

Contents

General Diagram	2
Prerequisites	2
Labs9: Lab Prerequisites (Using Cloudformation)	3
Lab 9c1: Create multiattach EBS and mount on 2 Instances	7
Create Multiattach EBS and attach to the instances	7
Create Filesystem, mount it to instances and create a file	10
Create Standard EBS and modifying size without instances actions	12
Evidences to send.....	17
Clean Resources	17
Detaching and deleting EBS	17
Deleting Cloudformation Stack	18
Clean resources from command line	19

General Diagram

EBS

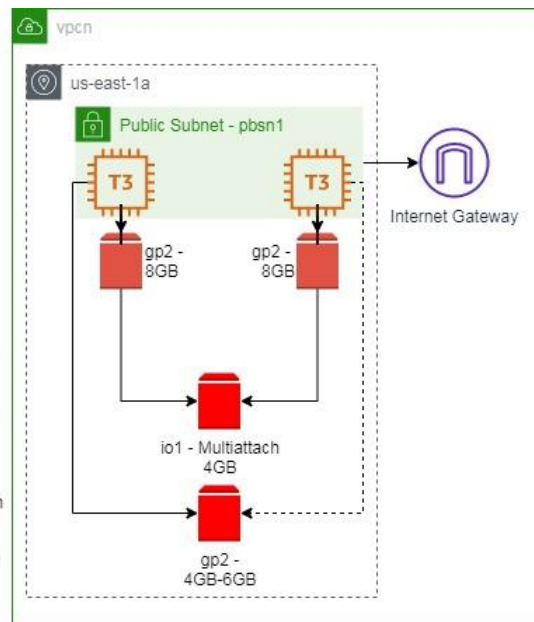
Starting an EC2 instance with a simple EBS.

After create a Multi-attach volume and synchronize on 2 Instances.

Only Nitro EC2 System is allowed, so used T3 (no-Free Tier)

Evidences:

1. Tested Read and Write from both Instances.
2. Increase the size of EBS volume without stopping.



Common Infrastructure

You can use CloudFormation to create this common infra, otherwise you can use commands that we use previously.

Create a well know infrastructure with 2 Public Instances to access them using SSH. Then create a multiattach EBS and see that its possible to access from both instances. In addition, we have to create another standard EBS and increase its size without detach from instances or unmount filesystem.

Prerequisites

Labs1c1 have to be done and the context for administrative user have to activated on Command Line Session.

Labs4c1 has the context to create infrastructure: Network (VPC, Subnets), Sec Groups and Instances.

As the instances must be T3 as minimum, the lab cost can be U\$1 or less, depends on the time that you spend.

You must create a Keypair before to launch Cloudformation, in this example, the key is ds4_kp, but it can be anything.

Labs9: Lab Prerequisites (Using Cloudformation)

Use Clouformation service to upload the JSON file, if you like read the file using a notepad or view it using CloudFormation Designer.

The screenshot displays the AWS CloudFormation console interface. At the top, the navigation bar includes links to various AWS services, with 'CloudFormation' highlighted. The main header reads 'AWS CloudFormation' and 'Model and provision all your cloud infrastructure'. A 'Create a CloudFormation stack' button is visible on the right.

The 'How it works' section explains the process: 1. Code your infrastructure using the CloudFormation template language in the YAML or JSON format, or start from many available sample templates. 2. Use AWS CloudFormation via the browser console, command line tools, or APIs to create a stack based on your template code.

The 'Create stack' wizard is shown in the 'Specify template' step. The 'Prerequisite - Prepare template' section has three options: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. The 'Specify template' section has two options: 'Amazon S3 URL' and 'Upload a template file' (selected). Below this, there is a 'Choose file' button and a 'No file chosen' message.

A file explorer window is open, showing the file 'VPC_With_PublicIPs_And_DNS' selected. The file is a JSON Source File, 18 KB in size, and was modified on 13/09/2020 03:58 a.m.

aws

Services

Resource Groups

EC2

VPC

CloudFormation

CloudFront

RDS

Route 53

IAM

CloudWatch

Elastic

CloudFormation > Stacks > Create stack

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready

☐ Use a sample template

☐ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL

☐ Upload a template file

Amazon S3 URL

Amazon S3 template URL

S3 URL:

console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/parameters

aws

Services

Resource Groups

EC2

VPC

CloudFormation

CloudFront

RDS

Route 53

IAM

CloudWatch

Elastic

CloudFormation > Stacks > Create stack

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Specify stack details

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

EC2InstanceType
EC2 instance types

KeyName
Key used for enabling EC2 KeyPair to enable SSH access to the bastion host
 A keypair that you create previously

SSHLocation
Lockdown SSH access to the bastion host (default can be accessed from anywhere)
 0.0.0.0/0 means any IP

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Key	Value	Remove
		R e m o v e

Add tag

Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name	▼	Sample-role-name	▼	Remove
---------------	---	------------------	---	--------

Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

► Stack policy

Defines the resources that you want to protect from unintentional updates during a stack update.

► Rollback configuration

Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)

► Notification options

► Stack creation options

Cancel

Previous

Next

Rollback configuration

Monitoring time

-

CloudWatch alarm ARN

-

Notification options

No notification options

There are no notification options defined

Stack creation options

Rollback on failure

Enabled

Timeout

-

Termination protection

Disabled

► Quick-create link

Cancel

Previous

Create change set

Create stack

While CloudFormation is creating the stack, see on Events and Resources where the progress and the details appear.

