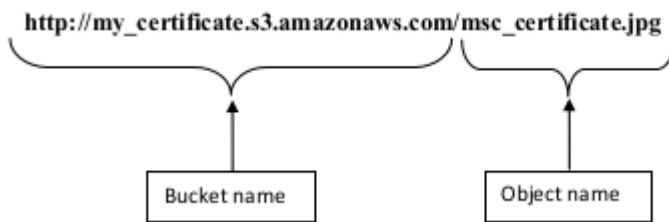


S3:

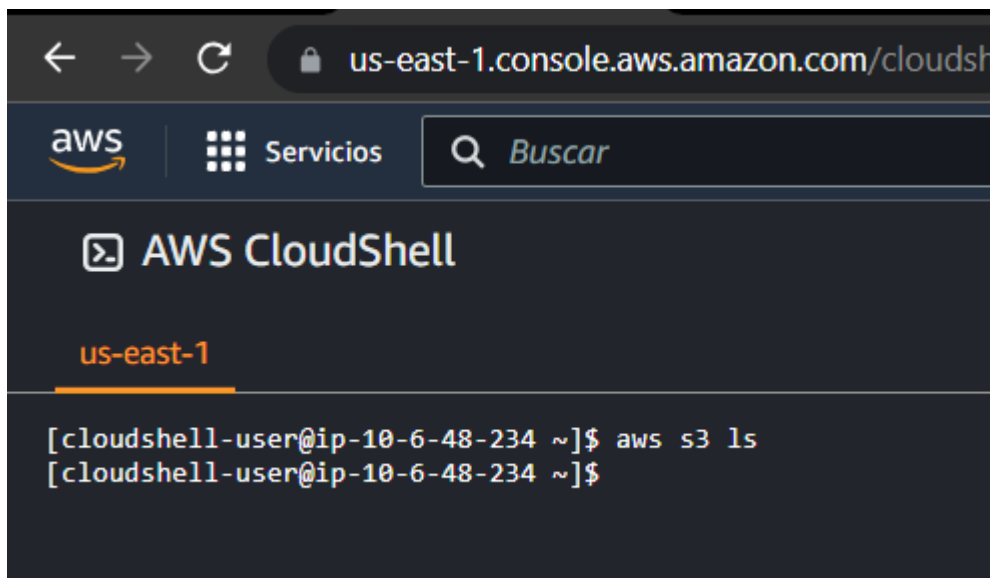
En este laboratorio, se estudiará el almacenamiento de Amazon S3. Utilizarás los comandos `aws s3` y `s3api` para administrar datos en Amazon S3. Amazon S3 es un almacenamiento de objetos accesible a través de Internet



Parte 1: Operaciones básicas con S3

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

1. Enumere todos los buckets propiedad del usuario a través del siguiente comando `ls`. ¿Cuál es la salida? **Se usa para mostrar los archivos y directorios guardados en el S3**



2. Haz un bucket a través del siguiente comando `mb`
`aws s3 mb s3://tu_nombre_de_usuario` ¿Cuál es la salida? **Este comando me permite crear un bucket, puedo usar cualquier nombre para mi bucket**

```
us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1

aws Servicios Buscar [A]

AWS CloudShell

us-east-1

[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 ls
[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 mb s3://stephtoribio
make_bucket: stephtoribio
[cloudshell-user@ip-10-6-48-234 ~]$
```

3. Enumera el contenido del bucket a través del siguiente comando `ls`
aws s3 ls s3://tu_nombre_de_usuario, ¿Cuál es la salida? **Muestra el bucket creado**

```
aws Servicios Buscar

AWS CloudShell

us-east-1

[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 ls
[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 mb s3://stephtoribio
make_bucket: stephtoribio
[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 ls s3://stephtoribio
[cloudshell-user@ip-10-6-48-234 ~]$
```

- Si deseamos ver el el bucket, se coloca el comando **aws s3 ls**

```
us-east-1

[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 ls
[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 mb s3://stephtoribio
make_bucket: stephtoribio
[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 ls s3://stephtoribio
[cloudshell-user@ip-10-6-48-234 ~]$ aws s3 ls
2023-06-19 18:38:58 stephtoribio
```

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>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?

- En la línea de comandos se usa echo para crear el archivo hello.html y su contenido

```
[cloudshell-user@ip-10-6-48-234 ~]$ mkdir "webpages"
[cloudshell-user@ip-10-6-48-234 ~]$ cd "webpages"
[cloudshell-user@ip-10-6-48-234 webpages]$ echo '<html><body><h1>Amazon S3</h1>Hello World!</body></html>' > hello.html
[cloudshell-user@ip-10-6-48-234 webpages]$
```

- Se verifica que el archivo creado se encuentre en el bucket

```
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 cp hello.html s3://stephtoribio
upload: ./hello.html to s3://stephtoribio/hello.html
```

- Para que este a disposición del público el contenido del bucket se deben otorgar ciertos permisos

Editar el bloqueo de acceso público (configuración del bucket)

Info

Bloquear acceso público (configuración del bucket)

Se concede acceso público a buckets y objetos a través de listas de control de acceso (ACL), políticas de bucket, políticas de puntos de acceso o todas las anteriores. A fin de garantizar que se bloquee el acceso público a todos sus buckets y objetos de S3, active Bloquear todo acceso público. Esta configuración se aplica en exclusiva a este bucket y a sus puntos de acceso. AWS recomienda activar Bloquear todo acceso público pero, antes de aplicar cualquiera de estos ajustes, asegúrese de que sus aplicaciones funcionarán correctamente sin acceso público. Si necesita cierto nivel de acceso público a sus buckets u objetos, puede personalizar los valores de configuración individuales a continuación para que se ajusten mejor a sus necesidades específicas de almacenamiento. [Más información](#)

☐ Bloquear *todo* el acceso público

Activar esta configuración equivale a activar las cuatro opciones que aparecen a continuación. Cada uno de los siguientes ajustes son independientes entre sí.

