)

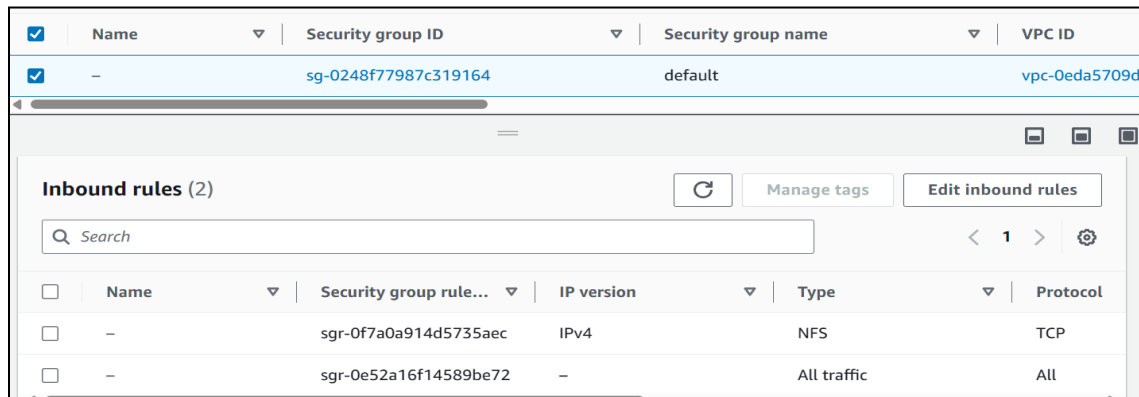
File systems (1)						
<input type="text" value="Filter by property values"/> < 1 >						
	Name	File system ID	Encrypte d	Total size	Size in Standard	Size in IA
<input type="radio"/>	EFS-Assignment	fs-0064da88fa96ee834	Encrypted	6.00 KiB	6.00 KiB	0 Bytes

- Click on the filesystem and go to Network
- Copy the SG id

Metered size	Monitoring	Tags	File system policy	Access points	Network	Replication
Network <input type="button" value="Manage"/>						
Availability zone	Mount target ID	Subnet ID	Mount target state	IP address	Network interface ID	Security groups
us-east-1a	fsmt-03a95b2b5be7c12f8	subnet-0534735fd7e7d4009	Available	172.31.5.29	eni-08a7fdc5a457f3686	sg-0248f77987c319164 (default)

- Go to EC2 Dashboard and click on security group

- Paste the SG id and edit the inbound rules ----> add rule for NFS



Step 3:

- Connect to all three EC2 instances and run below commands on all instances.

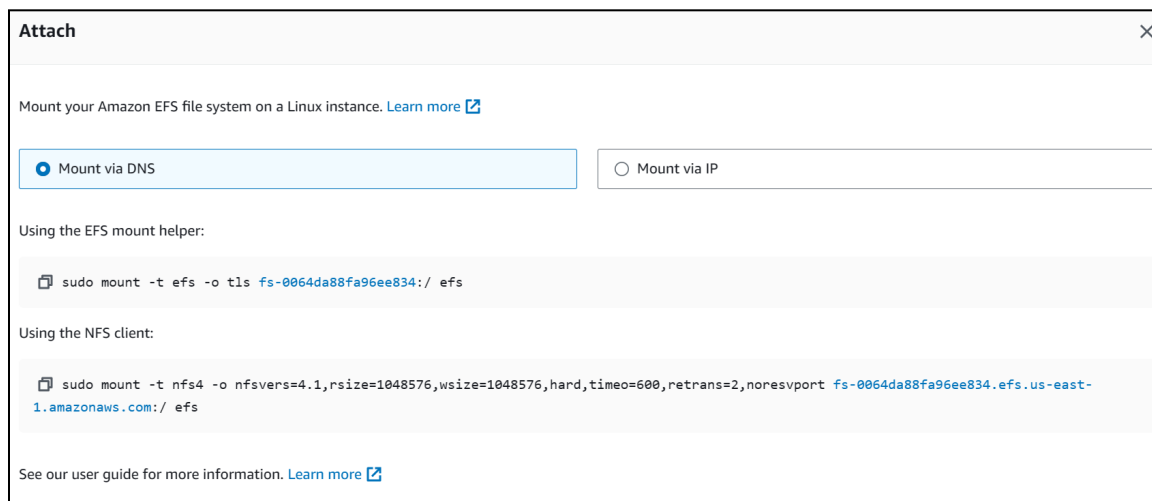
sudo apt update -y (for ubuntu)/ # sudo yum update -y (for Amazon Linux and RedHat)

