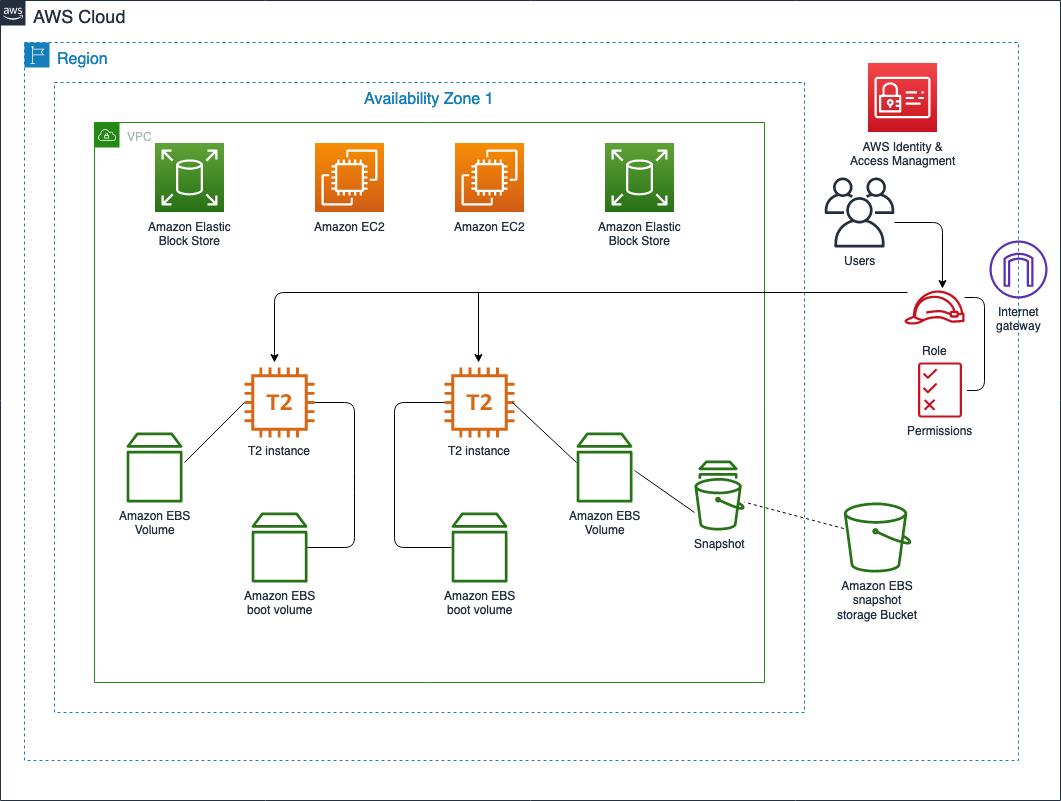
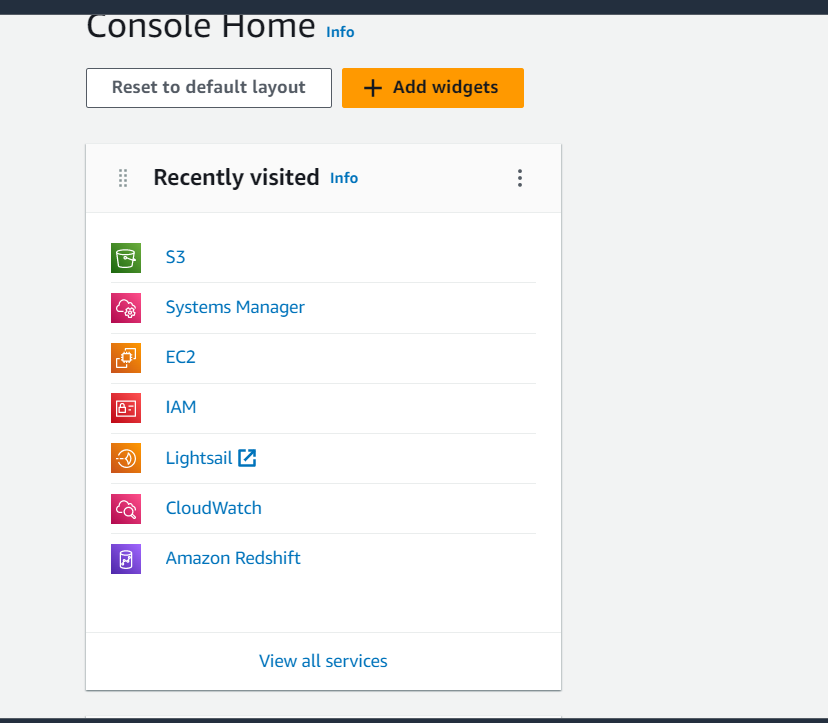
Introduction to Amazon Elastic Block Store

