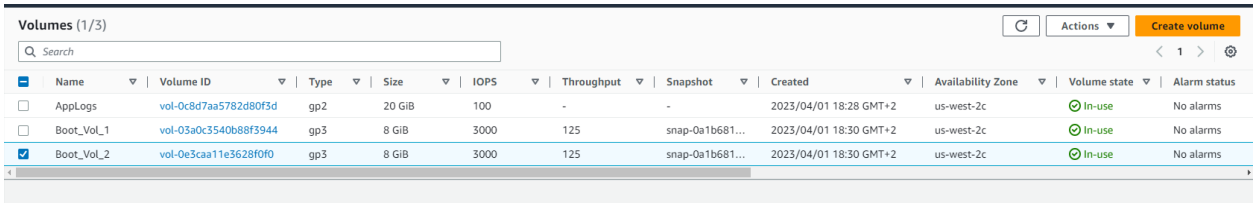
Introduction to Amazon Elastic Block Store (Amazon EBS)

Task 1.1: Name existing EBS Volumes



Volumes (1/3)

Actions | Create volume

Search

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	Availability Zone	Volume state	Alarm status
AppLogs	vol-0c8d7aa5782d80f3d	gp2	20 GiB	100	-	-	2023/04/01 18:28 GMT+2	us-west-2c	In-use	No alarms
Boot_Vol_1	vol-03a0c3540b88f3944	gp3	8 GiB	3000	125	snap-0a1b681...	2023/04/01 18:30 GMT+2	us-west-2c	In-use	No alarms
Boot_Vol_2	vol-0e3caa11e3628f0f0	gp3	8 GiB	3000	125	snap-0a1b681...	2023/04/01 18:30 GMT+2	us-west-2c	In-use	No alarms

Task 1.2: Create an EBS Volume

New_App_Main	vol-06b30b5cb10a6b816	gp3	30 GiB	3000	250	-	2023/04/01 18:40 GMT+2	us-west-2c	Available	No alarms

Your new EBS volume appears in the volume list. Wait for the **State** of the new volume to change to **available**. Your new EBS volume is ready to be attached to an EC2 instance. Attaching your EBS volume to an EC2 instance allows the instance to use the volume as block storage.

Task 1.3: Attach EBS Volume to an EC2 instance

After attaching the volume to the instance, the Volume state changes to In-use

In-use

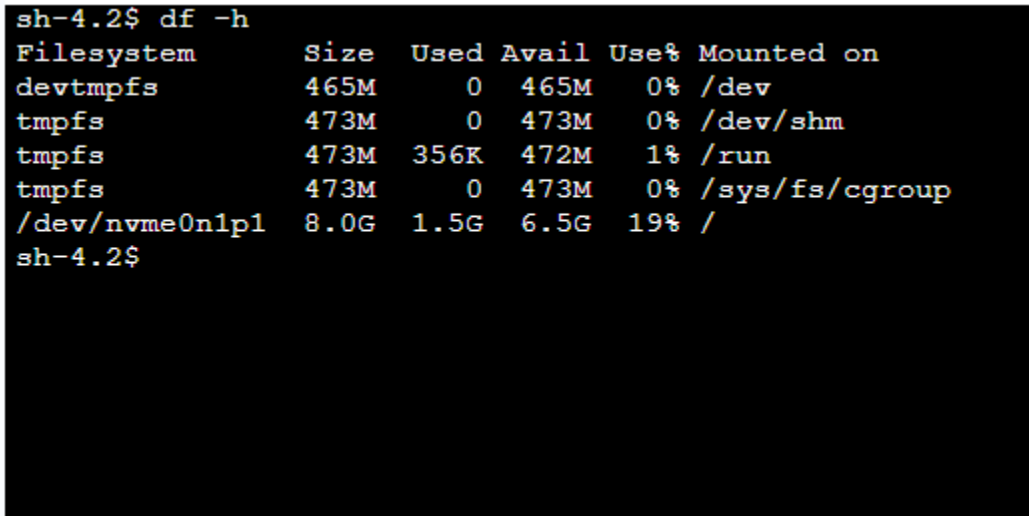
Task 2: Create and configure a file system on an attached EBS volume