sudo apt install nfs-common -y (for ubuntu)/ # sudo yum install nfs* -y (for Amazon Linux and RedHat)

sudo mkdir /efs-dir

Go to AWS console and look for EFS

Click on attach and new window will pop up



Step 4: Mount EFS filesystem on all 3 instances

- select "Mount via DNS" and copy the command under "Using the NFS client"
- paste the command on the terminals and change the /efs with /home/ubuntu/efs-dir (ubuntu Instance) and /home/ec2-user/efs-dir (Amazon Linux and Redhat)

```
ubuntu@ip-172-31-44-93:~$ df -TH
Filesystem                                Type      Size  Used Avail Use% Mounted on
/dev/root                                ext4       8.2G  1.9G  6.3G  24% /
tmpfs                                     tmpfs      498M    0  498M   0% /dev/shm
tmpfs                                     tmpfs      200M   881k  199M   1% /run
tmpfs                                     tmpfs       5.3M    0   5.3M   0% /run/lock
/dev/xvda15                              vfat      110M   6.4M  104M   6% /boot/efi
tmpfs                                     tmpfs      100M   4.1k  100M   1% /run/user/1000
fs-0064da88fa96ee834.efs.us-east-1.amazonaws.com:/ nfs4       9.3E    0   9.3E   0% /home/ubuntu/efs-dir
ubuntu@ip-172-31-44-93:~$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.3 LTS"
NAME="Ubuntu"
```

```
[ec2-user@ip-172-31-32-182 ~]$ df -TH
Filesystem                                Type      Size  Used Avail Use% Mounted on
devtmpfs                                 devtmpfs   4.2M    0   4.2M   0% /dev
tmpfs                                     tmpfs      396M    0  396M   0% /dev/shm
tmpfs                                     tmpfs      159M   6.7M  152M   5% /run
/dev/xvda4                              xfs        9.9G   1.5G   8.4G  16% /
/dev/xvda3                              xfs       563M  170M  393M  31% /boot
/dev/xvda2                              vfat       210M   7.3M  203M   4% /boot/efi
tmpfs                                     tmpfs       80M    0    80M   0% /run/user/1000
fs-0064da88fa96ee834.efs.us-east-1.amazonaws.com:/ nfs4       9.3E    0   9.3E   0% /home/ec2-user/efs-dir
[ec2-user@ip-172-31-32-182 ~]$ cat /etc/os-release
NAME="Red Hat Enterprise Linux"
```

```
[ec2-user@ip-172-31-45-84 ~]$ df -TH
Filesystem                                Type      Size  Used Avail Use% Mounted on
devtmpfs                                 devtmpfs   4.2M    0   4.2M   0% /dev
tmpfs                                     tmpfs      498M    0  498M   0% /dev/shm
tmpfs                                     tmpfs      200M   3.0M  197M   2% /run
/dev/xvda1                              xfs        8.6G   1.7G   6.9G  20% /
tmpfs                                     tmpfs      498M    0  498M   0% /tmp
/dev/xvda128                            vfat       11M    1.4M   9.2M  13% /boot/efi
tmpfs                                     tmpfs      100M    0   100M   0% /run/user/1000
fs-0064da88fa96ee834.efs.us-east-1.amazonaws.com:/ nfs4       9.3E    0   9.3E   0% /home/ec2-user/efs-dir
[ec2-user@ip-172-31-45-84 ~]$ cat /etc/os-release
NAME="Amazon Linux"
```

Problem Statement:

You work for XYZ Corporation. Your corporation is working on an application and they require secured web servers on Linux to launch the application.