CloudFormation > Stacks > Lab9

Stacks (1)

Filter by stack name

Active View nested

Lab9

2020-09-13 04:01:18 UTC-0500

CREATE_IN_PROGRESS

Lab9

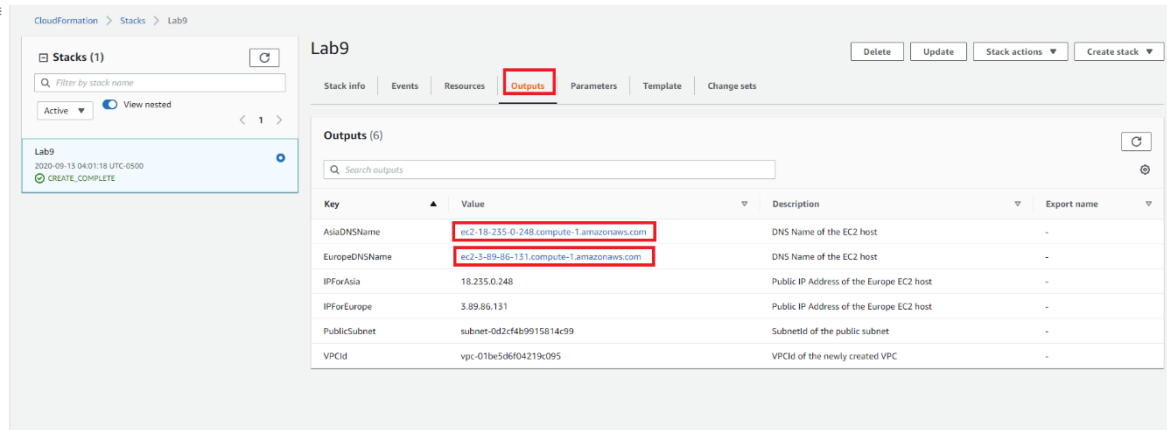
Stack info Events Resources Outputs Parameters Template Change sets

Events (5)

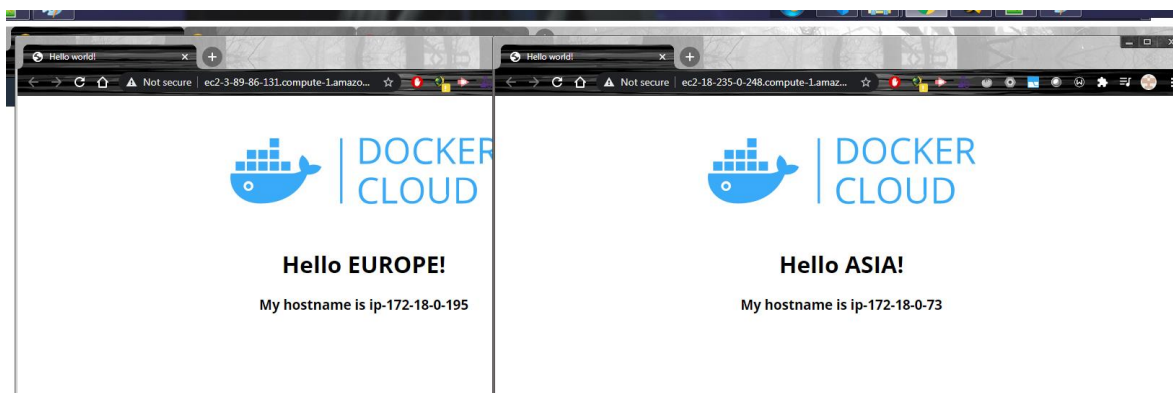
Search events

Timestamp	Logical ID	Status	Status reason
2020-09-13 04:01:24 UTC-0500	VPC	CREATE_IN_PROGRESS	Resource creation initiated
2020-09-13 04:01:24 UTC-0500	InternetGateway	CREATE_IN_PROGRESS	Resource creation initiated
2020-09-13 04:01:23 UTC-0500	VPC	CREATE_IN_PROGRESS	-
2020-09-13 04:01:23 UTC-0500	InternetGateway	CREATE_IN_PROGRESS	-
2020-09-13 04:01:18 UTC-0500	Lab9	CREATE_IN_PROGRESS	User initiated

After few minutes, the deployment is complete and the outputs appear on the right tab.

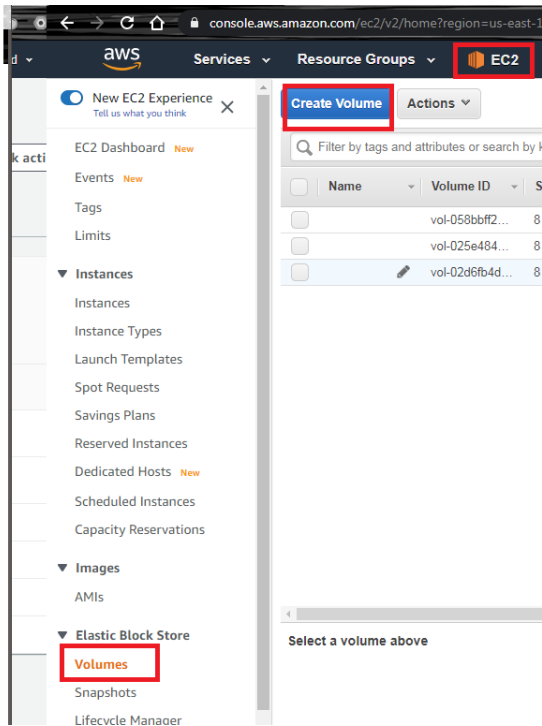


On click on the instance URLs, your browser show Docker on those instances.



