



Amazon S3 - AWS Elastic Block Store (EBS)

**Indicaciones (EL DESARROLLO DE
EBS ESTÁ EN LA PARTE DE ABAJO,
EN SU PROPIA SECCIÓN, YA QUE
NO PUDE ELIMINAR EL
ENCABEZADO)**

1. Las respuestas deben ser explicadas, solo colocar resultados sin ninguna referencia no puntúa en las preguntas de la evaluación.
2. Realiza una copia de este documento y coloca todas tus respuestas y sube a tu repositorio personal de github en formato markdown. Presenta capturas de pantalla del procedimiento y las explicaciones necesarias. No puntúa si solo se hace la presentación de imágenes.
3. De preferencia adiciona un video adicional explicando los pasos realizados. Utiliza el sandbox de AWS usado en la práctica anterior.
4. Sube a la plataforma de Blackboard el enlace de github donde están todas tus respuestas. No olvides colocar tu nombre y apellido antes de subir el enlace de tus respuestas a la plataforma
5. Cualquier evidencia de copia elimina el examen se informará de la situación a la coordinación.

S3

En este laboratorio, se estudiará el almacenamiento de Amazon S3. Utilizarás los comandos `aws s3` y `s3api` para administrar datos en Amazon S3. Amazon S3 es un almacenamiento de objetos accesible a través de Internet.

Parte 1: Operaciones básicas con S3

Suponga que su directorio actual es `/home/aws_user` (puedes cambiarlo). Envíe las siguientes instrucciones y responde las preguntas que siguen.

1. Enumere todos los buckets propiedad del usuario a través del siguiente comando

```
ls. aws s3 ls
```

¿Cuál es la salida?

2. Haz un bucket a través del siguiente comando
`mb. aws s3 mb s3://tu_nombre_de_usuario`



¿Cuál es la salida?

3. Enumera el contenido del bucket a través del siguiente comando ls.



```
aws s3 ls s3://tu_nombre_de_usuario
```

¿Cuál es la salida?

4. Crea un directorio llamado páginas web (mkdir webpages) y cd en ese directorio. Crea un archivo html simple llamado hello.html con el siguiente contenido.

```
<html><body>
<h1>Amazon
S3</h1> Hello World!
</body></html>
```

Carga el archivo en tu bucket s3 y póngalo a disposición del público con lo siguiente.

```
aws s3 cp hello.html s3://tu_nombre_de_usuario --acl public-read
```

¿Cuál es la salida?

5. Dado que se puede acceder a tu objeto s3 a través de Internet, probémoslo. En el navegador web de tu máquina virtual (u otra) accede a la URL http://s3.amazonaws.com/tu_nombre_de_usuario/hello.html. ¿Qué viste en el navegador?

Parte 2: alojamiento de sitios web estáticos con S3

6. Podemos usar el bucket como almacenamiento de sitios web estáticos. Experimentamos con eso aquí. Crea dos archivos html en el directorio actual llamados index.html y error.html. El contenido de los dos archivos se muestra a continuación.

```
<html><body>
```

```
This is an index page!
```

```
</body></html>
```

```
<html><body>
```

```
Sorry, we can't find that page!
```

```
</body></html>
```



El comando sync compara el directorio de origen con tu bucket S3 y carga solo archivos nuevos o modificados. Entonces puedes cargar ambos archivos fácilmente a través del siguiente comando.

```
aws s3 sync ./ s3://tu_nombre_de_usuario/ --acl public-read
```

¿Cuál es la salida? Ahora habilitamos el bucket para alojamiento de sitios web estáticos con las siguientes instrucciones.

```
aws s3 website s3://tu_nombre_de_usuario/  
--index-document index.html  
--error-document error.html
```

Observa cómo la instrucción enlaza ambos archivos con sus usos. En el navegador web de tu VM, acceda a la URL

http://tu_nombre_de_usuario.s3-website-us-east-1.amazonaws.com/

¿Qué viste en el navegador? ¿Por qué? Ahora, acceda a

http://tu_nombre_de_usuario.s3-website-us-east-1.amazonaws.com/hello.html

¿Qué viste en el navegador? A continuación, acceda a

http://tu_nombre_de_usuario.s3-website-us-east-1.amazonaws.com/2.html. ¿Qué viste en el navegador? ¿Por qué?

- Podemos definir reglas de redirección y agregar metadatos a los objetos en el bucket. Ejecuta el siguiente comando para hacerlo. Observa que este comando usa s3api, no s3.

```
aws s3api put-object --bucket tu_nombre_de_usuario  
--key hello.html  
--website-redirect-location http://www.nku.edu/~haow1 --acl public-read  
--metadata redirection_creator=aws_user
```

Ahora http://tu_nombre_de_usuario.s3-website-us-east-1.amazonaws.com/hello.html

¿Qué ves en el navegador? ¿Por qué?

- Para recuperar los metadatos de un objeto, usamos el subcomando head-object. Emite la siguiente instrucción.

```
aws s3api head-object --bucket tu_nombre_de_usuario --key hello.html
```

¿Cuál es la salida?

