

Lab Manual -- Attach Data Volume to existing EC2 instance.

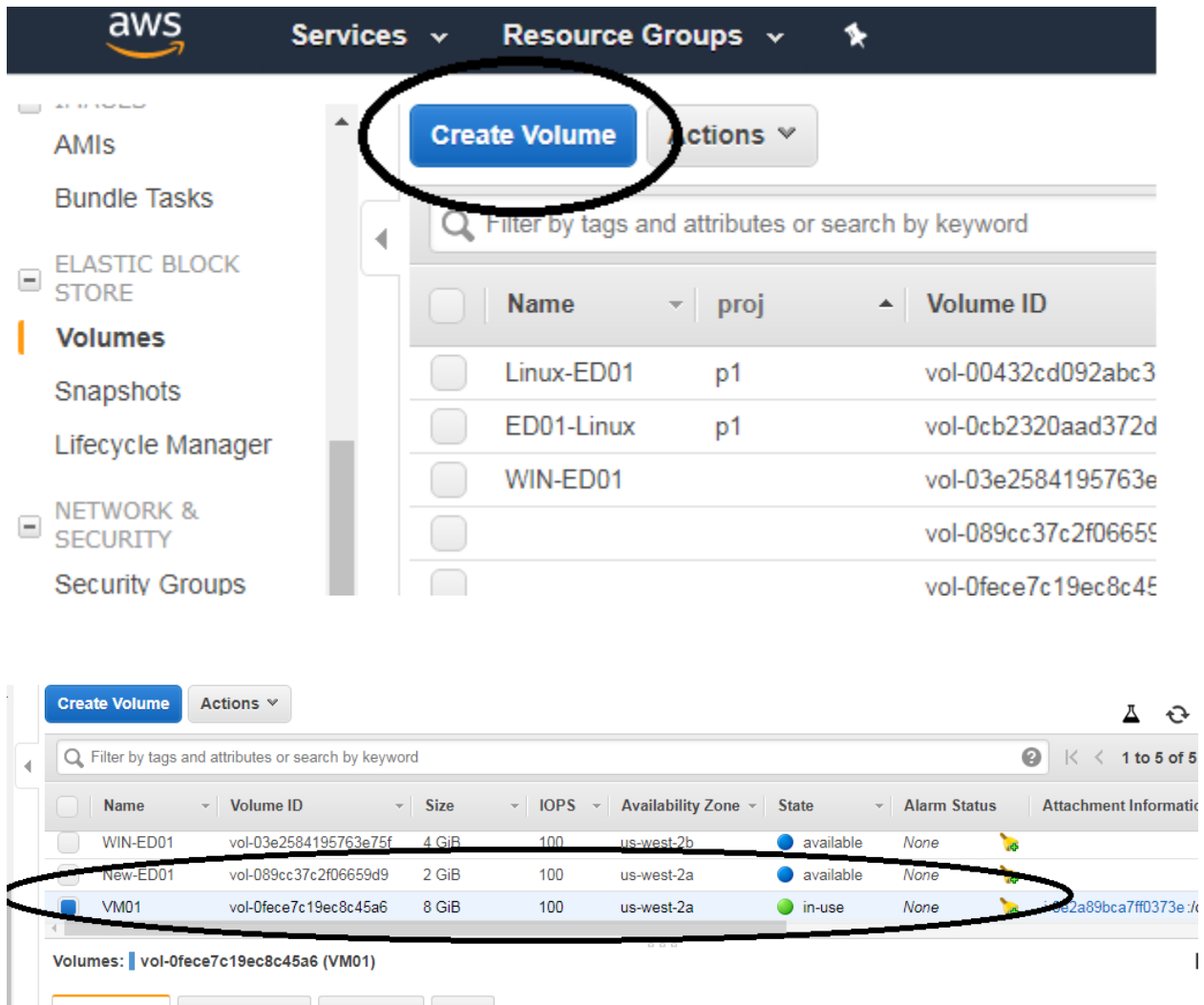
### Steps

1. Attach the volume to the EC2 instance from the AWS console
2. Format and mount the volume inside the Linux Machine.
3. Detach the Volume from the Linux Machine.