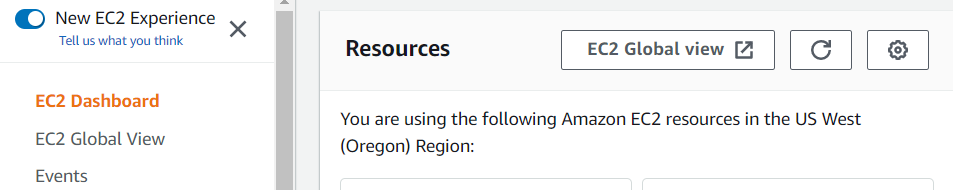


Access the AWS console 

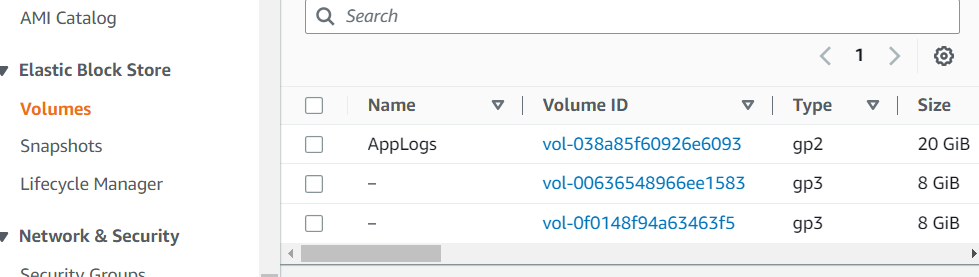
Task 1.1 Name existing EBS volumes

Open all services and open EC2

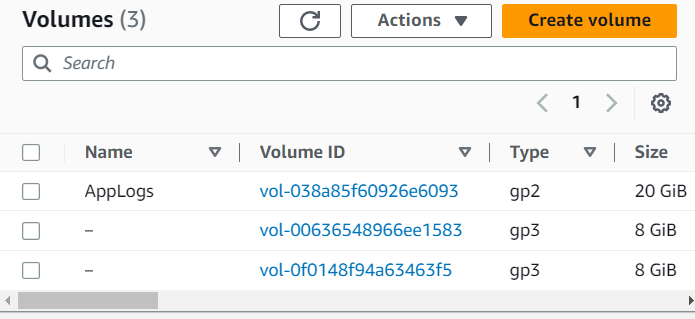
And enable New EC2 experience



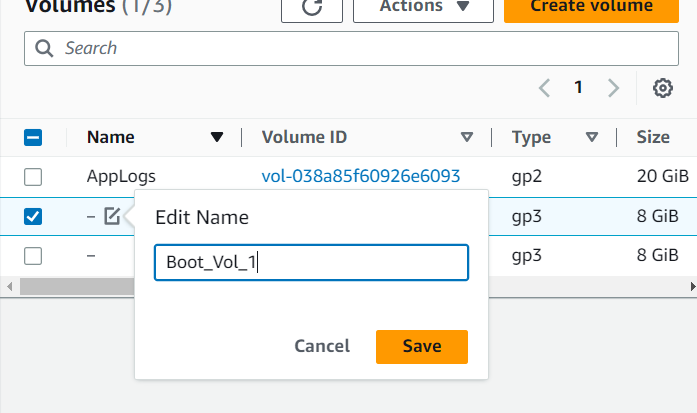
Choose elastic block store > volumes



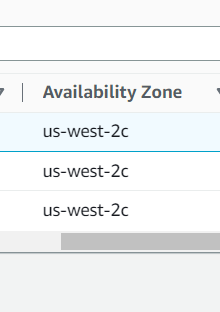
Two of the volumes are not named and we need to name them and tag the EBS volumes to keep track of their content and purpose