Lab 9c1: Create multiattach EBS and mount on 2 Instances

Create Multiattach EBS and attach to the instances



Create Volume

Volume Type Provisioned IOPS SSD (io1) ⓘ

Size (GiB) 4 (Min: 4 GiB, Max: 16384 GiB) ⓘ

IOPS 200 (Min: 100 IOPS, Max: 64000 IOPS) ⓘ

Availability Zone* us-east-1a ⓘ

Throughput (MB/s) Not applicable ⓘ

Snapshot ID Select a snapshot ⓘ

Multi-Attach ☒ Enable ⓘ

Multi-Attach enabled volumes do not support I/O fencing. I/O fencing protocols control write access in a shared storage environment to maintain data consistency. Your applications must provide write ordering between instances to maintain data consistency on a Multi-Attach enabled volume.

Encryption ☐ Encrypt this volume

Key	(128 characters maximum)	Value	(256 characters maximum)
-----	--------------------------	-------	--------------------------

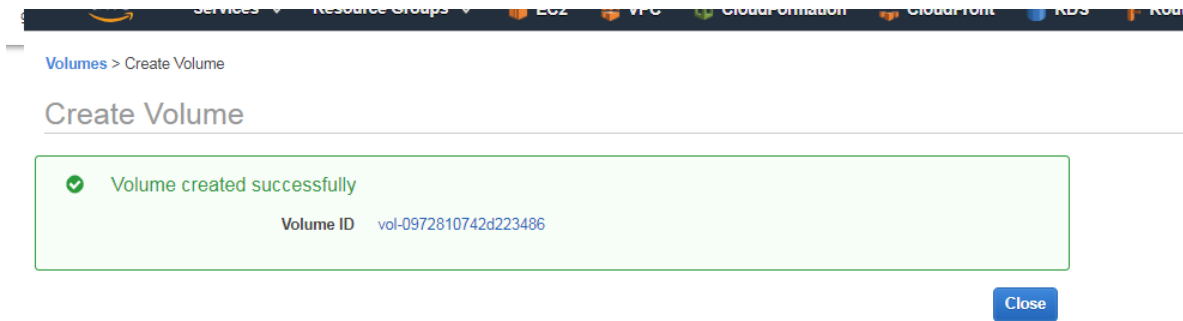
This resource currently has no tags

Choose the Add tag button or click to add a Name tag

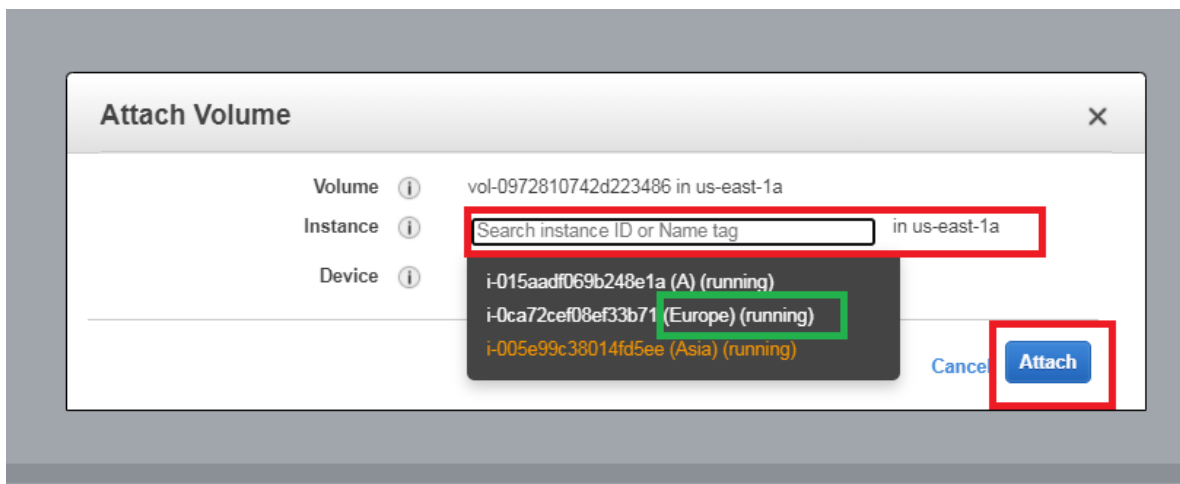
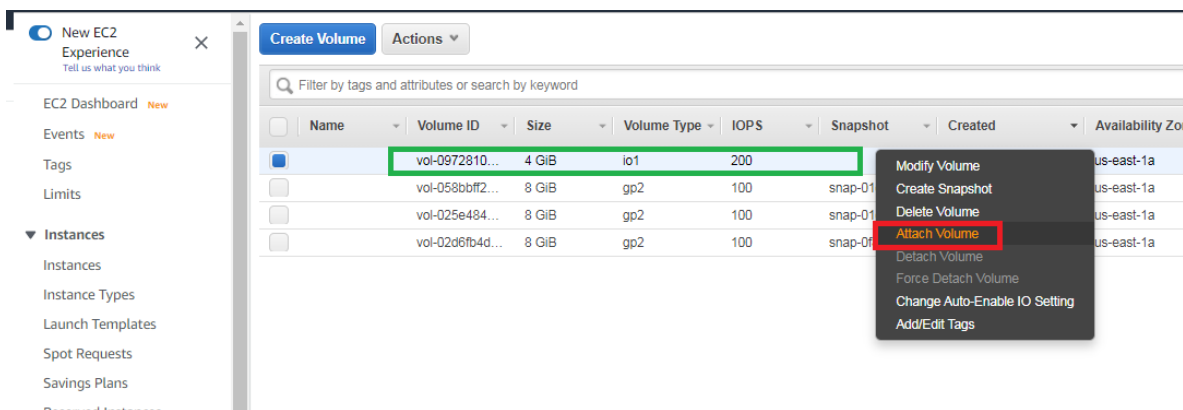
Add Tag 50 remaining (Up to 50 tags maximum)