### 1. Attach the volume to the EC2 instance from the AWS console.

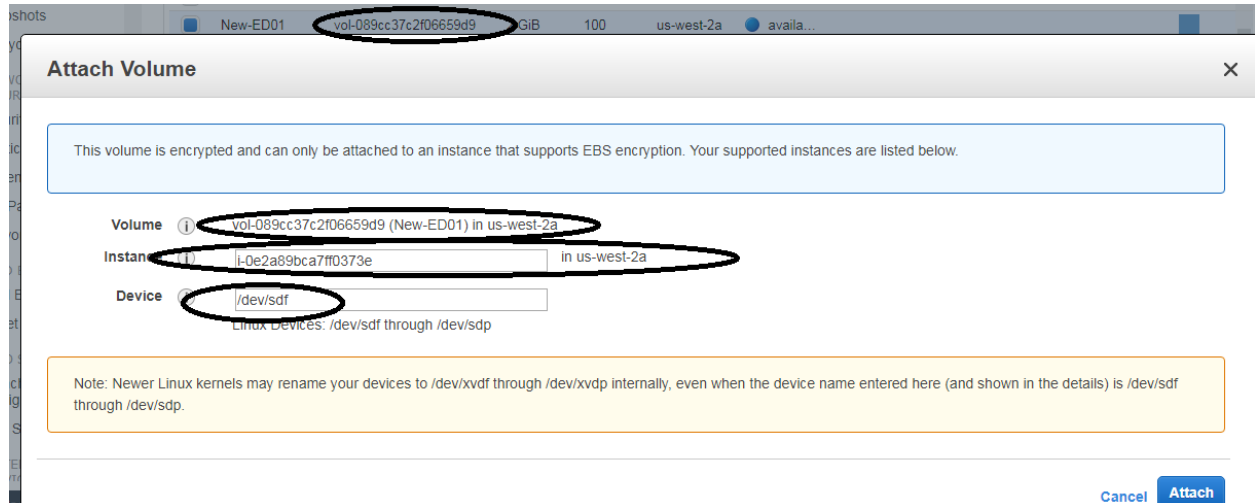
Note: -- Assuming the Volume and the EC2 instance are in the same “Availability Zone”

Click on “Volumes” on the left panel under EC2 page.



Make sure the EC2 instance and new volume (2 GB) are in the same AZ.

## AWS – EBS – Volumes



The Device says → **“/dev/sdf”** – which means that this virtual disk would be listed as this in the linux

Filter by tags and attributes or search by keyword							
	Name	Volume ID	Size	IOPS	Availability	State	Attachment Information
<input checked="" type="checkbox"/>	New-ED01	vol-089cc37c2f06659d9	2 GiB	100	us-west-2a	in-use	i-0e2a89bca7ff0373e:/dev/sdf (attached)
<input type="checkbox"/>	VM01	vol-0fece7c19ec8c45ab	8 GiB	100	us-west-2a	in-use	i-0e2a89bca7ff0373e:/dev/sda (attached)

The volume is attached to the EC2 instance now.

### 2. Format and mount the volume inside the Linux Machine.

#### Step 1:

Login to the EC2 instance

# `ls -l /dev | grep sd` – This command will list the files and folders in the directory “dev” with a filter for the name as “sd”

```
[root@ip-192-168-1-110 ec2-user]# ls -l /dev | grep sd
lrwxrwxrwx 1 root root          4 Jul 24 04:48 sda -> xvda
lrwxrwxrwx 1 root root          5 Jul 24 04:48 sda1 -> xvda1
lrwxrwxrwx 1 root root          4 Jul 24 05:00 sdf -> xvdf
[root@ip-192-168-1-110 ec2-user]#
```

The above screen shows that the disk is attached to the OS.

Ex: -- “sdf” in the above image.

#### Step2:

\$ `sudo su`

\$ `fdisk /dev/sdf`

```
[root@ip-172-31-29-186 ec2-user]# fdisk /dev/sdf

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xd547d5fd.

Command (m for help):
```

Press “n” for “new partition”

```

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): █

```

Select “p”

And keep pressing “Enter” ( 3times)

```

Select (default p): p
Partition number (1-4, default 1):
First sector (2048-4194303, default 2048):
Using default value 2048
Last sector, +sectors or +size[K,M,G] (2048-4194303, default 4194303):
Using default value 4194303
Partition 1 of type Linux and of size 2 GiB is set
Command (m for help): █

```

Now save the configuration “w”

```

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@localhost ~]# █

```

Now lets check if the partition is created.

```

[root@ip-172-31-29-186 ec2-user]# ls /dev
autofs          hpet            network_latency  sdf             tty13          tt
block           hugepages       network_throughput sdf1            tty14          tt
btrfs-control  initctl         null             snapshot        tty15          tt
char           input           port             stderr          tty16          tt
console        kmsg           ppp             stdin           tty17          tt
core           log            psaux           stdout          tty18          tt
cpu            loop-control    ptmx            tty             tty19          tt
cpu_dma_latency mapper          pts             tty0            tty2           tt
cuse           mcelog         random          tty1            tty20          tt
disk           mem            rtc             ttyl            tty21          tt

```

The one marked in the circle, shows that “sdf1” is created, which proves that the partition for the virtual disk “sdf” is created.