Tasks To Be Performed:

1. Create an instance in the US-East-1 (N. Virginia) region with Linux OS and manage the requirement of web servers of your company using AML.
2. Replicate the instance in the US-West-2 (Oregon) region.
3. Build two EBS volumes and attach them to the instance in the US-East-1 (N. Virginia) region.
4. Delete one volume after detaching it and extend the size of the other volume.
5. Take backup of this EBS volume.

Task 1: Create an instance in the US-East-1 (N. Virginia) region with Linux OS and manage the requirement of web servers of your company using AML.

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	Linux Instance	i-02f252b474b3f93eb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d

Connect to the instance and run below commands:

```
# sudo apt update -y
```

```
# sudo apt install apache2 -y
```

```
# cd /var/www/html
```

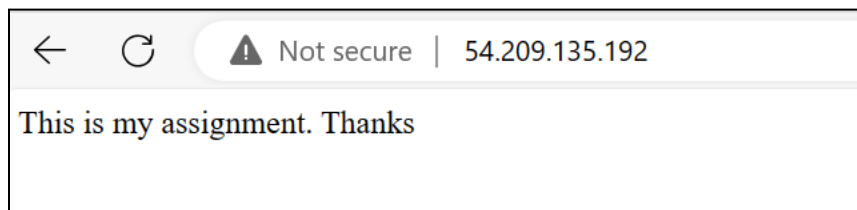
```
# sudo rm index.html
```

```
# sudo vim index.html -----> added content "This is my assignment. Thanks"
```

```
ubuntu@ip-172-31-33-36:~$ cd /var/www/html/
ubuntu@ip-172-31-33-36:/var/www/html$ ll
total 12
drwxr-xr-x 2 root root 4096 Oct 26 02:54 ./
drwxr-xr-x 3 root root 4096 Oct 26 02:43 ../
-rw-r--r-- 1 root root  14 Oct 26 02:54 index.html
ubuntu@ip-172-31-33-36:/var/www/html$ sudo rm index.html
ubuntu@ip-172-31-33-36:/var/www/html$ ll
total 8
drwxr-xr-x 2 root root 4096 Dec  4 16:10 ./
drwxr-xr-x 3 root root 4096 Oct 26 02:43 ../
```

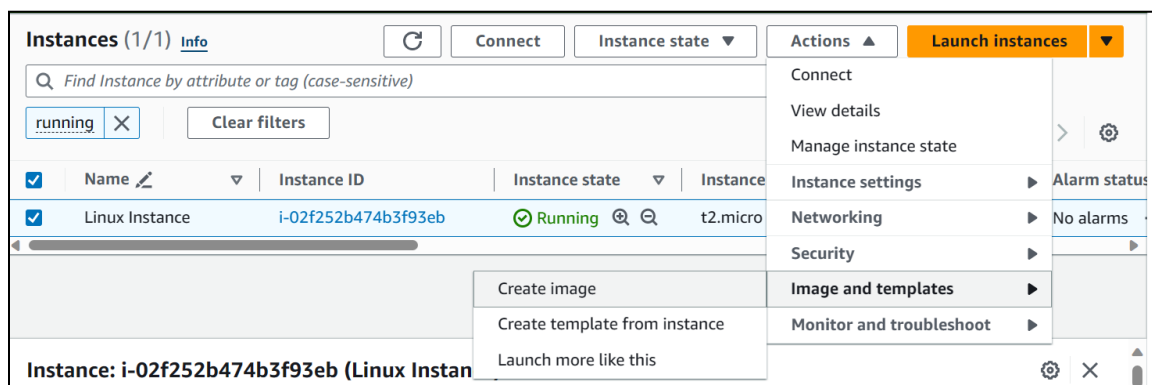
```
ubuntu@ip-172-31-33-36:/var/www/html$ sudo vim index.html
ubuntu@ip-172-31-33-36:/var/www/html$ cat index.html
This is my assignment. Thanks
ubuntu@ip-172-31-33-36:/var/www/html$
```

- Go to web browser and paste the public IP address, below is the screenshot for your reference.