* Required

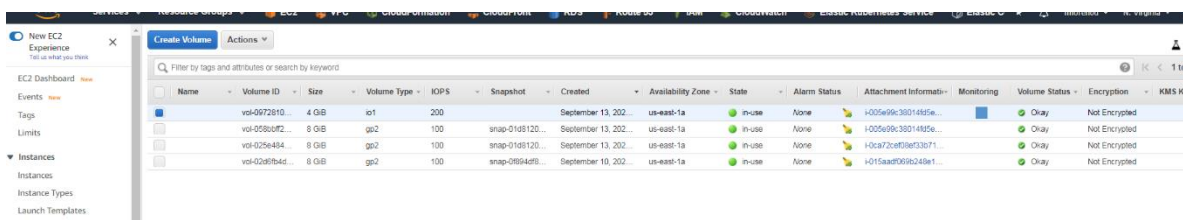
Cancel Create Volume



It's time to attach that multiattach volume to both instances, repeat those steps to both instances.



You can see the EBS status is change to in-use when you associate with one instance, at least.



Create Filesystem, mount it to instances and create a file

You can go to any instance using putty, to create the filesystem and mount it. Finally you can create a file to show on the another instance.

#Crear una carpeta para ser usado como punto de montaje

```
sudo mkdir /data
```

#Este comando comprueba que fue "attachado" a la instancia. Se verifica por tamaño la ruta del dispositivo.

```
lsblk
```

#Este comando crea el filesystem al dispositivo

```
sudo mkfs -t xfs /dev/nvme1n1
```

Este comando monta el dispositivo fisico a una carpeta de su instancia

```
sudo mount -o rw /dev/nvme1n1 /data
```

```
sudo chown ec2-user:ec2-user /data
```

```
ls -la
```

#Aqui ya se pueden crear archivos

```
echo "This a shared file using Multiattach EBS " >/data/Lab9.txt
```

```
cat /data/Lab9.txt
```

```
ec2-user@ip-172-18-0-195:/
[ec2-user@ip-172-18-0-195 ~]$ sudo mkdir /data
[ec2-user@ip-172-18-0-195 ~]$ lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme0n1     259:0    0  8G  0 disk
└─nvme0n1p1 259:1    0  8G  0 part /
└─nvme0n1p2 259:2    0 1M  0 part
nvme1n1     259:3    0 4G  0 disk
[ec2-user@ip-172-18-0-195 ~]$ sudo mkfs -t xfs /dev/nvme1n1
meta-data=/dev/nvme1n1          isize=512    agcount=4, agsize=262144 blks
                     =                       sectsz=512   attr=2, projid32bit=1
                     =                       crc=1      finobt=1, sparse=0
data        =                       bsize=4096   blocks=1048576, imaxpct=25
                     =                       sunit=0    swidth=0 blks
naming      =version 2           bsize=4096   ascii-ci=0 ftype=1
log         =internal log       bsize=4096   blocks=2560, version=2
                     =           sectsz=512   sunit=0 blks, lazy-count=1
realtime    =none                extsz=4096   blocks=0, rtextents=0
[ec2-user@ip-172-18-0-195 ~]$ sudo mount -o rw /dev/nvme1n1 /data
[ec2-user@ip-172-18-0-195 ~]$ sudo chown ec2-user:ec2-user /data
[ec2-user@ip-172-18-0-195 ~]$ ls -la
total 116
dr-xr-xr-x 26 root    root    4096 Sep 13 09:41 .
dr-xr-xr-x 26 root    root    4096 Sep 13 09:41 ..
-rw-r--r-- 1 root    root      0 Sep 13 09:02 .autofsck
-rw-r--r-- 1 root    root      0 Sep 13 09:02 .autorelabel
dr-xr-xr-x 2 root    root    4096 Sep 13 09:03 bin
dr-xr-xr-x 4 root    root    4096 Sep 13 09:03 boot
drwxr-xr-x 12 root    root    4096 Sep 13 09:04 cgroup
drwxr-xr-x 2 ec2-user ec2-user 6 Sep 13 09:41 data
drwxr-xr-x 15 root    root    2800 Sep 13 09:41 dev
drwxr-xr-x 80 root    root    4096 Sep 13 09:04 etc
drwxr-xr-x 3 root    root    4096 Sep 13 09:02 home
dr-xr-xr-x 7 root    root    4096 Nov 16 2019 lib
dr-xr-xr-x 10 root    root    12288 Sep 13 09:03 lib64
drwxr-xr-x 2 root    root    4096 Nov 16 2018 local
drwx----- 2 root    root    16384 Nov 16 2018 lost+found
drwxr-xr-x 2 root    root    4096 Jan 6 2012 media
drwxr-xr-x 2 root    root    4096 Jan 6 2012 mnt
drwxr-xr-x 4 root    root    4096 Sep 13 09:04 opt
dr-xr-xr-x 112 root   root      0 Sep 13 09:02 proc
dr-xr-xr-x 3 root    root    4096 Sep 13 09:16 root
drwxr-xr-x 4 root    root    4096 Sep 13 09:04 run
dr-xr-xr-x 2 root    root    12288 Sep 13 09:04/sbin
drwxr-xr-x 2 root    root    4096 Jan 6 2012 selinux
drwxr-xr-x 2 root    root    4096 Jan 6 2012 srv
dr-xr-xr-x 13 root    root      0 Sep 13 09:18 sys
drwxrwxrwt 4 root    root    4096 Sep 13 09:04 tmp
drwxr-xr-x 13 root    root    4096 Nov 16 2018 usr
drwxr-xr-x 19 root    root    4096 Nov 16 2018 var
[ec2-user@ip-172-18-0-195 ~]$ echo "This a shared file using Multiattach EBS" >/data/Lab9.txt
[ec2-user@ip-172-18-0-195 ~]$ cat /data/Lab9.txt
This a shared file using Multiattach EBS
[ec2-user@ip-172-18-0-195 ~]$
```

```
ec2-user@ip-172-18-0-73:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

_ _ _ _ _
|_| ( |_|_| )
_|_| \_|_|_|_|_| Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-18-0-73 ~]$ lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme0n1     259:0    0  8G  0 disk
└─nvme0n1p1 259:1    0  8G  0 part /
└─nvme0n1p2 259:2    0 1M  0 part
nvme1n1     259:3    0 4G  0 disk
[ec2-user@ip-172-18-0-73 ~]$ sudo mkdir /data
[ec2-user@ip-172-18-0-73 ~]$ sudo mount -o rw /dev/nvme1n1 /data
[ec2-user@ip-172-18-0-73 ~]$ sudo chown ec2-user:ec2-user /data
[ec2-user@ip-172-18-0-73 ~]$ cat /data/Lab9.txt
This a shared file using Multiattach EBS
[ec2-user@ip-172-18-0-73 ~]$
```