### Step3: Format the disk

Command → “mkfs.ext4 /dev/sdf1”

```
[root@ip-172-31-29-186 ec2-user]# mkfs.ext4 /dev/sdf1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
327680 inodes, 1310464 blocks
65523 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-29-186 ec2-user]#
```

### Step4: Mount the disk “mount”

Create a new folder and mount the new partition

Command → **mkdir ed01**

→ “**mount /dev/sdf1 ed01**”

```
[root@ip-172-31-29-186 ec2-user]# mkdir ed01
[root@ip-172-31-29-186 ec2-user]# mount /dev/sdf1 ed01
[root@ip-172-31-29-186 ec2-user]# ls -l
total 4
drwxr-xr-x 3 root root 4096 Jun  2 08:32 ed01
[root@ip-172-31-29-186 ec2-user]#
```

```
[root@ip-172-31-29-186 ec2-user]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        475M   0    475M   0% /dev
tmpfs           492M   0    492M   0% /dev/shm
tmpfs           492M 452K    492M   1% /run
tmpfs           492M   0    492M   0% /sys/fs/cgroup
/dev/xvda1       8.0G  1.2G   6.9G  15% /
tmpfs           99M   0     99M   0% /run/user/0
tmpfs           99M   0     99M   0% /run/user/1000
/dev/xvdf1       4.8G  20M   4.6G   1% /home/ec2-user/ed01
[root@ip-172-31-29-186 ec2-user]#
```

The above screen shows that the new volume is attached to a folder inside the LINUX OS.

### Step5: Create a file in the new volume.

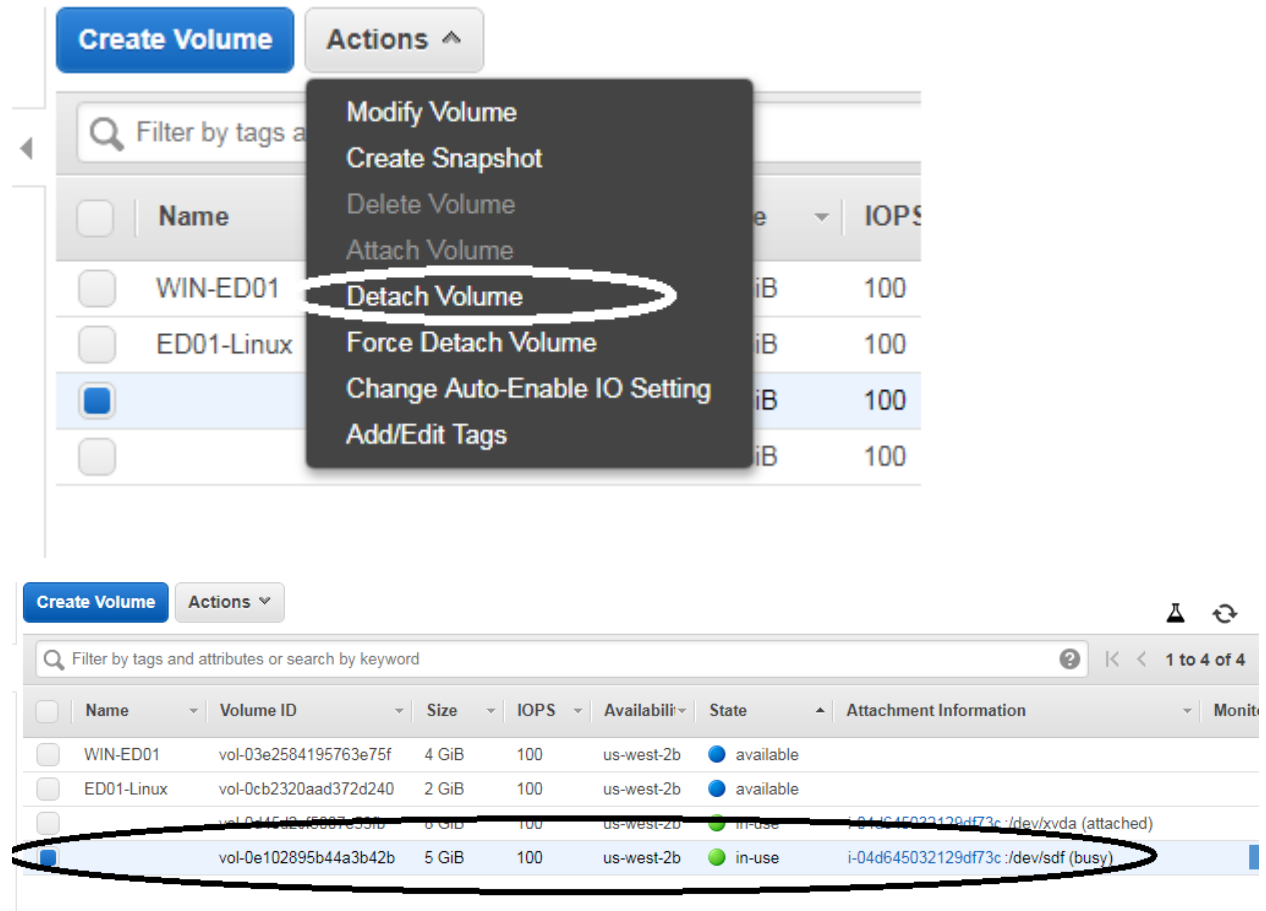
```
[root@localhost ~]# cd ed01
[root@localhost ed01]# ls
lost+found
[root@localhost ed01]# vi file
[root@localhost ed01]# ls -l
total 20
-rw-r--r--. 1 root root    40 Nov 18 04:39 file
drwx-----. 2 root root 16384 Nov 18 04:37 lost+found
[root@localhost ed01]# cd ..
[root@localhost ~]# ls -l
total 58600
-rw-----. 1 root root    1448 May 12 2018 anaconda-ks.cfg
drwxr-xr-x. 3 root root    4096 Nov 18 04:37 ed01
```