Parte 3: Limpieza

- Podemos eliminar objetos usando rm. Elimina tu página de índice de la siguiente manera.



aws s3 rm s3://tu_nombre_de_usuario/index.html



¿Cuál es la salida?

10. Y podemos quitar el bucket como un todo. Usa lo

siguiente. `aws s3 rb s3://tu_nombre_de_usuario --force`

¿Cuál es la salida? ¿Qué hace `--force`?

EBS

En este laboratorio, se utilizará la CLI de AWS para crear un volumen y una instantánea de Amazon EBS y configurar tu almacenamiento de EBS como un arreglo RAID.

Parte 1. Crea un nuevo volumen de EBS

1. Inicia sesión en el sandbox del curso. Crea un nuevo volumen de EBS con el siguiente comando.

```
aws ec2 create-volume --size 1 --region us-east-1  
--availability-zone us-east-1c
```

¿Qué significa este comando? ¿Cuál es la salida?

Al colocar 'aws ec2' estamos especificando el servicio de Amazon que vamos a utilizar. Después, el 'create-volume' se coloca para hacer referencia a que vamos a crear un volumen. Para ello, se adicionarán los parámetros '--size, --region, --availability-zone'. "--size" hace referencia al tamaño del disco (en GiB); "--region" a la región en donde vamos a crear dicho volumen y "--availability-zone" a la zona de disponibilidad. Colocando el comando en consola y ejecutándolo:

```
AWS CloudShell
us-east-1
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 create-volume --size 1 --region us-east-1 --availability-zone u
s-east-1c
{
  "AvailabilityZone": "us-east-1c",
  "CreateTime": "2023-06-19T19:52:50+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-066480a7afd20cb98",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
[cloudshell-user@ip-10-2-115-242 ~]$
```

La salida vendrían a ser las características del volumen que acabamos de crear. Se muestra,



por ejemplo, la zona de disponibilidad donde está ubicado, su ID de volumen, entre otros.

2. Utiliza el siguiente comando para ver la información de tu volumen de EBS donde se te proporcionó volume_id en el resultado del comando anterior.

aws ec2 describe-volume-status --volume-ids volume_id

¿Cuál es la salida?