On the another instance you must mount the unit, **but not to create the filesystem.**

#Este comando comprueba que fue "attachado" a la instancia. Se verifica por tamaño la ruta del dispositivo.

lsblk

Este comando monta el dispositivo fisico a una carpeta de su instancia

sudo mount -o rw /dev/nvme1n1 /data

sudo chown ec2-user:ec2-user /data

ls -la

Create Standard EBS and modifying size without instances actions

Create a simple EBS volumen of 4GB and attach to any instance.

Create Volume

Volume Type **General Purpose SSD (gp2)** ⓘ

Size (GiB) **4** (Min: 1 GiB, Max: 16384 GiB) ⓘ

IOPS 100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS) ⓘ

Availability Zone* us-east-1a ⓘ

Throughput (MB/s) Not applicable ⓘ

Snapshot ID Select a snapshot ⓘ ⓘ

Encryption ☐ Encrypt this volume

Key (128 characters maximum) Value (256 characters maximum)

This resource currently has no tags

Choose the Add tag button or click to add a Name tag

Add Tag 50 remaining (Up to 50 tags maximum)

* Required Cancel **Create Volume**

Create Volume

✓ Volume created successfully

Volume ID **vol-0b96b38af8f4eb9e2**

Close

Volume ID	Size (GiB)	Type	IOPS	Snapshot ID	Created	Zone	Availability	Encryption	Progress	Status	Tags
vol-0b96b38...	4 GiB	gp2	100		September 13, 202...	us-east-1a	available	None		✓ Okay	Not Encrypted
vol-03a8f8f...	4 GiB	io1	200		September 13, 202...	us-east-1a	in-use	None		✓ Okay	Not Encrypted
vol-05688f2...	8 GiB	gp2	100	snap-01d8120...	September 13, 202...	us-east-1a	in-use	None		✓ Okay	Not Encrypted
vol-025e484...	8 GiB	gp2	100	snap-01d8120...	September 13, 202...	us-east-1a	in-use	None		✓ Okay	Not Encrypted
vol-02a8b4d...	8 GiB	gp2	100	snap-0f84d8f...	September 10, 202...	us-east-1a	in-use	None		✓ Okay	Not Encrypted

Attach Volume ✕

Volume ⓘ vol-0b96b38af8f4eb9e2 in us-east-1a

Instance ⓘ **i-0a72ce03ef33b71...** in us-east-1a

Device ⓘ /dev/sdg
Linux Devices: /dev/xvdf through /dev/xvdp

Note: 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.