### 3. Detach the volume from the Linux Machine.

Note: -- The volume is attached inside the linux machine to a folder .

Lets detach the volume from the AWS Console

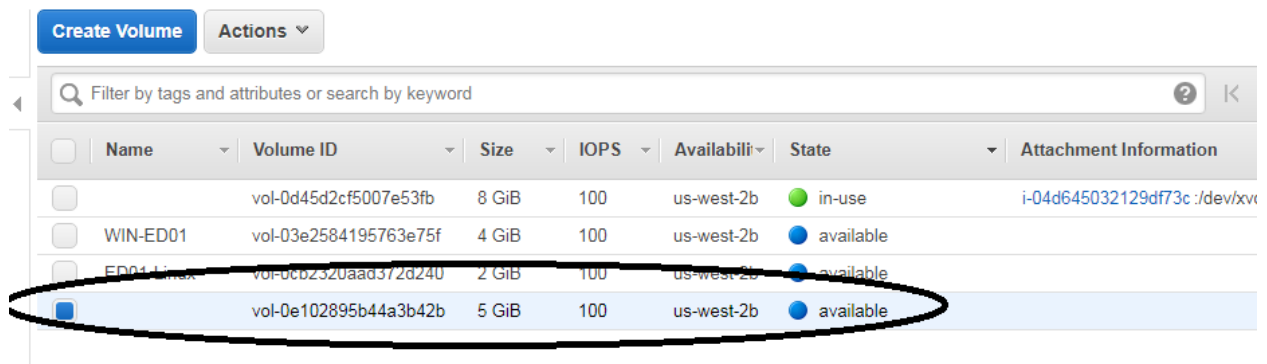


The Volume is **NOT** getting detached, Because the **LINUX OS** will not allow it to do so, until we “**UN MOUNT**” the partition inside the LINUX OS.

Command

➔ `$ umount ed01`

```
[root@ip-172-31-29-186 ec2-user]# umount ed01
[root@ip-172-31-29-186 ec2-user]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        475M   0    475M   0% /dev
tmpfs           492M   0    492M   0% /dev/shm
tmpfs           492M 444K   492M   1% /run
tmpfs           492M   0    492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.2G   6.9G  15% /
tmpfs           99M   0     99M   0% /run/user/0
tmpfs           99M   0     99M   0% /run/user/1000
[root@ip-172-31-29-186 ec2-user]#
```



The screenshot shows the AWS Management Console interface for EBS volumes. At the top, there are buttons for 'Create Volume' and 'Actions'. Below is a search bar with the text 'Filter by tags and attributes or search by keyword'. The main table lists several EBS volumes. The volume 'vol-0e102895b44a3b42b' with a size of 5 GiB and state 'available' is circled in blue. The table has columns for Name, Volume ID, Size, IOPS, Availability, State, and Attachment Information.

Name	Volume ID	Size	IOPS	Availability	State	Attachment Information
	vol-0d45d2cf5007e53fb	8 GiB	100	us-west-2b	in-use	i-04d645032129df73c :/dev/xv
WIN-ED01	vol-03e2584195763e75f	4 GiB	100	us-west-2b	available	
ED01 Linux	vol-0c6b2320aad372d240	2 GiB	100	us-west-2b	available	
	vol-0e102895b44a3b42b	5 GiB	100	us-west-2b	available	

Now the 5GB Volume is got Detached on the AWS Console.