- First we connect to the New_App instance

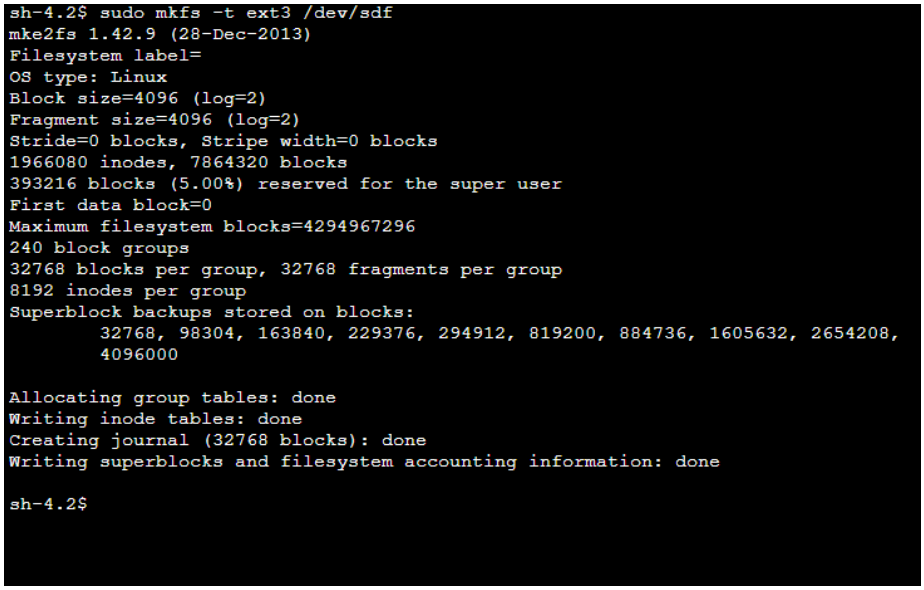
A new browser window opens with a running terminal session. **Session Manager** creates a secure connection to the **New_App** EC2 instance. This secure connection allows you to run Linux commands on the EC2 instance.

Run the following command to determine the available storage on the instance:
`df -h`

- **df** is for disk free (free disk space), and **h** for the available volume sizes in a human-readable format



Create a Linux file system with the following command:
`sudo mkfs -t ext3 /dev/sdf`



A new mounted volume created:

```
sh-4.2$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        465M     0  465M   0% /dev
tmpfs           473M     0  473M   0% /dev/shm
tmpfs           473M  356K  472M   1% /run
tmpfs           473M     0  473M   0% /sys/fs/cgroup
/dev/nvme0n1p1  8.0G  1.5G  6.5G  19% /
/dev/nvme1n1     30G  156K   28G   1% /mnt/data-store
sh-4.2$
```

We create a new text file in the mounted volume and display the contents of it with the **cat** command:

```
sh-4.2$ sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
sh-4.2$ cat /mnt/data-store/file.txt
some text has been written
sh-4.2$
```

Task 3: Modify the EBS volume size and expand the file system on the volume

- Volume size can only be increased using EBS Dynamic Volumes

50 GiB size:

☐	New_App_Main	vol-06b30b5cb10a6b816	gp3	50 GiB

Task 3.2: Expand the volume of your file system

lslbk - for displaying list block devices

```
sh-4.2$ 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
nvme1n1       259:3    0  50G  0 disk /mnt/data-store
sh-4.2$
```

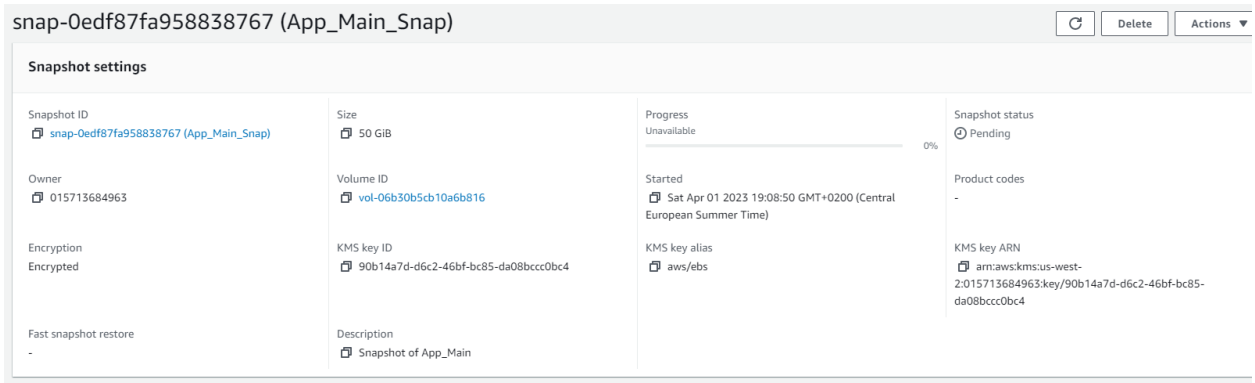
- Notice the size of the new file system of the volume is 50G