Create AMI :

Go to Actions -----> Images & templates -----> Create Image -----> Fill the details -----> Create Image



Task 2 : Replicate the instance in the US-West-2 (Oregon) region.

- Go to Actions -----> Copy AMI

The screenshot shows the Amazon Machine Images (AMIs) console. At the top, there are buttons for 'Recycle Bin', 'EC2 Image Builder', and 'Actions'. Below these is a search bar and a table of AMIs. One AMI, 'Linux instance AMI', is selected. The 'Actions' menu is open, showing options like 'Copy AMI', 'Edit AMI permissions', 'Request Spot Instances', 'Manage tags', 'Deregister AMI', 'Change description', 'Configure fast launch', 'Manage AMI Deprecation', 'Register instance store-backed AMI', and 'Disable AMI'. A 'Launch instance from AMI' button is also visible.

Source
167278238680/Linux insta

AMI ID: ami-00121b6468677a72a

Details | Permissions | Storage | Tags

- Select the region where AMI needs to be copied, in our case selected Oregon

The screenshot shows the 'Copy Amazon Machine Image (AMI)' dialog box. It contains the following fields and options:

- Original AMI ID:** ami-00121b6468677a72a
- AMI copy name:** Linux instance AMI
- AMI copy description:** [Copied ami-00121b6468677a72a from us-east-1] Linux instance AMI
- Destination Region:** US West (Oregon)
- ☐ **Copy tags** (Includes your user-defined AMI tags when copying the AMI.)
- ☐ **Encrypt EBS snapshots of AMI copy** (Encrypts all snapshots in the AMI copy with the same key.)

Buttons at the bottom: Cancel, Copy AMI

- Copied AMI to Oregon region.

The screenshot shows the Amazon Machine Images (AMIs) console for the Oregon region. The table of AMIs now includes the copied AMI:

Name	AMI name	AMI ID	Source
Linux instance AMI		ami-0a94fb35252407b1f	167278238680/Linux instance AMI

Task 3: Build two EBS volumes and attach them to the instance in the US-East-1 (N. Virginia) region.

- Go to Volumes -----> Create Volume

Volumes (6) Info									
<input type="text" value="Search"/>									
<div>< 1 > ⚙</div>									
<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	
<input type="checkbox"/>	Volume1	vol-0e4ebf24cb56363f1	gp3	1 GiB	3000	125	-	2023/12/04 22:41 GMT	
<input type="checkbox"/>	Volume2	vol-0b0527e5b377a0620	gp3	1 GiB	3000	125	-	2023/12/04 22:42 GMT	

- Go to Actions -----> Attach Volume

Volumes (1/6) Info							Actions	Create volume
<input type="text" value="Search"/>							<div>Modify volume</div> <div>Create snapshot</div> <div>Create snapshot lifecycle policy</div> <div>Delete volume</div> <div>Attach volume</div>	
<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS			
<input checked="" type="checkbox"/>	Volume1	vol-0814f90b28b4852fb	gp3	1 GiB	3000			
<input type="checkbox"/>	Volume2	vol-0d307c490de5da8af	gp3	1 GiB	3000			
<input type="checkbox"/>		vol-09b76c8bd338e72c8	gp3	8 GiB	3000			

Volume ID

 **vol-0814f90b28b4852fb (Volume1)**

Availability Zone

us-east-1d

Instance

[Info](#)

i-02f252b474b3f93eb


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**.