Colocamos y ejecutamos el comando en consola:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 describe-volume-status --volume-ids vol-066480a7afd20cb98
{
  "VolumeStatuses": [
    {
      "Actions": [],
      "AvailabilityZone": "us-east-1c",
      "Events": [],
      "VolumeId": "vol-066480a7afd20cb98",
      "VolumeStatus": {
        "Details": [
          {
            "Name": "io-enabled",
            "Status": "passed"
          },
          {
            "Name": "io-performance",
            "Status": "not-applicable"
          }
        ]
      },
      "Status": "ok"
    }
  ]
}
```

Como salida podemos ver la información de volumen, pero más detallada. Se muestra, de igual manera, la zona de disponibilidad, su ID y el 'status' (estado).

3. Para crear una instancia de EBS, hazlo siguiente.

**aws ec2 run-instances --image-id ami-d9a98cb0 --count 1 --instance-type t1.micro
--key-name tu_nombre_de_usuario-key --security-groups tu_nombre_de_usuario
--placement AvailabilityZone=us-east-1c**

Para crear una instancia, previamente debemos haber definido el nombre de par de claves así como el grupo de seguridad. Sin embargo, como recién estamos creando una instancia, tendremos que crear previamente ambos (el par de claves y grupo de seguridad) para poder lanzar la instancia. Para crear el par de claves, lo realizaré mediante la misma página de AWS, ya que de esta manera se descargará, ni bien terminemos de crear el par de claves, el documento con extensión .PPK para poder realizar la conexión por SSH mediante el programa 'puTTY'. Entonces, creamos el par de claves:



Par de claves

Un par de claves, compuesto por una clave privada y una clave pública, es un conjunto de credenciales de seguridad que se utilizan para demostrar su identidad cuando se conecta a una instancia.

Nombre

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves [Información](#)

☒ RSA ☐ ED25519

Formato de archivo de clave privada

☐ .pem
Para usar con OpenSSH

☒ .ppk
Para usar con **Putty**

Etiquetas: *opcional*

No hay etiquetas asociadas a este recurso.

[Agregar nueva etiqueta](#)

Puede agregar hasta 50 etiquetas más.

[Cancelar](#) [Crear par de claves](#)

Y después, el grupo de seguridad correspondiente (habilitando el puerto 22 que hace referencia a SSH):

EC2 > ... > Crear grupo de seguridad

Regla de entrada 1

[Eliminar](#)

Tipo [Información](#)

SSH

Intervalo de puertos [Información](#)

22

Origen [Información](#)

0.0.0.0/0

[Agregar regla](#)

Protocolo [Información](#)

TCP

Tipo de origen [Información](#)

Anywhere-IPv4

Descripción: *opcional* [Información](#)

Reglas de salida [Información](#)

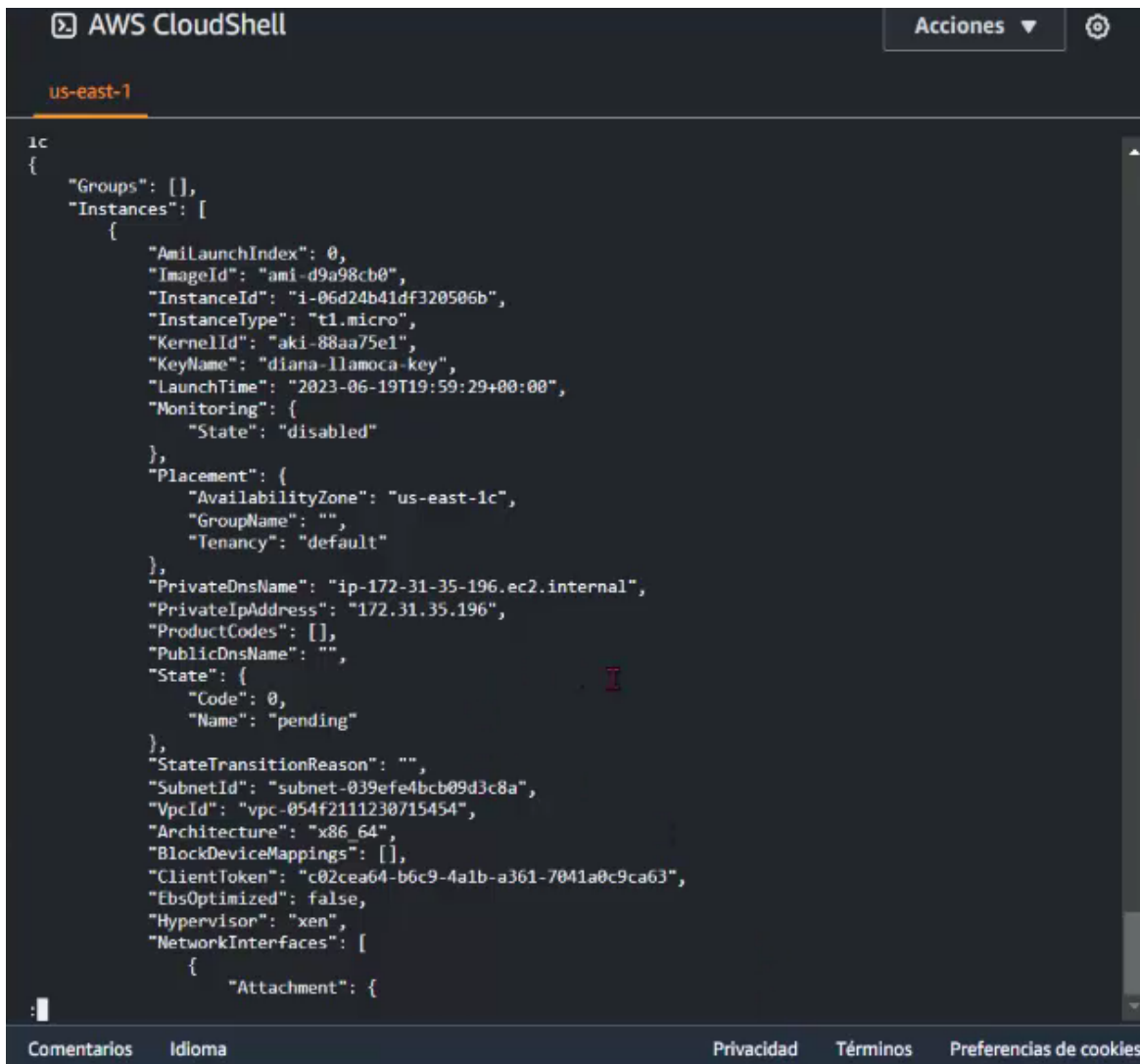
Regla de salida 1

[Eliminar](#)

Una vez creado el par de claves, así como el grupo de seguridad, procedemos a lanzar la instancia mediante consola:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 run-instances --image-id ami-d9a98cb0 --count 1 --instance-type t1.micro --key-name diana-llamoca-key --security-groups diana-llamoca --placement AvailabilityZone=us-east-1c
```

Y lo que carga por consola después de colocar dicho comando es el siguiente:



```
AWS CloudShell
us-east-1

{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-d9a98cb0",
      "InstanceId": "i-06d24b41df320506b",
      "InstanceType": "t1.micro",
      "KernelId": "aki-88aa75e1",
      "KeyName": "diana-llamoca-key",
      "LaunchTime": "2023-06-19T19:59:29+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1c",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-35-196.ec2.internal",
      "PrivateIpAddress": "172.31.35.196",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-039efe4bcb09d3c8a",
      "VpcId": "vpc-054f2111230715454",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "c02cea64-b6c9-4a1b-a361-7041a0c9ca63",
      "EbsOptimized": false,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
```

Ahora, adjunta el volumen de EBS a la instancia. Esto lo colocas en el directorio /dev/sdf en tu instancia EC2.

```
aws ec2 attach-volume --volume-id volume_id --instance-id id_instance
--device /dev/sdf
```

¿Cuál es la salida?