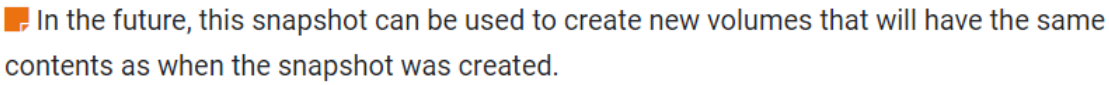
Task 4: Modify the EBS volume type and provisioned performance for an existing application

- Modify the IOPS and Throughput of the AppLogs volume:

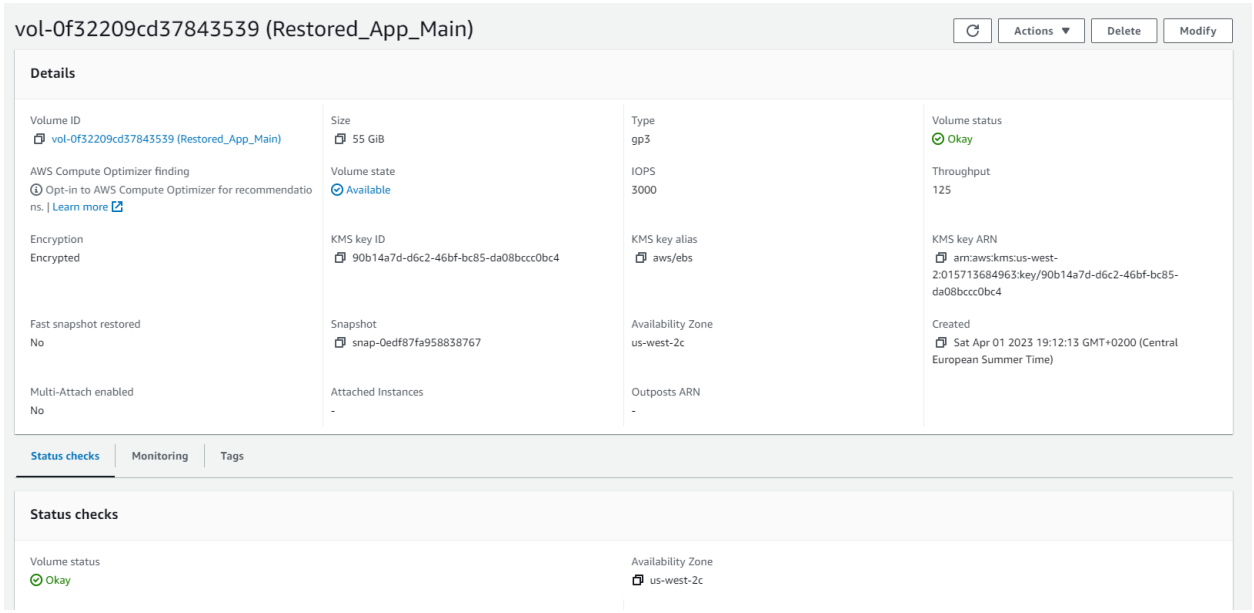
AppLogs	vol-0c8d7aa5782d80f3d	gp3	20 GiB	6000	250	-	2023/04/01 18:28 GMT+2	us-west-2c	In-use - optimizi	No alarms

Task 5: Configure a snapshot for an existing EBS volume





Task 6: Restore an EBS volume from an existing snapshot



- We repeat the same steps as in task 3 for mounting the restored storage volume

```
sh-4.2$ sudo mkdir /mnt/data-store2
sh-4.2$ sudo mount /dev/sdg /mnt/data-store2
sh-4.2$ 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
nvme1n1       259:3    0  50G  0 disk /mnt/data-store
nvme2n1       259:4    0  55G  0 disk /mnt/data-store2
sh-4.2$ df -h
```

- Notice the difference in size