Volumes (1/6) Info							Actions	Create volume
<input type="text" value="Search"/>							<div>Modify volume</div> <div>Create snapshot</div> <div>Create snapshot lifecycle policy</div> <div>Delete volume</div> <div>Attach volume</div>	
<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS			
<input type="checkbox"/>	Volume1	vol-0814f90b28b4852fb	gp3	1 GiB	3000			
<input checked="" type="checkbox"/>	Volume2	vol-0d307c490de5da8af	gp3	1 GiB	3000			
<input type="checkbox"/>		vol-09b76c8bd338e72c8	gp3	8 GiB	3000			

Basic details

Volume ID

 **vol-0d307c490de5da8af (Volume2)**

Availability Zone

us-east-1d

Instance [Info](#)

i-02f252b474b3f93eb ▼



Only instances in the same Availability Zone as the selected volume are displayed.

Device name [Info](#)

/dev/sdg

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**.

Connect to the instance and run below commands for attaching volume 1:

```
# lsblk
```

```
# sudo mkdir vol1 vol2
```

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

```
# sudo mount /dev/xvdf vol1
```

```
ubuntu@ip-172-31-33-36:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0         7:0    0   24.6M  1 loop /snap/amazon-ssm-agent/7528
loop1         7:1    0   24.9M  1 loop /snap/amazon-ssm-agent/7628
loop2         7:2    0   55.7M  1 loop /snap/core18/2796
loop3         7:3    0   55.7M  1 loop /snap/core18/2812
loop4         7:4    0   63.5M  1 loop /snap/core20/2015
loop5         7:5    0  111.9M  1 loop /snap/lxd/24322
loop6         7:6    0   40.8M  1 loop /snap/snapd/20092
loop7         7:7    0   40.9M  1 loop /snap/snapd/20290
xvda        202:0    0    8G    0 disk
├─xvda1      202:1    0    7.9G  0 part /
├─xvda14     202:14   0     4M    0 part
└─xvda15     202:15   0   106M  0 part /boot/efi
xvdf         202:80   0    1G    0 disk
xvdg         202:96   0    1G    0 disk
ubuntu@ip-172-31-33-36:~$ sudo mkdir vol1 vol2
ubuntu@ip-172-31-33-36:~$ ls
vol1  vol2
```

```

ubuntu@ip-172-31-33-36:~$ sudo mkfs -t ext4 /dev/xvdf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 29c037a9-966a-49a0-a81b-0392ca72a3de
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

ubuntu@ip-172-31-33-36:~$ sudo mount /dev/xvdf vol1
ubuntu@ip-172-31-33-36:~$ █

```

Now attach Volume 2, same like volume 1 :

```
# lsblk
```

```
# sudo mkfs -t ext4 /dev/xvdg
```

```
# sudo mount /dev/xvdg vol2
```

```

ubuntu@ip-172-31-33-36:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0   24.6M  1 loop /snap/amazon-ssm-agent/7528
loop1        7:1      0   24.9M  1 loop /snap/amazon-ssm-agent/7628
loop2        7:2      0   55.7M  1 loop /snap/core18/2796
loop3        7:3      0   55.7M  1 loop /snap/core18/2812
loop4        7:4      0   63.5M  1 loop /snap/core20/2015
loop5        7:5      0  111.9M  1 loop /snap/lxd/24322
loop6        7:6      0   40.8M  1 loop /snap/snapd/20092
loop7        7:7      0   40.9M  1 loop /snap/snapd/20290
xvda        202:0     0    8G   0 disk
├─xvda1     202:1     0   7.9G   0 part /
├─xvda14    202:14    0    4M   0 part
└─xvda15    202:15    0  106M   0 part /boot/efi
xvdf        202:80    0    1G   0 disk /home/ubuntu/vol1
xvdg        202:96    0    1G   0 disk
ubuntu@ip-172-31-33-36:~$ sudo mkfs -t ext4 /dev/xvdg
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 7b561ce3-d65e-4daa-82d8-731fdb95177f
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

ubuntu@ip-172-31-33-36:~$ sudo mount /dev/xvdg vol2
ubuntu@ip-172-31-33-36:~$ █

```

```

xvda        202:0     0    8G   0 disk
├─xvda1     202:1     0   7.9G   0 part /
├─xvda14    202:14    0    4M   0 part
└─xvda15    202:15    0  106M   0 part /boot/efi
xvdf        202:80    0    1G   0 disk /home/ubuntu/vol1
xvdg        202:96    0    1G   0 disk /home/ubuntu/vol2
ubuntu@ip-172-31-33-36:~$ █

```


Step 4: Delete one volume after detaching it and extend the size of the other volume.