Adjuntamos el volumen de EBS a la instancia:

```
Attachment : {  
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 attach-volume --volume-id vol-066480a7afd20cb98 --instance-id i-06d24b41df320506b --device /dev/sdf
```

Para ello, haremos uso del comando 'attach-volume'. Y la salida que se muestra mediante consola es la siguiente:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 attach-volume --volume-id vol-066480a7afd20cb98 --instance-id i-06d24b41df320506b --device /dev/sdf  
{  
  "AttachTime": "2023-06-19T20:01:48.725000+00:00",  
  "Device": "/dev/sdf",  
  "InstanceId": "i-06d24b41df320506b",  
  "State": "attaching",  
  "VolumeId": "vol-066480a7afd20cb98"  
}
```

Lo que muestra la salida es el ID de la instancia (InstanceId) a la que estamos adjuntando el volumen con el ID que se muestra en pantalla (VolumeId). El 'state'(estado) es 'attaching' (puesto que recién hemos adjuntado la instancia con el volumen).

4. Inicia sesión en la instancia EC2 a través de ssh. En tu instancia EC2, cambie a root. Ahora queremos crear un sistema de archivos en el volumen de EBS (el volumen de EBS es básicamente un dispositivo de almacenamiento en blanco). Luego necesitamos montar el volumen para que sea accesible. Utiliza los siguientes comandos desde tu EC2. Ten en cuenta que, según el controlador del dispositivo de bloque del kernel, el dispositivo puede estar conectado con un nombre diferente al que ha especificado. Por ejemplo, si especificas un nombre de dispositivo de /dev/sdf, el kernel podría cambiar el nombre de tu dispositivo a /dev/xvdf, en la mayoría de los casos, la letra final sigue siendo la misma. Ejecuta lsblk en tu terminal para ver tus dispositivos de disco disponibles y tus puntos de montaje (si corresponde) para ayudarte a determinar el nombre de dispositivo correcto que debe usar. Suponga que el kernel cambia el nombre del dispositivo a /dev/xvdf.

mkfs -F /dev/xvdf

¿Cuál es la salida?

Para ello, previamente debemos ingresar a la instancia (los pasos para iniciar sesión en nuestra instancia EC2 a través de ssh se muestra en el vídeo). Sin embargo, de manera resumida, podemos decir mediante el documento que para iniciar sesión en la instancia, debemos abrir el programa puTTY y cargar el par de claves que se descargó inicialmente cuando creamos uno. Después, realizamos la conexión, dando click en 'open' y automáticamente estaremos en nuestra instancia.

Una vez que estemos dentro, colocaremos el comando 'mkfs -F /dev/xvdf' (ya que estamos suponiendo que el kernel va a cambiar el nombre del dispositivo al nuevo que le estamos asignando mediante ese comando). Obviamente, antes de colocar este comando, debemos logearnos como usuario root, y ello se realizará colocando el comando 'sudo su' mediante la terminal de nuestra instancia:

```
root@ip-172-31-35-196: /home/ubuntu  
Run 'do-release-upgrade' to upgrade to it.  
*** /dev/xvda1 will be checked for errors at next reboot ***  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-35-196:~$ lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  
xvda1 202:1 0 8G 0 disk /  
xvdf 202:80 0 1G 0 disk  
ubuntu@ip-172-31-35-196:~$ mkfs -F /dev/xvdf  
mkfs.ext2 1.42 (29-Nov-2011)  
mkfs.ext2: Permission denied while trying to determine filesystem size  
ubuntu@ip-172-31-35-196:~$ sudo su  
root@ip-172-31-35-196:/home/ubuntu# mkfs -F /dev/xvdf
```



Después de colocar ese comando, como salida obtenemos lo siguiente:

```
root@ip-172-31-35-196:/home/ubuntu# mkfs -F /dev/xvdf
mke2fs 1.42 (29-Nov-2011)
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
Writing superblocks and filesystem accounting information: done

root@ip-172-31-35-196:/home/ubuntu#
```

```
mkdir /data
mount /dev/xvdf /data
cd /data/
df
¿Cuál es la salida?
```

Colocando dichos comandos en el terminal de la instancia:

```
root@ip-172-31-35-196:/home/ubuntu# mkdir /data
root@ip-172-31-35-196:/home/ubuntu# mount /dev/xvdf /data
root@ip-172-31-35-196:/home/ubuntu# cd /data/
root@ip-172-31-35-196:/data# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/xvda1      8256952  914708   6922816  12% /
udev            264876      8    264868    1% /dev
tmpfs           109256     164    109092    1% /run
none            5120        0      5120     0% /run/lock
none            273140      0    273140    0% /run/shm
/dev/xvdf       1032088    1284    978376    1% /data
root@ip-172-31-35-196:/data#
```

Como salida, podemos ver que se ha montado el disco al directorio /data.

Esto sucede porque al colocar 'mkdir /data', estamos creando un directorio que actuará como punto de montaje. Después, cuando se especifica 'mount /dev/xvdf /data' estamos



diciendo que se monte el disco en ese directorio que actuará como punto de montaje.

Al colocar 'cd /data/' lo que estamos haciendo es ingresar a dicho directorio, para ahí colocar 'df' y se muestre la información respectiva de los discos montados. (EN LA OTRA PÁGINA ESTÁ EL OTRO PASO, POR EL ENCABEZADO DE PÁGINA ES QUE APARECEN ESTOS SALTOS)



Parte 2. Instantáneas de EBS

5. Crea un archivo llamado `aws_user.txt` y escribe lo que desees en el archivo. Ahora, veremos cómo crear una copia de seguridad de todo tu volumen de EBS. El primer paso es asegurarte de que todos los datos en memoria se hayan escrito en el volumen (disco), ya que es posible que el archivo creado aún no se haya guardado en el disco. Para forzar que esto suceda, usamos el comando `sync` (sincronización). En la ventana de tu terminal para su instancia EC2, ejecuta las siguientes instrucciones.

```
root@ip-10-45-185-154:/data# sync
```

Para ello, creamos un archivo llamado `'aws_user.txt'` mediante el comando `'cat'` en la terminal de la instancia:

```
root@ip-172-31-35-196:/data# cat > aws_user.txt
Practica 4 - Diana Llamoca
```

Después, coloqué aquí el comando `'df'` para verificar que el archivo que he creado haya sido correctamente colocado (puesto que el tamaño del disco debería de disminuir al agregar un archivo nuevo). Es por ello que una vez que haya verificado que sí se colocó, entonces procedo a escribir el comando `'sync'` para asegurarme de que todos los datos en memoria se hayan escrito en el disco:

```
root@ip-172-31-35-196:/data# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/xvda1      8256952 914708   6922816  12% /
udev            264876    8       264868   1% /dev
tmpfs           109256    164     109092   1% /run
none            5120      0       5120    0% /run/lock
none            273140    0       273140   0% /run/shm
/dev/xvdf       1032088   1288     978372   1% /data
root@ip-172-31-35-196:/data# sync
root@ip-172-31-35-196:/data# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/xvda1      8256952 914708   6922816  12% /
udev            264876    8       264868   1% /dev
tmpfs           109256    164     109092   1% /run
none            5120      0       5120    0% /run/lock
none            273140    0       273140   0% /run/shm
/dev/xvdf       1032088   1288     978372   1% /data
root@ip-172-31-35-196:/data#
```

Abre una segunda ventana de terminal en tu máquina virtual. Emite el siguiente comando.

```
aws ec2 create-snapshot --volume-id volume_id
--description "Esta es mi instantánea de volumen".
```

donde `volume_id` es el id obtenido del paso 1. ¿Cuál es el resultado?

Para ello, colocamos en la consola de AWS el comando que se indica para poder crear una instantánea a partir del volumen principal:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 create-snapshot --volume-id vol-066480a7afd20cb98 --description "Esta es mi instantanea de volumen"
{
  "Description": "Esta es mi instantanea de volumen",
  "Encrypted": false,
  "OwnerId": "423514121265",
  "Progress": "",
  "SnapshotId": "snap-0971b95cdf2af62d2",
  "StartTime": "2023-06-19T20:11:10.621000+00:00",
  "State": "pending",
  "VolumeId": "vol-066480a7afd20cb98",
  "VolumeSize": 1,
  "Tags": []
}
```

Como salida, podemos ver que se muestran automáticamente los detalles de la instantánea que acabamos de crear. Como por ejemplo, la 'descripción' que hemos colocado al momento de crearla. Podemos ver, además, el ID de la snapshot (así como el ID del volumen a partir del cual hemos creado dicha instantánea) y demás características de dicha snapshot.

Puedes verificar el estado de tu instantánea usando las siguientes instrucciones.

aws ec2 describe-snapshots --snapshot-id snapshot_id

El snapshot_id debe ser parte de la salida de la instrucción de creación de instantáneas que acaba de ejecutar. ¿Cuál es el resultado del comando describe-snapshot? Continúa repitiendo este comando hasta que vea que el estado de la instantánea cambia a "completado", lo que significa que se ha realizado una copia de seguridad del volumen.