☐ Bloquear el acceso público a buckets y objetos concedido a través de *nuevas* listas de control de acceso (ACL)

S3 bloqueará los permisos de acceso público aplicados a objetos o buckets agregados recientemente, y evitará la creación de nuevas ACL de acceso público para buckets y objetos existentes. Esta configuración no cambia los permisos existentes que permiten acceso público a los recursos de S3 mediante ACL.

☐ Bloquear el acceso público a buckets y objetos concedido a través de *cualquier* lista de control de acceso (ACL)

S3 ignorará todas las ACL que conceden acceso público a buckets y objetos.

☐ Bloquear el acceso público a buckets y objetos concedido a través de políticas de bucket y puntos de acceso *públicas nuevas*

S3 bloqueará las nuevas políticas de bucket y puntos de acceso que concedan acceso público a buckets y objetos. Esta

Política de bucket

La política del bucket, escrita en JSON, proporciona acceso a los objetos almacenados en el bucket. Las políticas de bucket no se aplican a los objetos que pertenecen a otras cuentas. [Más información](#)

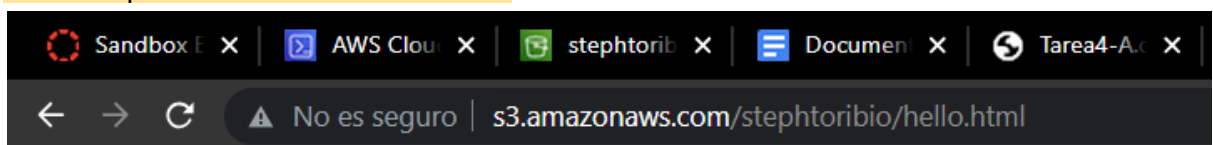
[Editar](#) [Eliminar](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::stephtoribio/*"
    }
  ]
}
```

[Copiar](#)

```
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 cp hello.html s3://stephtoribio/hello.html
upload: ./hello.html to s3://stephtoribio/hello.html
[cloudshell-user@ip-10-6-48-234 webpages]$
```

5. Dado que se puede acceder a tu objeto s3 a través de Internet, probémoslo. En el navegador web de tu máquina virtual (u otra9 accede a la URL **http://s3.amazonaws.com/tu_nombre_de_usuario/hello.html**. ¿Qué viste en el navegador? Los permisos otorgados anteriormente permiten que se pueda visualizar de forma pública el contenido el bucket



Amazon S3

Hello World!

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.

- Con echo creo el contenido de los dos archivos y con el comando **aws s3 cp** archivo **s3://stephtoribio/** copio el archivo hacia el bucket

```
[cloudshell-user@ip-10-6-48-234 webpages]$ echo '<html><body>This is an index page!</body></html>' > index.html
[cloudshell-user@ip-10-6-48-234 webpages]$ echo '<html><body>Sorry, we can\'t find that page!</body></html>' > error.html
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 cp index.html s3://stephtoribio/
upload: ./index.html to s3://stephtoribio/index.html
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 cp error.html s3://stephtoribio/
upload: ./error.html to s3://stephtoribio/error.html
[cloudshell-user@ip-10-6-48-234 webpages]$
```

```

[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 sync ./ s3://stephtoribio/ --acl public-read
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 website s3://stephtoribio/ --index-document index.html --error-document error.html
[cloudshell-user@ip-10-6-48-234 webpages]$

```

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? Muestra el contenido del archivo indice.html

Ahora, acceda a

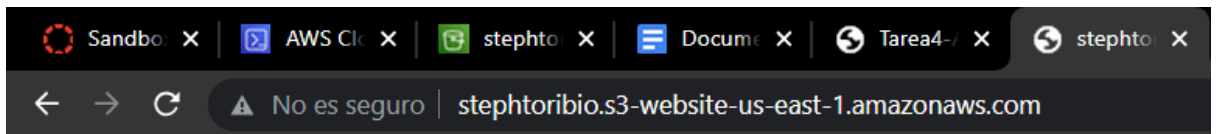
http://tu_nombre_de_usuario.s3-website-us-east-1.amazonaws.com/ hello.html

¿Qué viste en el navegador? Recordemos que en un inicio en hello.html decía Hello World!, ahora la url redirige a la de indice.html porque se enlazo indice y error para que sean sitios web estáticos

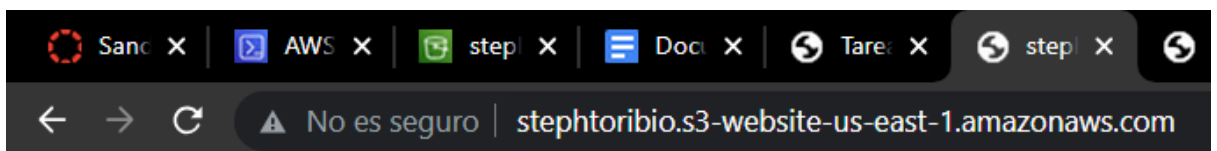
A continuación, acceda a

http://tu_nombre_de_usuario.s3-website-us-east-1.amazonaws.com/2.html.

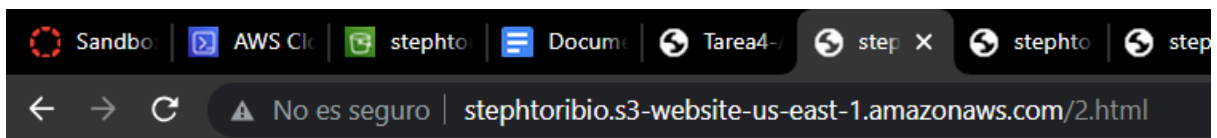
¿Qué viste en el navegador? ¿Por qué? Muestra el contenido de error.html



This is an index page!



This is an index page!



Sorry, we can't find that page!

- 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. Uso s3api porque las funciones que ahora deseo realizar no se pueden abastecer mediante s3 y esto permite interactuar directamente con la API de 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**

Por ejemplo, con s3api puedo modificar los metadatos del objeto en S3