Cancel **Attach**

You go to any instance and mount that volumen in a new mount point.

```

ec2-user@ip-172-18-0-195:/
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=ip-172-18-0-195
TERM=xterm
NAME=EUROPE
PORT=80
PROTO=TCP
LISTEN_PORT=80
HOME=/root
[ec2-user@ip-172-18-0-195 ~]$ lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme0n1     259:0    0  8G  0 disk
└─nvme0n1p1 259:1    0  8G  0 part /
└─nvme0n1p128 259:2    0  1M  0 part
nvme2n1     259:4    0  4G  0 disk
nvme1n1     259:3    0  4G  0 disk /data

[ec2-user@ip-172-18-0-195 ~]$ sudo mkfs -t xfs /dev/nvme2n1
meta-data=/dev/nvme2n1             isize=512    agcount=4, agsize=262144 blks
=                               sectsz=512    attr=2, projid32bit=1
=                               crc=1        finobt=1, sparse=0
data      =                       bsize=4096   blocks=1048576, imaxpct=25
=                               sunit=0       swidth=0 blks
naming    =version 2              bsize=4096   ascii-ci=0 ftype=1
log       =internal log          bsize=4096   blocks=2560, version=2
=                               sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                  extsz=4096   blocks=0, rtextents=0
[ec2-user@ip-172-18-0-195 ~]$ sudo mount -o rw /dev/nvme2n1 /data2
[ec2-user@ip-172-18-0-195 ~]$ sudo chown ec2-user:ec2-user /data2
[ec2-user@ip-172-18-0-195 ~]$ ls -la
total 116
dr-xr-xr-x 27 root    root    4096 Sep 13 09:54 .
dr-xr-xr-x 27 root    root    4096 Sep 13 09:54 ..
-rw-r--r-- 1 root    root      0 Sep 13 09:02 .autofsck
-rw-r--r-- 1 root    root      0 Sep 13 09:02 .autorelabel
dr-xr-xr-x 2 root    root    4096 Sep 13 09:03 bin
dr-xr-xr-x 4 root    root    4096 Sep 13 09:03 boot
drwxr-xr-x 12 root    root    4096 Sep 13 09:04 cgroup
drwxr-xr-x 2 ec2-user ec2-user 39 Sep 13 09:45 data
drwxr-xr-x 2 ec2-user ec2-user 6 Sep 13 09:55 data2
drwxr-xr-x 15 root    root   2860 Sep 13 09:54 dev
drwxr-xr-x 80 root    root    4096 Sep 13 09:04 etc
drwxr-xr-x 3 root    root    4096 Sep 13 09:02 home
dr-xr-xr-x 7 root    root    4096 Nov 16 2018 lib
dr-xr-xr-x 10 root    root   12288 Sep 13 09:03 lib64
drwxr-xr-x 2 root    root    4096 Nov 16 2018 local
drwx----- 2 root    root   16384 Nov 16 2018 lost+found
drwxr-xr-x 2 root    root    4096 Jan 6 2012 media
drwxr-xr-x 2 root    root    4096 Jan 6 2012 mnt
drwxr-xr-x 4 root    root    4096 Sep 13 09:04 opt
dr-xr-xr-x 117 root    root      0 Sep 13 09:02 proc
dr-xr-x--- 3 root    root    4096 Sep 13 09:16 root
drwxr-xr-x 4 root    root    4096 Sep 13 09:04 run
dr-xr-xr-x 2 root    root   12288 Sep 13 09:04 sbin
drwxr-xr-x 2 root    root    4096 Jan 6 2012 selinux
drwxr-xr-x 2 root    root    4096 Jan 6 2012 srv
dr-xr-xr-x 13 root    root      0 Sep 13 09:18 sys
drwxrwxrwt 4 root    root    4096 Sep 13 09:53 tmp
drwxr-xr-x 13 root    root    4096 Nov 16 2018 usr
drwxr-xr-x 19 root    root    4096 Nov 16 2018 var
[ec2-user@ip-172-18-0-195 ~]$ echo "This a shared file using Standard EBS" >/data2/Lab9.txt
[ec2-user@ip-172-18-0-195 ~]$ cat /data2/Lab9.txt
This a shared file using Standard EBS
[ec2-user@ip-172-18-0-195 ~]$

```

One is multiattach EBS
and othe is the standard
EBS

Choose the volume and modify the size.

Instances | EC2 x Volumes | EC2 x Relaxing x moun multi-a x New - Multi-A x

console.aws.amazon.com/ec2/v2/home?region=us-east-1#Volumes:sort=desc:createTime

aws Services Resource Groups EC2 VPC CloudFormation

New EC2 Experience Tell us what you think

EC2 Dashboard New

Events New

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Create Volume Actions

Filter by tags and attributes or search by keyword

	Name	Volume ID	Size	Volume Type	IOPS
<input checked="" type="checkbox"/>		vol-0b96b38...	4 GiB		
<input type="checkbox"/>		vol-03a6fc8f...	4 GiB		
<input type="checkbox"/>		vol-058bbff2...	8 GiB		
<input type="checkbox"/>		vol-025e484...	8 GiB		
<input type="checkbox"/>		vol-02d6fb4d...	8 GiB		

Modify Volume

Create Snapshot

Delete Volume

Attach Volume

Detach Volume

Force Detach Volume

Change Auto-Enable IO Setting

Add/Edit Tags

Modify Volume

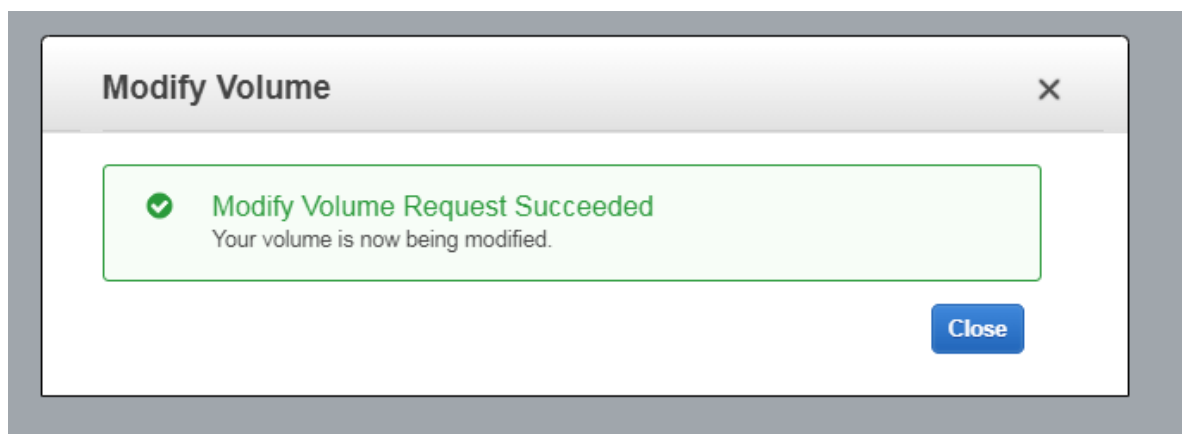
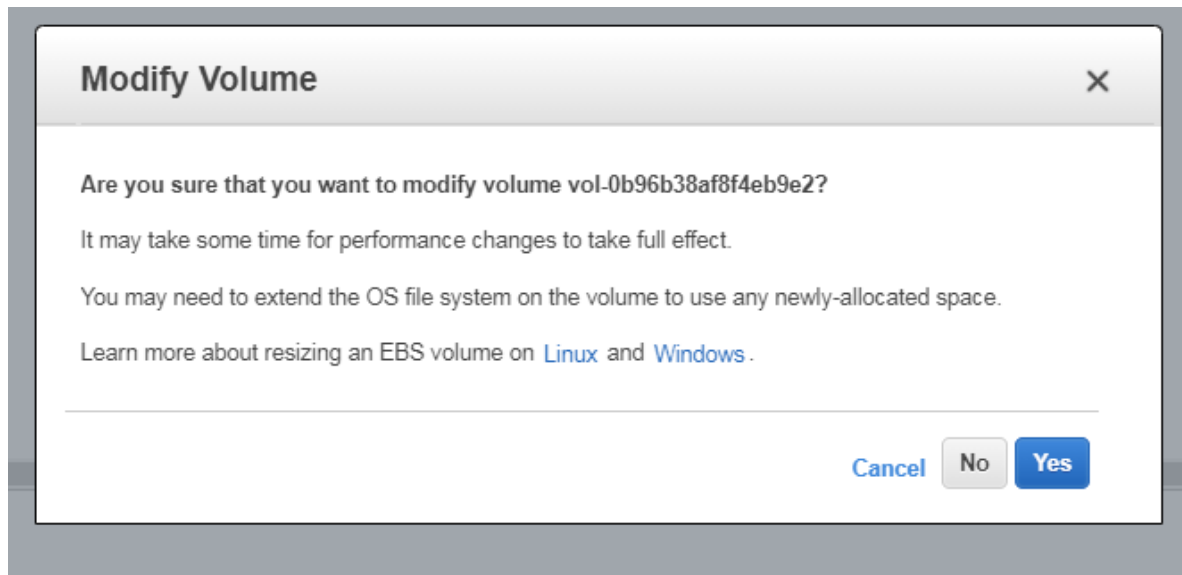
Volume ID vol-0b96b38af8f4eb9e2