Para verificar el estado de la instantánea, colocamos el comando en la consola de AWS:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 describe-snapshots --snapshot-id snap-0971b95cdf2af62d2
{
  "Snapshots": [
    {
      "Description": "Esta es mi instantanea de volumen",
      "Encrypted": false,
      "OwnerId": "423514121265",
      "Progress": "100%",
      "SnapshotId": "snap-0971b95cdf2af62d2",
      "StartTime": "2023-06-19T20:11:10.621000+00:00",
      "State": "completed",
      "VolumeId": "vol-066480a7afd20cb98",
      "VolumeSize": 1,
      "StorageTier": "standard"
    }
  ]
}
```

Como resultado al colocar dicho comando, podemos ver que nos sale la información de la snapshot que acabamos de crear, como su descripción, snapshot, el ID del volumen a partir del cual fue creada, el tamaño del disco (se ve 1GB) y demás.



6. Dada una instantánea, podemos usarla para crear un nuevo volumen. Ejecuta el siguiente comando. Utiliza el ID de instantánea del paso 5.

aws ec2 create-volume --region us-east-1

--availability-zone us-east-1c

--snapshot-id snapshot_id

¿Cuál es la salida?

Para crear un volumen a partir de la instantánea, agregamos el parámetro '--snapshot-id' con su respectivo ID de la instantánea. Así que colocamos el comando en la CLI de Amazon y lo ejecutamos:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 create-volume --region us-east-1 --availability-zone us-east-1c --snapshot-id snap-0971b95cdf2af62d2
{
  "AvailabilityZone": "us-east-1c",
  "CreateTime": "2023-06-19T20:13:55+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "snap-0971b95cdf2af62d2",
  "State": "creating",
  "VolumeId": "vol-0809085b6b74c082c",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
```

La salida que se muestra en pantalla son las características del nuevo volumen que se ha creado a partir de la instantánea. Podemos ver el ID del nuevo volumen, la zona de disponibilidad, el ID de la snapshot a partir de la cual fue creada y demás.

Comprueba el estado del volumen. ¿Qué comando ejecutaste para verificar el estado? ¿Cuál es la salida?

Para comprobar el estado del volumen, colocamos el comando 'aws ec2 describe-snapshots --snapshot-id (id_de_la_instantánea)'. De esta manera (gráficamente):

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 describe-snapshots --snapshot-id snap-0971b95cdf2af62d2
{
  "Snapshots": [
    {
      "Description": "Esta es mi instantanea de volumen",
      "Encrypted": false,
      "OwnerId": "423514121265",
      "Progress": "100%",
      "SnapshotId": "snap-0971b95cdf2af62d2",
      "StartTime": "2023-06-19T20:11:10.621000+00:00",
      "State": "completed",
      "VolumeId": "vol-066480a7afd20cb98",
      "VolumeSize": 1,
      "StorageTier": "standard"
    }
  ]
}
```



Como salida, podemos ver la descripción del nuevo volumen (su ID, ID de la snapshot a partir de la cual fue creado, el tamaño del volumen, etc). (EN LA OTRA PÁGINA, ESTÁ EL OTRO PASO. ESTOS SALTOS SON POR LOS ENCABEZADOS DE PÁGINA)



7. Repite el comando de adjuntar volumen del paso 3 para adjuntar este nuevo volumen. El ID de volumen será el que se devolvió al obtener el estado 6, mientras que el ID de instancia es el de tu instancia EC2 que obtuvo en el paso 3.

`aws ec2 attach-volume --volume-id volume_id`

`--instance-id instance_id --device /dev/sdg`

¿Cuál es la salida?

Para atar el nuevo volumen a la instancia EC2, haremos uso del comando 'attach':

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 attach-volume --volume-id vol-0809085b6b74c082c --instance-id i-06d24b41df320506b --device /dev/sdg
{
  "AttachTime": "2023-06-19T20:16:34.230000+00:00",
  "Device": "/dev/sdg",
  "InstanceId": "i-06d24b41df320506b",
  "State": "attaching",
  "VolumeId": "vol-0809085b6b74c082c"
}
```

Como salida, podemos ver el ID de la instancia a la cual le hemos asignado dicho volumen y el ID del volumen que estamos asignando. En este caso, obviamente le tenemos que asignar el volumen nuevo que fue creado a partir de la instantánea. En otras palabras, nuestra instancia ahora tiene 2 volúmenes adjuntados.

8. Vuelve a la ventana de la terminal en la que se tiene ssh en tu instancia EC2. Desde ese terminal, crea un punto de montaje llamado /data2 y monte el nuevo volumen allí. ¿Qué comandos se ejecutó para lograr ambas tareas? Cambia el directorio de su instancia EC2 a /data2. ¿Viste el archivo aws_user.txt?

Ahora, lo primero que realizaremos es crear un directorio llamado /data2. Ese directorio cumplirá el papel de punto de montaje. Después, para montar el volumen, tenemos que colocar el comando 'mkfs -F /dev/xvdf' pues el device del nuevo volumen que hemos creado es /dev/sdg.

```
root@ip-172-31-35-196:/data# mkdir /data2
root@ip-172-31-35-196:/data# mkfs -F /dev/xvdf
mke2fs 1.42 (29-Nov-2011)
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
Writing superblocks and filesystem accounting information: done

root@ip-172-31-35-196:/data#
```