```
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3api put-object --bucket stephtoribio --key hello.html --website-redirect-location http://www.nku.edu/~haow1 --metadata redirection_creator=aws_user
{
  "ETag": "\"d41d8cd98f00b204e9800998ecf8427e\"",
  "ServerSideEncryption": "AES256"
}
[cloudshell-user@ip-10-6-48-234 webpages]$
```

Ahora

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

¿Qué ves en el navegador? ¿Por qué?. Con el comando anteriormente realizado se redirige hacia <http://www.nku.edu/~haow1> el contenido del bucket



Wei Hao, Ph.D.

MSCS and MSCIT Program Director
Professor
[Department of Computer Science](#)
[Northern Kentucky University](#)
Highland Heights, KY 41099
Email: haow1@nku.edu
Office: GH 518
Phone: (859)572-5659

Research Interests

Cloud Computing, Mobile Computing, Web Technologies (Web Caching, Web Services, Content Distribution, Content Adaptation, and Web-based Systems)

[Publications and Grants](#)

Teaching

[Industry Experience](#)

[Professional Activities](#)

[Awards and Honors](#)

8. 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? Permite recuperar los metadatos del objeto, que se redirigió hacia nku

```
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3api head-object --bucket stephtoribio --key hello.html
{
  "AcceptRanges": "bytes",
  "LastModified": "2023-06-19T19:45:49+00:00",
  "ContentLength": 0,
  "ETag": "\"d41d8cd98f00b204e9800998ecf8427e\"",
  "ContentType": "binary/octet-stream",
  "WebsiteRedirectLocation": "http://www.nku.edu/~haow1",
  "ServerSideEncryption": "AES256",
  "Metadata": {
    "redirection_creator": "aws_user"
  }
}
```

Parte 3: Limpieza

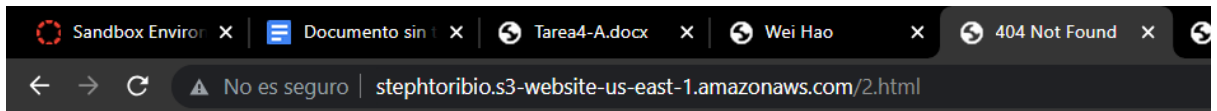
9. Podemos eliminar objetos usando **rm**. Elimina tu página de índice de la siguiente manera. **aws s3 rm s3://tu_nombre_de_usuario/index.html** Se eliminó el archivo de índice

```
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 rm s3://stephtoribio/index.html
delete: s3://stephtoribio/index.html
[cloudshell-user@ip-10-6-48-234 webpages]$
```

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? Elimina un bucket forzosamente, incluyendo los archivos contenidos, se observa que primero se eliminan los objetos y por último el bucket

```
[cloudshell-user@ip-10-6-48-234 webpages]$ aws s3 rb s3://stephtoribio --force
delete: s3://stephtoribio/error.html
delete: s3://stephtoribio/hello.html
remove_bucket: stephtoribio
```

- El contenido de la página web ahora:



404 Not Found

- Code: NoSuchBucket
- Message: The specified bucket does not exist
- BucketName: stephtoribio
- RequestId: 653SM1HC3FAENVAT
- HostId: ck61pfObPktBD2LuLVLZGGNL2vT5xq2ndb/EJ4HdtFPSHCMwEamAyo1ll/vj7bQYu44+55P/cKM=

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? Crea un volumen con un tamaño específico en la zona de disponibilidad 1c

```
us-east-1

[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 create-volume --size 1 --region us-east-1 --availability-zone us-east-1c
{
  "AvailabilityZone": "us-east-1c",
  "CreateTime": "2023-06-19T22:03:26+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-0f2c251bc2c127054",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
```

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? El comando permite visualizar los detalles del volumen creado de la misma forma que se observa en la interfaz

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 describe-volume-status --volume-ids "vol-0f2c251bc2c127054"
{
  "VolumeStatuses": [
    {
      "Actions": [],
      "AvailabilityZone": "us-east-1c",
      "Events": [],
      "VolumeId": "vol-0f2c251bc2c127054",
      "VolumeStatus": {
        "Details": [
          {
            "Name": "io-enabled",
            "Status": "passed"
          },
          {
            "Name": "io-performance",
            "Status": "not-applicable"
          }
        ],
        "Status": "ok"
      }
    }
  ]
}
```