Volume Type General Purpose SSD (gp2) ⓘ

Size 6 (Min: 1 GiB, Max: 16384 GiB) ⓘ

IOPS 100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS) ⓘ

Cancel Modify



Show the change on the instance.

```

ec2-user@ip-172-18-0-195:~$ df -h
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1        259:0    0   8G  0 disk 
└─nvme0n1p1    259:1    0   8G  0 part /
└─nvme0n1p128  259:2    0   1M  0 part 
nvme2n1        259:4    0   4G  0 disk 
nvme1n1        259:3    0   4G  0 disk /data
[ec2-user@ip-172-18-0-195 ~]$ sudo mkdir /data2
[ec2-user@ip-172-18-0-195 ~]$ sudo mkfs -t xfs /dev/nvme2n1
meta-data=/dev/nvme2n1             isize=512    agcount=4, agsize=262144 blks
      =                       sectsz=512    attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=0
data      =                       bsize=4096   blocks=1048576, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2              bsize=4096   ascii-ci=0 ftype=1
log       =internal log          bsize=4096   blocks=2560, version=2
      =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                  extsz=4096   blocks=0, rtextents=0
[ec2-user@ip-172-18-0-195 ~]$ sudo mount -o rw /dev/nvme2n1 /data2
[ec2-user@ip-172-18-0-195 ~]$ sudo chown ec2-user:ec2-user /data2
[ec2-user@ip-172-18-0-195 ~]$ ls -la
total 116
dr-xr-xr-x  27 root    root    4096 Sep 13 09:54 .
dr-xr-xr-x  27 root    root    4096 Sep 13 09:54 ..
-rw-r--r--   1 root    root         0 Sep 13 09:02 .autofsck
-rw-r--r--   1 root    root         0 Sep 13 09:02 .autorelabel
dr-xr-xr-x   2 root    root    4096 Sep 13 09:03 bin
dr-xr-xr-x   4 root    root    4096 Sep 13 09:03 boot
drwxr-xr-x  12 root    root    4096 Sep 13 09:04 cgroup
drwxr-xr-x   2 ec2-user ec2-user   39 Sep 13 09:45 data
drwxr-xr-x   2 ec2-user ec2-user   6 Sep 13 09:55 data2
drwxr-xr-x  15 root    root    2860 Sep 13 09:54 dev
drwxr-xr-x  80 root    root    4096 Sep 13 09:04 etc
drwxr-xr-x   3 root    root    4096 Sep 13 09:02 home
dr-xr-xr-x   7 root    root    4096 Nov 16 2018 lib
dr-xr-xr-x  10 root    root   12288 Sep 13 09:03 lib64
drwxr-xr-x   2 root    root    4096 Nov 16 2018 local
drwx-----  2 root    root   16384 Nov 16 2018 lost+found
drwxr-xr-x   2 root    root    4096 Jan 6 2012 media
drwxr-xr-x   2 root    root    4096 Jan 6 2012 mnt
drwxr-xr-x   4 root    root    4096 Sep 13 09:04 opt
dr-xr-xr-x  117 root    root         0 Sep 13 09:02 proc
dr-xr-x---   3 root    root    4096 Sep 13 09:16 root
drwxr-xr-x   4 root    root    4096 Sep 13 09:04 run
dr-xr-xr-x   2 root    root   12288 Sep 13 09:04 sbin
drwxr-xr-x   2 root    root    4096 Jan 6 2012 selinux
drwxr-xr-x   2 root    root    4096 Jan 6 2012 srv
dr-xr-xr-x  13 root    root         0 Sep 13 09:18 sys
drwxrwxrwt   4 root    root    4096 Sep 13 09:53 tmp
drwxr-xr-x  13 root    root    4096 Nov 16 2018 usr
drwxr-xr-x  19 root    root    4096 Nov 16 2018 var
[ec2-user@ip-172-18-0-195 ~]$ echo "This a shared file using Standard EBS" >/data2/Lab9.txt
[ec2-user@ip-172-18-0-195 ~]$ cat /data2/Lab9.txt
This a shared file using Standard EBS
[ec2-user@ip-172-18-0-195 ~]$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1        259:0    0   8G  0 disk 
└─nvme0n1p1    259:1    0   8G  0 part /
└─nvme0n1p128  259:2    0   1M  0 part 
nvme2n1        259:4    0   6G  0 disk /data2
nvme1n1        259:3    0   4G  0 disk /data
[ec2-user@ip-172-18-0-195 ~]$ cat /data2/Lab9.txt
This a shared file using Standard EBS
[ec2-user@ip-172-18-0-195 ~]$