Una vez realizado ello, procederemos a montar el volumen mediante el comando 'mount /dev/xvdg /data2', donde especificamos el volumen que queremos montar y además el punto de montaje que estará representado mediante el directorio. Colocamos 'df' para ver si ya se realizó el montaje correctamente:

```
root@ip-172-31-35-196:/data# mount /dev/xvdg /data2
root@ip-172-31-35-196:/data# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/xvda1      8256952  914720   6922804  12% /
udev            264876      8    264868    1% /dev
tmpfs           109256     168    109088    1% /run
none            5120        0      5120     0% /run/lock
none            273140      0    273140    0% /run/shm
/dev/xvdf       1032088    1288    978372    1% /data
/dev/xvdg       1032088    1284    978376    1% /data2
```

Y podemos ver que el disco que le hemos especificado (/dev/xvdg) ya se ha montado al punto de montaje '/data2' (en el vídeo también se explica)

9. Ahora queremos desmontar nuestros volúmenes, para lo cual usamos el comando umount. Luego separaremos los volúmenes de la instancia EC2 y los destruiremos. Los siguientes son los comandos a ejecutar. Ten en cuenta que los primeros tres comandos están en su instancia EC2 y el resto está en tu VM.

```
root@ip-10-45-185-154:/data3# cd /
```

```
root@ip-10-45-185-154:/# umount /dev/xvdf
```

```
root@ip-10-45-185-154:/# umount /dev/xvdg
```

Ahora desconecta y elimina el primer volumen, cuyo volume_id obtuvo en el paso 1. Espera unos 10 segundos después de desconectar antes de intentar eliminar.

```
aws ec2 detach-volume --volume-id volume_id aws
```

```
ec2 delete-volume --volume-id volume_id
```

¿Cuáles son las salidas?

Primero, salimos del directorio que usamos como punto de montaje con 'cd ..' Después, para desmontar los volúmenes, haremos uso del comando 'umount' (le especificamos el /dev correspondiente asociado a cada uno de los volúmenes) en el terminal de la instancia:

```
root@ip-172-31-35-196:/data2# cd ..
root@ip-172-31-35-196:/# umount /dev/xvdf
root@ip-172-31-35-196:/# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/xvda1      8256952  914724   6922800  12% /
udev            264876      8    264868    1% /dev
tmpfs           109256     168    109088    1% /run
none            5120        0      5120     0% /run/lock
none            273140      0    273140    0% /run/shm
/dev/xvdg       1032088    1284    978376    1% /data2
root@ip-172-31-35-196:/# umount /dev/xvdg
root@ip-172-31-35-196:/# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/xvda1      8256952  914724   6922800  12% /
udev            264876      8    264868    1% /dev
tmpfs           109256     168    109088    1% /run
none            5120        0      5120     0% /run/lock
none            273140      0    273140    0% /run/shm
```

Una vez hecho esto, regresamos a la consola de AWS y ejecutamos el comando 'detach' y después 'delete' para eliminar el volumen:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 detach-volume --volume-id vol-066480a7afd20cb98
{
  "AttachTime": "2023-06-19T20:01:48+00:00",
  "Device": "/dev/sdf",
  "InstanceId": "i-06d24b41df320506b",
  "State": "detaching",
  "VolumeId": "vol-066480a7afd20cb98"
}
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 delete-volume --volume-id vol-066480a7afd20cb98
```

De esta manera, ya habremos eliminado el volumen principal (puesto que dicho ID le corresponde) (esto también está explicado en el vídeo). Las salidas que se muestran, vendría a ser el estado, que antes de eliminar dicho volumen estará en 'detaching', puesto que si no detenemos primero, no se podrá eliminar. También podemos ver el ID del volumen al cual le estamos aplicando dicho comando con su respectiva.

Repite estos dos comandos para el segundo volumen, cuyo volume_id deberías haber obtenido del paso 6. ¿Qué comandos usastes?

¿Cuáles son las salidas?

Ahora, repetimos los mismos comandos para el segundo volumen (el volumen que fue creado a partir de la instantánea):

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 detach-volume --volume-id vol-0809085b6b74c082c
{
  "AttachTime": "2023-06-19T20:16:34+00:00",
  "Device": "/dev/sdg",
  "InstanceId": "i-06d24b41df320506b",
  "State": "detaching",
  "VolumeId": "vol-0809085b6b74c082c"
}
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 delete-volume --volume-id vol-0809085b6b74c082c
```

Los comandos que usé vendrían a ser el 'detach' (especificándole el ID del otro volumen, claro está). Y también 'delete'. De esta manera, ya habremos eliminado el segundo volumen.