- El estado del volumen está como disponible

<input type="checkbox"/>	-	vol-0f2c251bc2c127054	gp2	1 GiB	100
2023/06/19 17:03 GMT-5		us-east-1c		🟢 Disponible	

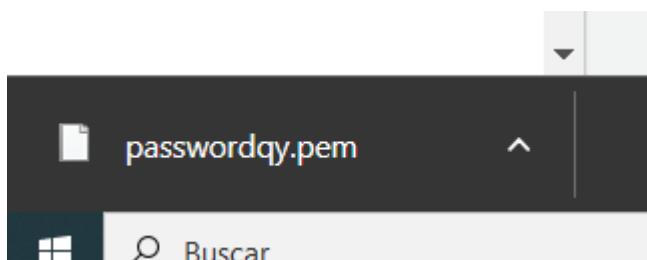
3. Para crear una instancia de EBS, hazlo siguiente.

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 Se está creando una instancia EC2 con determinados valores, pero antes de que se pueda realizar este comando se deben crear el grupo de seguridad y configurar el par de claves

- He creado el par de clave con el tipo rsa y con pem para que permita el acceso a SSH

- Se observa que se descargó el par de clave

<input type="checkbox"/>	passwordqy	rsa	2023/06/19 17:03 GMT-5	64:ef:dc:73:fc:7b:2f:ad:a6:cd:f6:b7:ea:...	key-0c4dcf5025a44918e
--------------------------	------------	-----	------------------------	--	-----------------------



- Creó el grupo de seguridad para que esto y el par de claves permita el acceso a SSH

sg-03300d9d9ee41d0c9 - SecurityEc2

Acciones ▼

Detalles

Nombre del grupo de seguridad SecurityEc2	ID del grupo de seguridad sg-03300d9d9ee41d0c9	Descripción Permite acceso a SSH	ID de la VPC vpc-0359a0ad11ac69833
Propietario 769139452531	Número de reglas de entrada 1 Entrada de permiso	Número de reglas de salida 1 Entrada de permiso	

- Tipo SSH

Reglas de entrada (1/1)



Administrar etiquetas

Editar reglas de entrada

🔍 Filtrar reglas de grupo de seguridad

< 1 > ⚙️

<input checked="" type="checkbox"/>	Name ▼	ID de la regla del g... ▼	Versión ... ▼	Tipo ▼	Protoc... ▼	Interva... ▼	Origen
<input checked="" type="checkbox"/>	-	sgr-0d957d7f547f9eca4	IPv4	SSH	TCP	22	0.0.0.0/0

- Ahora ejecuto el comando `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`

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 run-instances --image-id ami-022e1a32d3f742bd8 --count 1 --instance-type t1.micro --key-name passwordy --security-groups SecurityEc2 --placement AvailabilityZone=us-east-1c
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-022e1a32d3f742bd8",
      "InstanceId": "i-0784db36944aa4eb1",
      "InstanceType": "t1.micro",
      "KeyName": "passwordy",
      "LaunchTime": "2023-06-19T22:20:28+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1c",
        "GroupName": "",
        "Tenancy": "default"
      }
    }
  ]
}
```

```
InstanceId": "i-0787ab50544da4eb1",  
  "InstanceType": "t1.micro",  
  "KeyName": "passwordqy",  
  "LaunchTime": "2023-06-19T22:20:28+00:00",  
  "Monitoring": {  
    "State": "disabled"  
  },  
  "Placement": {  
    "AvailabilityZone": "us-east-1c",  
    "GroupName": "",  
    "Tenancy": "default"  
  },  
  "PrivateDnsName": "ip-172-31-85-105.ec2.internal",  
  "PrivateIpAddress": "172.31.85.105",  
  "ProductCodes": [],  
  "PublicDnsName": "",  
  "State": {  
    "Code": 0,  
    "Name": "pending"  
  },  
  "StateTransitionReason": "",  
  "SubnetId": "subnet-04c60ffdbf2cf1a23",  
  "VpcId": "vpc-0359a0ad11ac69833",  
  "Architecture": "x86_64",  
  "BlockDeviceMappings": [],  
  "ClientToken": "07ca29e7-5a64-4407-86ee-04581ce71f59",  
  "EbsOptimized": false,  
  "EnaSupport": true,  
  "Hypervisor": "xen",  
  "NetworkInterfaces": [  
    {  
      "Attachment": {
```

```
"EnaSupport": true,
"Hypervisor": "xen",
"NetworkInterfaces": [
  {
    "Attachment": {
      "AttachTime": "2023-06-19T22:20:28+00:00",
      "AttachmentId": "eni-attach-0673770c8f87ecac8",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching",
      "NetworkCardIndex": 0
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "SecurityEc2",
        "GroupId": "sg-03300d9d9ee41d0c9"
      }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "12:2c:3c:1e:07:15",
    "NetworkInterfaceId": "eni-01cbb15c6d2772a2b",
    "OwnerId": "769139452531",
    "PrivateDnsName": "ip-172-31-85-105.ec2.internal",
    "PrivateIpAddress": "172.31.85.105",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-172-31-85-105.ec2.internal",
        "PrivateIpAddress": "172.31.85.105"
      },
      {
        "Primary": false,
        "PrivateDnsName": "ip-172-31-85-105.ec2.internal",
        "PrivateIpAddress": "172.31.85.105"
      }
    ]
  }
]
```