```

To check if the instance expand this hard disk, you have to use

```
df -hT
```

```
sudo xfs_growfs -d /data2
```


df -hT

You have more information on https://docs.aws.amazon.com/ebs/latest/userguide/recognize-expanded-volume-linux.html?icmpid=docs_ec2_console

Evidences to send

To have a review, the student has to send some screenshots to instructor email:

1. Putty Screenshot of both instances with Multiattach EBS mounted and the shared file. It's similar to the last picture of [Create Filesystem, mount it to instances and create a file](#)
2. Putty screenshot to visualize size change without instance stopping or remount it. It's similar to the last picture of [Create Standard EBS and modifying size without instances actions.](#)

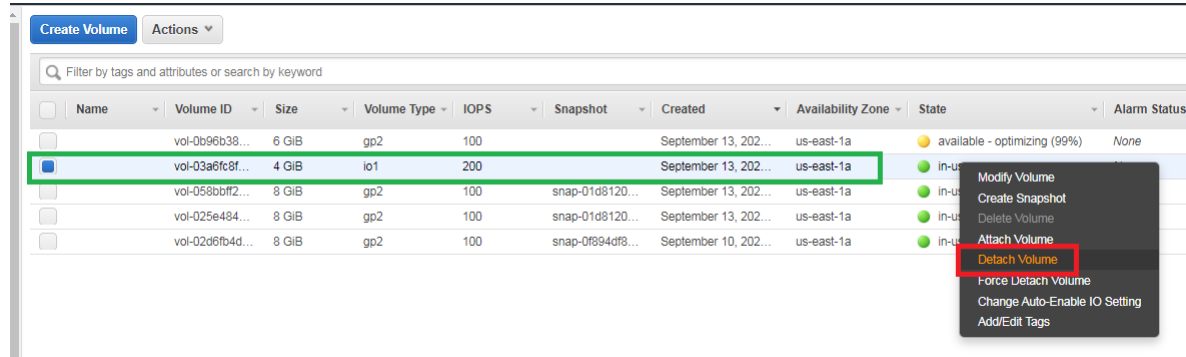
Clean Resources

Detaching and deleting EBS

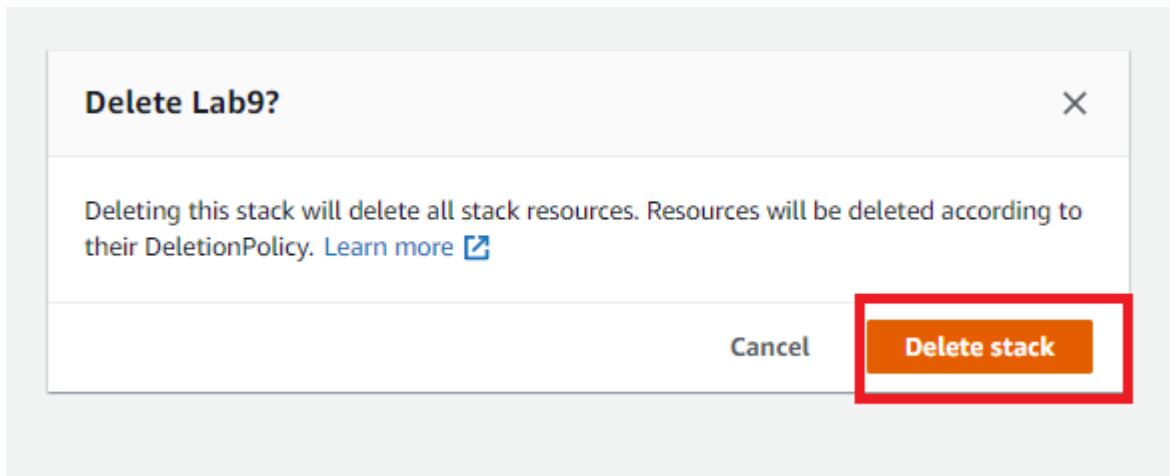
Para borrar un volumen, primero se desmonta desde el sistema operativo usando

```
rem Quitando el punto de montaje desde Linux, umount /data
sudo umount <punto de montaje>
```

Luego, se va a la



Cuando es multiattach se selecciona la instancia, mientras que para el Standard EBS es por defecto.



Clean resources from command line

```
rem ----- ELIMINAR RECURSOS -----  
aws ec2 terminate-instances --instance-ids <Codigo de las Instancias>  
aws ec2 delete-security-group --group-id %SecGroup_A_Id%  
aws ec2 detach-internet-gateway --internet-gateway-id %IGW_Id% --vpc-id %vpcn_Id%  
aws ec2 delete-internet-gateway --internet-gateway-id %IGW_Id%  
aws ec2 delete-subnet --subnet-id %pbsn1_Id%  
aws ec2 delete-route-table --route-table-id %Public_RT_Id%  
aws ec2 delete-vpc --vpc-id %vpcn_Id%  
aws ec2 delete-key-pair --key-name Lab10a
```