Volumes (1/6) Info

	Name	Volume ID	Type	Size	IOPS
<input type="checkbox"/>	Volume1	vol-0814f90b28b4852fb	gp3	1 GiB	3000
<input checked="" type="checkbox"/>	Volume2	vol-0d307c490de5da8af	gp3	1 GiB	3000
<input type="checkbox"/>	-	vol-09b76c8bd338e72c8	gp3	8 GiB	3000
<input type="checkbox"/>	-	vol-0e0032f79f8efbdfc	gp2	10 GiB	100

Actions ▲
Create volume
Modify volume
Create snapshot
Create snapshot lifecycle policy
Delete volume
Attach volume
Detach volume

Detach vol-0d307c490de5da8af?

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-0d307c490de5da8af?

Cancel Detach

✓ Successfully detached volume.

Volumes (1/6) Info

	Name	Volume ID	Type	Size	IOPS
<input type="checkbox"/>	Volume1	vol-0814f90b28b4852fb	gp3	1 GiB	3000
<input checked="" type="checkbox"/>	Volume2	vol-0d307c490de5da8af	gp3	1 GiB	3000

Actions ▲
Create volume
Modify volume
Create snapshot
Create snapshot lifecycle policy
Delete volume
Attach volume

Delete vol-0d307c490de5da8af?

After you delete a volume, its data is permanently deleted and the volume can no longer be attached to an instance.

Are you sure that you want to delete vol-0d307c490de5da8af?


Cancel Delete

- Increase the size of Volume1, Select Volume1 -----> Go to Actions -----> Modify Volume

Volumes (1/5) Info						Actions ▲	Create volume
<input type="text" value="Search"/>						Modify volume	
						Create snapshot	
						Create snapshot lifecycle policy	
						Delete volume	
<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS		
<input checked="" type="checkbox"/>	Volume1	vol-0814f90b28b4852fb	gp3	1 GiB	3000		

- Update the volume from 1 GB to 2 GB

Volume details

Volume ID
 [vol-0814f90b28b4852fb \(Volume1\)](#)

Volume type [Info](#)
 General Purpose SSD (gp3)

Size (GiB) [Info](#)

 Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS [Info](#)

 Min: 3000 IOPS, Max: 16000 IOPS. The value must be an integer.

Throughput (MiB/s) [Info](#)

 Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.

Requested volume modification for volume [vol-0814f90b28b4852fb](#).

The volume is being modified.

Volumes (1/5) [Info](#)

Search

</

```

xvda1s 202:13 0 100M 0 part /boot/efi
xvdf 202:80 0 2G 0 disk /home/ubuntu/vol1
ubuntu@ip-172-31-33-36:~$

```

Task 5: Take backup of this EBS volume

Volumes (1/5) Info						Actions ▲	Create volume
<input type="text" value="Search"/>						Modify volume	
						Create snapshot	
						Create snapshot lifecycle policy	
						Delete volume	
<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS		
<input checked="" type="checkbox"/>	Volume1	vol-0814f90b28b4852fb	gp3	2 GiB	3000		

Snapshots (2) [Info](#)

Recycle Bin

Actions

Create sn

Owned by me

Q Search

< 1

<input type="checkbox"/>	Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status
<input type="checkbox"/>	Volume1 snapshot	snap-0143e835e6bd1ec4c	2 GiB	Volume1 backup	Standard	✔ Completed