```
    ],
    "Ipv6Addresses": [],
    "MacAddress": "12:2c:3c:1e:07:15",
    "NetworkInterfaceId": "eni-01cbb15c6d2772a2b",
    "OwnerId": "769139452531",
    "PrivateDnsName": "ip-172-31-85-105.ec2.internal",
    "PrivateIpAddress": "172.31.85.105",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-172-31-85-105.ec2.internal",
        "PrivateIpAddress": "172.31.85.105"
      }
    ],
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-04c60ffdbf2cf1a23",
    "VpcId": "vpc-0359a0ad11ac69833",
    "InterfaceType": "interface"
  }
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
  {
    "GroupName": "SecurityEc2",
    "GroupId": "sg-03300d9d9ee41d0c9"
  }
],
"SourceDestCheck": true,
"StateReason": {
  "Code": "pending",
  "Message": "pending"
}
```

```
        "InterfaceType": "Interface",
    },
    ],
    "RootDeviceName": "/dev/xvda",
    "RootDeviceType": "ebs",
    "SecurityGroups": [
        {
            "GroupName": "SecurityEc2",
            "GroupId": "sg-03300d9d9ee41d0c9"
        }
    ],
    "SourceDestCheck": true,
    "StateReason": {
        "Code": "pending",
        "Message": "pending"
    },
    "VirtualizationType": "hvm",
    "CpuOptions": {
        "CoreCount": 1,
        "ThreadsPerCore": 1
    },
    "CapacityReservationSpecification": {
        "CapacityReservationPreference": "open"
    },
    "MetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 2,
        "HttpEndpoint": "enabled",
        "HttpProtocolIpv6": "disabled",
        "InstanceMetadataTags": "disabled"
    }
}
```

```

    },
    "CapacityReservationSpecification": {
      "CapacityReservationPreference": "open"
    },
    "MetadataOptions": {
      "State": "pending",
      "HttpTokens": "required",
      "HttpPutResponseHopLimit": 2,
      "HttpEndpoint": "enabled",
      "HttpProtocolIpv6": "disabled",
      "InstanceMetadataTags": "disabled"
    },
    "EnclaveOptions": {
      "Enabled": false
    },
    "BootMode": "uefi-preferred",
    "PrivateDnsNameOptions": {
      "HostnameType": "ip-name",
      "EnableResourceNameDnsARecord": false,
      "EnableResourceNameDnsAAAARecord": false
    },
    "MaintenanceOptions": {
      "AutoRecovery": "default"
    },
    "CurrentInstanceBootMode": "legacy-bios"
  },
  "OwnerId": "769139452531",
  "ReservationId": "r-07979b0c31eb9f5d7"
}

```

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

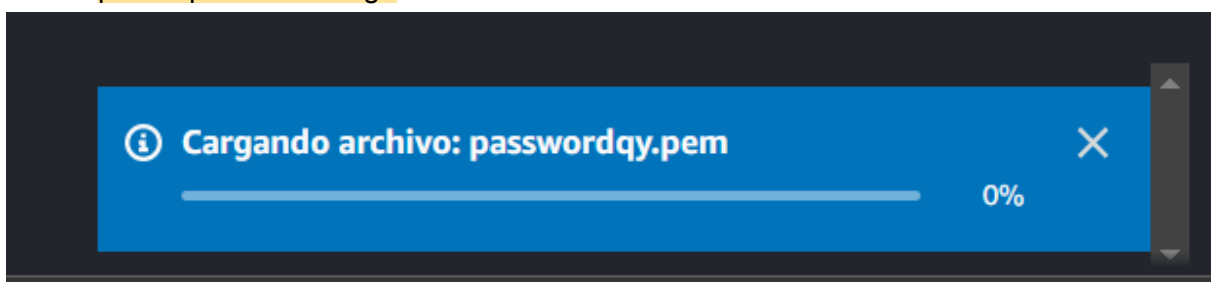
aws ec2 attach-volume --volume-id volume_id --instance-id id_instancia --device /dev/sdf ¿Cuál es la salida? Anteriormente se creó la instancia EC2, ahora adjunto el volumen de EBS a la instancia para proporcionarle almacenamiento

```

[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 attach-volume --volume-id "vol-0f2c251bc2c127054" --instance-id "i-0784db36944aa4eb1" --device /dev/sdf
{
  "AttachTime": "2023-06-19T22:27:10.760000+00:00",
  "Device": "/dev/sdf",
  "InstanceId": "i-0784db36944aa4eb1",
  "State": "attaching",
  "VolumeId": "vol-0f2c251bc2c127054"
}

```

4. Inicia sesión en la instancia EC2 a través de **ssh**. En tu instancia EC2, cambie a **root**.
 - En una nueva ventana en aws cloud shell cargare el archivo de la clave de pares que se descargó



- En la instancia ingreso a conectar y me dirijo hacia Cliente SSh para conectarse mediante el DNS público

[EC2](#) > [Instancias](#) > [i-0784db36944aa4eb1](#) > [Conectarse a la instancia](#)

Conectarse a la instancia [Información](#)

Conéctese a la instancia i-0784db36944aa4eb1 mediante cualquiera de estas opciones


Conexión de la instancia EC2



Administrador de sesiones

Cliente SSH


Consola de serie de EC2

ID de la instancia

 [i-0784db36944aa4eb1](#)

1. Abra un cliente SSH.
2. Localice el archivo de clave privada. La clave utilizada para lanzar esta instancia es passwordqy.pem
3. Ejecute este comando, si es necesario, para garantizar que la clave no se pueda ver públicamente.
 `chmod 400 passwordqy.pem`
4. Conéctese a la instancia mediante su DNS público:
 `ec2-3-82-176-137.compute-1.amazonaws.com`

Ejemplo:

 `ssh -i "passwordqy.pem" ec2-user@ec2-3-82-176-137.compute-1.amazonaws.com`

- Uso el comando `chmod 400 passwordqy(nombre de mi par de claves).pem`