10. Elimina la instantánea con lo siguiente usando su snapshot_id del paso 5. `aws ec2 delete-snapshot --snapshot-id snapshot_id`}

Para eliminar la instantánea, haremos uso del comando 'aws ec2 delete-snapshot --snapshot-id (id_de_la_instantánea)', como se muestra en la imagen:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 delete-snapshot --snapshot-id snap-0971b95cdf2af62d2
```

(En el vídeo puede verse con mejor claridad el comando)

De esta manera, ya hemos eliminado también la snapshot que creamos



anteriormente.

11. Cambie a la terminal. De lo que aprendiste en la parte 1, crea dos volúmenes de 1 GB en la zona de disponibilidad us-east-1c. ¿Qué comandos ejecutaste? ¿Cuáles son las salidas? Adjunta ambos volúmenes a tu instancia EC2, haciendo que aparezcan como /dev/sdh1 y /dev/sdh2, respectivamente. ¿Qué comandos ejecutaste? ¿Cuáles son las salidas?

Entonces, para crear ambos volúmenes, colocamos los siguientes comandos:

```
aws ec2 create-volume --size 1 --region us-east-1 --availability-zone us-east-1c
```

```
aws ec2 create-volume --size 1 --region us-east-1 --availability-zone us-east-1c
```

Es decir, estamos creando 2 volúmenes con 1 GB de espacio y ambos están ubicados en la zona de disponibilidad 'us-east-1c'.

Si colocamos ambos comandos en la consola de Amazon, obtendremos lo siguiente como salida:

```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 create-volume --size 1 --region us-east-1 --availability-zone us-east-1c
{
  "AvailabilityZone": "us-east-1c",
  "CreateTime": "2023-06-19T20:34:50+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-085ab1045534cdcb2",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 create-volume --size 1 --region us-east-1 --availability-zone us-east-1c
{
  "AvailabilityZone": "us-east-1c",
  "CreateTime": "2023-06-19T20:35:21+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-0a1a89068fe96e457",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
```

Como salida, vemos la información de cada uno de los volúmenes que hemos creado. Como por ejemplo, sus zonas de disponibilidad, sus respectivos ID's, el tamaño de volumen, etc.

Ahora, procederemos a adjuntar ambos volúmenes a la instancia EC2 que tenemos:



```
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 attach-volume --volume-id vol-085ab1045534cdcb2 --instance-id i-06d24b41df320506b --device /dev/sdh1
{
  "AttachTime": "2023-06-19T20:38:11.313000+00:00",
  "Device": "/dev/sdh1",
  "InstanceId": "i-06d24b41df320506b",
  "State": "attaching",
  "VolumeId": "vol-085ab1045534cdcb2"
}
[cloudshell-user@ip-10-2-115-242 ~]$ aws ec2 attach-volume --volume-id vol-0a1a89068fe96e457 --instance-id i-06d24b41df320506b --device /dev/sdh2
{
  "AttachTime": "2023-06-19T20:39:36.190000+00:00",
  "Device": "/dev/sdh2",
  "InstanceId": "i-06d24b41df320506b",
  "State": "attaching",
  "VolumeId": "vol-0a1a89068fe96e457"
}
```

Para ello, pues hemos usado el comando 'attach' a cada uno de los volúmenes para adjuntarlo a la instancia, con sus respectivos **/dev/sdh1** y **/dev/sdh2**. Las salidas vendrían a ser el 'device' correspondiente a cada uno de los volúmenes que hemos asignado mediante consola, que ahora pues es lo que se muestra en la pantalla y en la CLI de Amazon. Además, podemos ver el ID de los volúmenes correspondientes, así como a la instancia en la cual estamos adjuntando ambos volúmenes.

12. Cambia al terminal de la instancia EC2. Usaremos el programa mdadm de Linux para configurar los volúmenes en una configuración RAID. Instala mdadm de la siguiente manera.

apt-get update

apt-get install mdadm

Escribe "y" y presiona enter cuando se te solicite, seleccione "No configuration" cuando se te solicite y presiona enter. Ahora ejecutamos mdadm para crear un arreglo RAID 0 en los dos volúmenes. Ejecuta lo siguiente. Donde vea "renamed_/dev/sdh1" y "renamed_/dev/sdh2", usa los nombres que se te proporcionó AWS en el paso 11.

mdadm --create /dev/md0 --level 0 --metadata=1.1

--raid-devices 2 renamed_/dev/sdh1 renamed_/dev/sdh2

¿Cuál es la salida?

Aquí tuve errores al instalar el programa 'mdadm'. Intenté probar también otros comandos para ver si es que podía instalarlo de otro modo, pero pues no me logró salir y es por ello que me quedé en el paso 12 (está en el vídeo).



13. Ahora, podemos comprobar el estado de la matriz RAID 0. Emite lo siguiente. mdadm --detail /dev/md0

¿Cuál es la salida? Tenemos que agregar un sistema de archivos al arreglo RAID 0. Entonces queremos montarlo. Haz lo siguiente.

mkfs /dev/md0

mkdir /data3

mount /dev/md0 /data3

El comando df de Linux muestra información sobre los sistemas de archivos montados. ¿Cuál es la salida?

14. Finalizamos este laboratorio deteniendo el arreglo RAID 0, separando y eliminando ambos volúmenes de EBS y luego finalizando la instancia EC2. Para detener el arreglo RAID 0, haz lo siguiente desde su instancia EC2.

cd /

umount /dev/md0

mdadm --stop /dev/md0

Ahora, cambia a tu terminal. Separa y elimina ambos volúmenes de EBS. ¿Qué comandos ejecutaste? ¿Cuáles son las salidas? Finaliza tu instancia EC2. ¿Qué comando ejecutaste? ¿Cuál es la salida?