```
[cloudshell-user@ip-10-4-87-28 ~]$ pwd
/home/cloudshell-user
[cloudshell-user@ip-10-4-87-28 ~]$ ls
passwordqy.pem
[cloudshell-user@ip-10-4-87-28 ~]$ curl http://www.icanhazip.com
3.85.231.220
[cloudshell-user@ip-10-4-87-28 ~]$ ssh -i "passwordqy.pem" ec2-user@ec2-3-82-176-137.compute-1.amazonaws.com
The authenticity of host 'ec2-3-82-176-137.compute-1.amazonaws.com (3.82.176.137)' can't be established.
ECDSA key fingerprint is SHA256:05XRQWR4pkvK5G6l8v45OWuhQMCmbx/pncBuqS8D8Y.
ECDSA key fingerprint is MD5:bf:9a:4c:57:20:fb:e4:7a:bd:d6:7f:95:8a:cc:24:09.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-82-176-137.compute-1.amazonaws.com,3.82.176.137' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: UNPROTECTED PRIVATE KEY FILE! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'passwordqy.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "passwordqy.pem": bad permissions
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[cloudshell-user@ip-10-4-87-28 ~]$ chmod 400 passwordqy.pem
[cloudshell-user@ip-10-4-87-28 ~]$ ssh -i "passwordqy.pem" ec2-user@ec2-3-82-176-137.compute-1.amazonaws.com
#
      _
     _\ #####          Amazon Linux 2023
    _/ \#####\
   _/  \###|
  _/   \#/
 _/    V~'  ' ->
      /
     /
    /
   /
  /
 /
/m/

[ec2-user@ip-172-31-85-105 ~]$ sudo -i
[root@ip-172-31-85-105 ~]#
```

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?

```
[root@ip-172-31-85-105 ~]# 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
xvdf         202:80    0   1G  0 disk
[root@ip-172-31-85-105 ~]# mkfs -F /dev/xvdf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 0338c389-bd32-4e7a-81da-becc963aee6f
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

mkdir /data

mount /dev/xvdf /data

cd /data/

df ¿Cuál es la salida?

```
[root@ip-172-31-85-105 ~]# mkdir /data
[root@ip-172-31-85-105 ~]# mount /dev/xvdf /data
[root@ip-172-31-85-105 ~]# cd /data/
[root@ip-172-31-85-105 data]# df
Filesystem      1K-blocks    Used Available Use% Mounted on
devtmpfs         4096         0      4096   0% /dev
tmpfs           279984         0    279984   0% /dev/shm
tmpfs           111996     1812    110184   2% /run
/dev/xvda1      8310764 1545684   6765080  19% /
tmpfs           279988         0    279988   0% /tmp
tmpfs           55996         0     55996   0% /run/user/1000
/dev/xvdf       1030828      24     978376   1% /data
[root@ip-172-31-85-105 data]#
```

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

- Creó el archivo `aws_user.txt` con `echo` y redirijo el contenido hacia el archivo y fuerzo la sincronización

```
[root@ip-172-31-85-105 data]# echo "Este es mi primer archivo txt" > aws_user.txt
[root@ip-172-31-85-105 data]# sync
```

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? **Se crea la instantánea para respaldar el contenido**

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 create-snapshot --volume-id "vol-0f2c251bc2c127054" --description 'Esta es mi instantanea de volumen'
{
  "Description": "Esta es mi instantanea de volumen",
  "Encrypted": false,
  "OwnerId": "769139452531",
  "Progress": "",
  "SnapshotId": "snap-06f6537cd34a66295",
  "StartTime": "2023-06-19T22:58:30.253000+00:00",
  "State": "pending",
  "VolumeId": "vol-0f2c251bc2c127054",
  "VolumeSize": 1,
  "Tags": []
}
```

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

`aws ec2 describe-snapshots --snapshot-id snapshot_id` Para verificar el estado de la instantánea

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.

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 describe-snapshots --snapshot-id "snap-06f6537cd34a66295"
{
  "Snapshots": [
    {
      "Description": "Esta es mi instantanea de volumen",
      "Encrypted": false,
      "OwnerId": "769139452531",
      "Progress": "100%",
      "SnapshotId": "snap-06f6537cd34a66295",
      "StartTime": "2023-06-19T22:58:30.253000+00:00",
      "State": "completed",
      "VolumeId": "vol-0f2c251bc2c127054",
      "VolumeSize": 1,
      "StorageTier": "standard"
    }
  ]
}
```

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 --región us-east-1 --availability-zone us-east-1c --snapshot-id snapshot_id`

¿Cuál es la salida?

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 create-volume --region us-east-1 --availability-zone us-east-1c --snapshot-id "snap-06f6537cd34a66295"
{
  "AvailabilityZone": "us-east-1c",
  "CreateTime": "2023-06-19T23:05:41+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "snap-06f6537cd34a66295",
  "State": "creating",
  "VolumeId": "vol-05f6017aec81bf8a5",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
```

Comprueba el estado del volumen. ¿Qué comando ejecutaste para verificar el estado? ¿Cuál es la salida? **aws ec2 describe-volumes --volume-ids "vol-05f6017aec81bf8a5"**, el estado es disponible

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 describe-volumes --volume-ids "vol-05f6017aec81bf8a5"
{
  "Volumes": [
    {
      "Attachments": [],
      "AvailabilityZone": "us-east-1c",
      "CreateTime": "2023-06-19T23:05:41.437000+00:00",
      "Encrypted": false,
      "Size": 1,
      "SnapshotId": "snap-06f6537cd34a66295",
      "State": "available",
      "VolumeId": "vol-05f6017aec81bf8a5",
      "Iops": 100,
      "VolumeType": "gp2",
      "MultiAttachEnabled": false
    }
  ]
}
```

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? Se adjunta el volumen a la instancia creada

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 attach-volume --volume-id "vol-05f6017aec81bf8a5" --instance-id "i-0784db36944aa4eb1" --device /dev/sdg
{
  "AttachTime": "2023-06-19T23:18:23.425000+00:00",
  "Device": "/dev/sdg",
  "InstanceId": "i-0784db36944aa4eb1",
  "State": "attaching",
  "VolumeId": "vol-05f6017aec81bf8a5"
}
```

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? **sudo mkdir /data2**
sudo mount /dev/xvdcg cd/data2

```
[root@ip-172-31-85-105 data2]# sudo mkdir /data2
[root@ip-172-31-85-105 data2]# sudo mount /dev/xvdcg /data2
[root@ip-172-31-85-105 data2]# cd /data2
```

Cambia el directorio de su instancia EC2 a **/data2**. ¿Viste el archivo **aws_user.txt**? Si, se observa en la imagen

```
[root@ip-172-31-85-105 data2]# ls
aws_user.txt  lost+found
```

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:/data2# cd /

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

root@ip-10-45-185-154:/# unmount /dev/xvdg **Umount desmonta el volumen**

```
[root@ip-172-31-85-105 data2]# cd /  
[root@ip-172-31-85-105 /]# umount /dev/xvdf  
[root@ip-172-31-85-105 /]# umount /dev/xvdg  
[root@ip-172-31-85-105 /]#
```

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 desconecte el volumen de la instancia ec2**

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 detach-volume --volume-id "vol-0f2c251bc2c127054"  
{  
  "AttachTime": "2023-06-19T22:27:10+00:00",  
  "Device": "/dev/sdf",  
  "InstanceId": "i-0784db36944aa4eb1",  
  "State": "detaching",  
  "VolumeId": "vol-0f2c251bc2c127054"  
}
```

Repita estos dos comandos para el segundo volumen, cuyo volume_id deberías haber obtenido del paso 6. ¿Qué comandos usastes? ¿Cuáles son las salidas?

Desconecto el segundo volumen de la instancia ec2

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 detach-volume --volume-id "vol-05f6017aec81bf8a5"  
{  
  "AttachTime": "2023-06-19T23:18:23+00:00",  
  "Device": "/dev/sdg",  
  "InstanceId": "i-0784db36944aa4eb1",  
  "State": "detaching",  
  "VolumeId": "vol-05f6017aec81bf8a5"  
}
```

Ahora elimino los volúmenes

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 delete-volume --volume-id "vol-0f2c251bc2c127054"  
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 delete-volume --volume-id "vol-05f6017aec81bf8a5"  
[cloudshell-user@ip-10-4-87-28 ~]$
```

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

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 delete-volume --volume-id "vol-05f6017aec81bf8a5"  
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 delete-snapshot --snapshot-id "snap-06f6537cd34a66295"  
[cloudshell-user@ip-10-4-87-28 ~]$
```

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?

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 create-volume --size 1 --region us-east-1 --availability-zone us-east-1c  
{  
  "AvailabilityZone": "us-east-1c",  
  "CreateTime": "2023-06-19T23:40:52+00:00",  
  "Encrypted": false,  
  "Size": 1,  
  "SnapshotId": "",  
  "State": "creating",  
  "VolumeId": "vol-0f7867513137d1523",  
  "Iops": 100,  
  "Tags": [],  
  "VolumeType": "gp2",  
  "MultiAttachEnabled": false  
}
```

```
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 create-volume --size 1 --region us-east-1 --availability-zone us-east-1c
{
  "AvailabilityZone": "us-east-1c",
  "CreateTime": "2023-06-19T23:41:24+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-0779a2ea49f6ab4ef",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}

[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 attach-volume --volume-id "vol-0f7867513137d1523" --instance-id "i-0784db36944aa4eb1" --device /dev/xvdf
{
  "AttachTime": "2023-06-19T23:45:48.607000+00:00",
  "Device": "/dev/xvdf",
  "InstanceId": "i-0784db36944aa4eb1",
  "State": "attaching",
  "VolumeId": "vol-0f7867513137d1523"
}

[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 attach-volume --volume-id "vol-0779a2ea49f6ab4ef" --instance-id "i-0784db36944aa4eb1" --device /dev/xvdf
{
  "AttachTime": "2023-06-19T23:47:40.242000+00:00",
  "Device": "/dev/xvdf",
  "InstanceId": "i-0784db36944aa4eb1",
  "State": "attaching",
  "VolumeId": "vol-0779a2ea49f6ab4ef"
}
```

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**

```
[root@ip-172-31-85-105 /]# yum update -y
Last metadata expiration check: 1:30:18 ago on Mon Jun 19 22:21:12 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-85-105 /]# yum install mdadm -y
Last metadata expiration check: 1:30:49 ago on Mon Jun 19 22:21:12 2023.
Dependencies resolved.
=====
Package                                Architecture                               Version
=====
Installing:
mdadm                                   x86_64                                     4.2-3.amzn2023.0.5

Transaction Summary
=====
Install 1 Package

Total download size: 425 k
Installed size: 990 k
Downloading Packages:
mdadm-4.2-3.amzn2023.0.5.x86_64.rpm
=====
```

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/shd2"**, 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?

```
Complete!
[root@ip-172-31-85-105 /]# mdadm --create /dev/md0 --level 0 --metadata=1.1 --raid-devices 2 /dev/xvdf /dev/xvdf
mdadm: array /dev/md0 started.
[root@ip-172-31-85-105 /]#
```

13. Ahora, podemos comprobar el estado de la matriz RAID 0. Emite lo siguiente. **mdadm --detail /dev/md0** ¿Cuál es la salida?

```
[root@ip-172-31-85-105 /]# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.1
    Creation Time : Mon Jun 19 23:55:28 2023
    Raid Level : raid0
    Array Size : 2093056 (2044.00 MiB 2143.29 MB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Mon Jun 19 23:55:28 2023
    State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Layout : -unknown-
    Chunk Size : 512K

    Consistency Policy : none

    Name : ip-172-31-85-105.ec2.internal:0 (local to host ip-172-31-85-105.ec2.internal)
    UUID : ae524791:c862b542:1e262577:42c32325
    Events : 0

    Number Major Minor RaidDevice State
       0     202      80         0   active sync  /dev/sdf
       1     202      96         1   active sync  /dev/sdg
```

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?

```
[root@ip-172-31-85-105 /]# mkfs /dev/md0
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 523264 4k blocks and 130816 inodes
Filesystem UUID: b0b5f8e9-647e-481a-909b-fa2bb3db810c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-85-105 /]# mkdir /data3
[root@ip-172-31-85-105 /]# mount /dev/md0 /data3
[root@ip-172-31-85-105 /]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
devtmpfs                4096         0      4096   0% /dev
tmpfs                  279984         0    279984   0% /dev/shm
tmpfs                   111996    1824    110172   2% /run
/dev/xvda1             8310764 1549124   6761640  19% /
tmpfs                   279988         0    279988   0% /tmp
tmpfs                   55996         0     55996   0% /run/user/1000
/dev/md0                2057128     24   1952452   1% /data3
```

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 / unmount /dev/md0 mdadm --stop /dev/md0**

```

/dev/md0 2037128 24 1932432 18 70
[root@ip-172-31-85-105 /]# cd /
[root@ip-172-31-85-105 /]# umount /dev/md0
[root@ip-172-31-85-105 /]# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
[root@ip-172-31-85-105 /]#

```

- Se detiene el arreglo RAID

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? Recuerda, después de 10 segundos

```

}
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 detach-volume --volume-id "vol-0f7867513137d1523"
{
  "AttachTime": "2023-06-19T23:45:48+00:00",
  "Device": "/dev/xvdf",
  "InstanceId": "i-0784db36944aa4eb1",
  "State": "detaching",
  "VolumeId": "vol-0f7867513137d1523"
}

[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 detach-volume --volume-id "vol-0779a2ea49f6ab4ef"
{
  "AttachTime": "2023-06-19T23:47:40+00:00",
  "Device": "/dev/xvdg",
  "InstanceId": "i-0784db36944aa4eb1",
  "State": "detaching",
  "VolumeId": "vol-0779a2ea49f6ab4ef"
}

[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 delete-volume --volume-id "vol-0f7867513137d1523"
[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 delete-volume --volume-id "vol-0779a2ea49f6ab4ef"
[cloudshell-user@ip-10-4-87-28 ~]$

```

- Finalizó la instancia

```

[cloudshell-user@ip-10-4-87-28 ~]$ aws ec2 terminate-instances --instance-ids "i-0784db36944aa4eb1"
{
  "TerminatingInstances": [
    {
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "InstanceId": "i-0784db36944aa4eb1",
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
[cloudshell-user@ip-10-4-87-28 ~]$

```

- Se comprueba que la instancia está detenida

ar

[Alt+S]

Norte de Virginia

voclabs/user2539893=TORIBIO_ALVARADO_STEP

Instancias (2) Información

Conectar

Estado de la instancia

Acciones

Lanzar

🔍

Buscar instancia por atributo o etiqueta (case-sensitive)

<input type="checkbox"/>	Name	ID de la instancia	Estado de la ins...	Tipo de inst...	Comprobación ...	Estado de la
<input type="checkbox"/>	-	i-0784db36944aa4eb1	Terminada	t1.micro	-	Sin alarmas
<input type="checkbox"/>	Bastion ...	i-00ae0f0e02343dc6c	En ejecución	t2.micro	2/2 comprobador	Sin alarmas