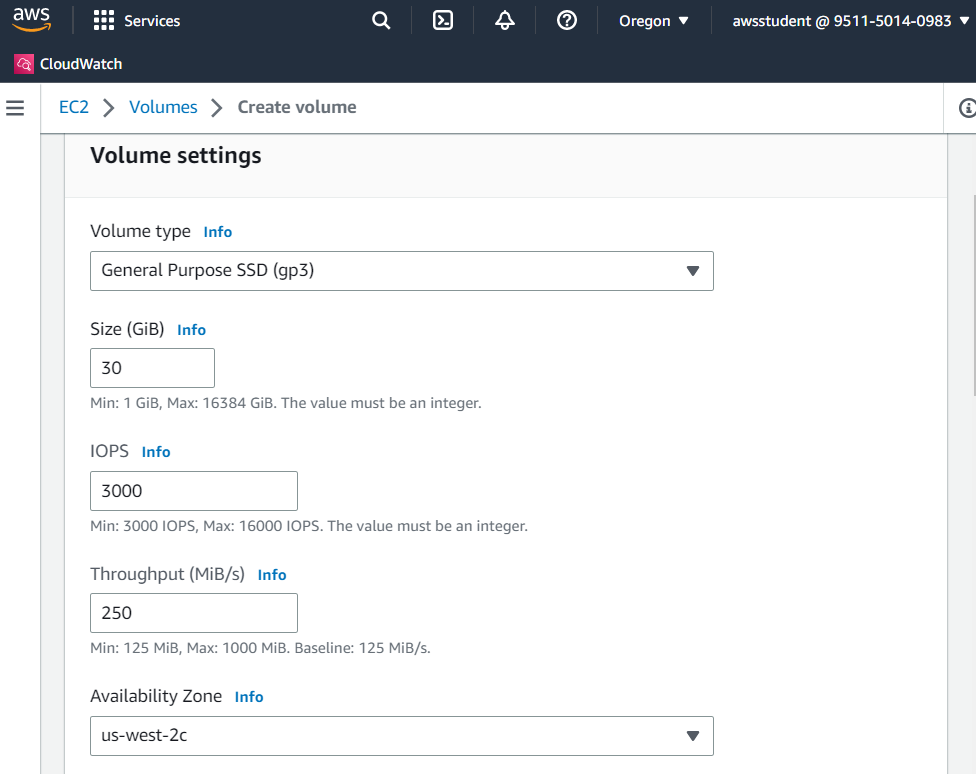
Chose the edit button and type name

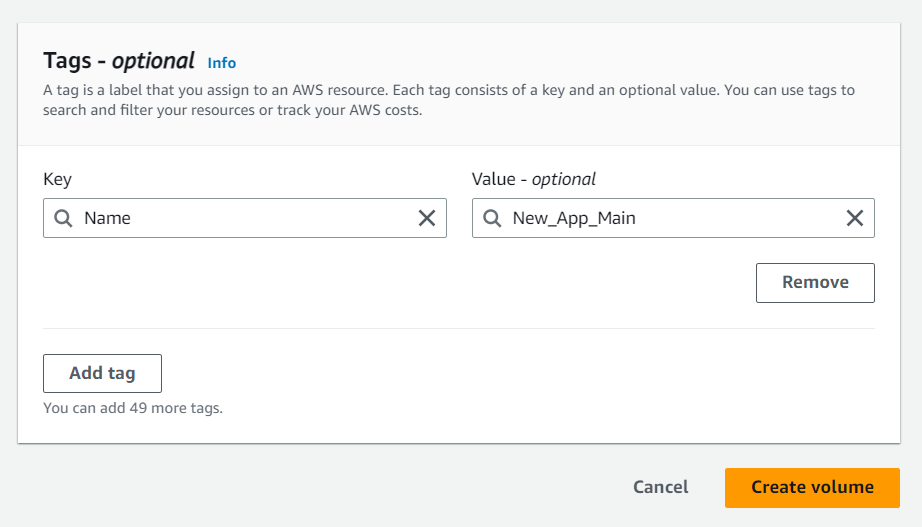


Repeat the same for the second volume and make a note from the availability zones

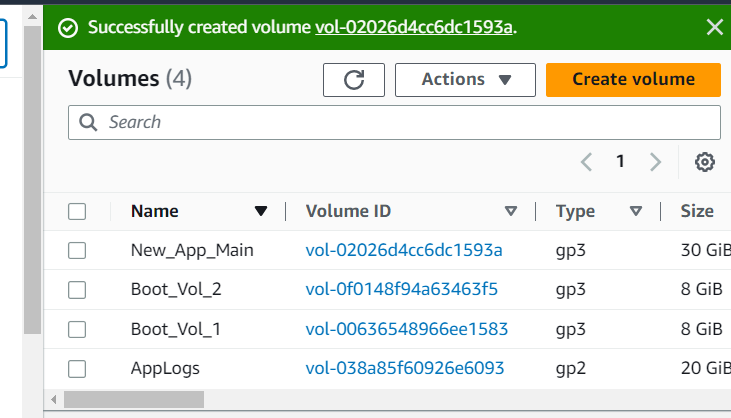


Task 1.2   
Create a new EBS volume



In the same window we add a tag 

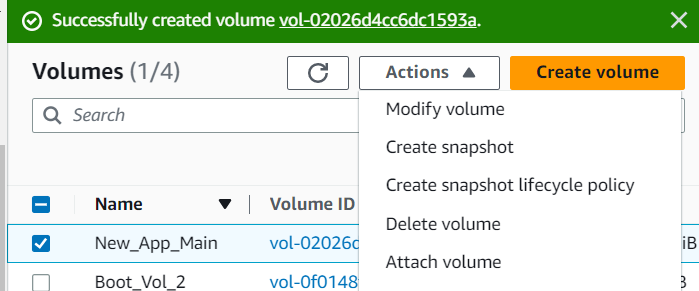
create the volume and check if that was successful



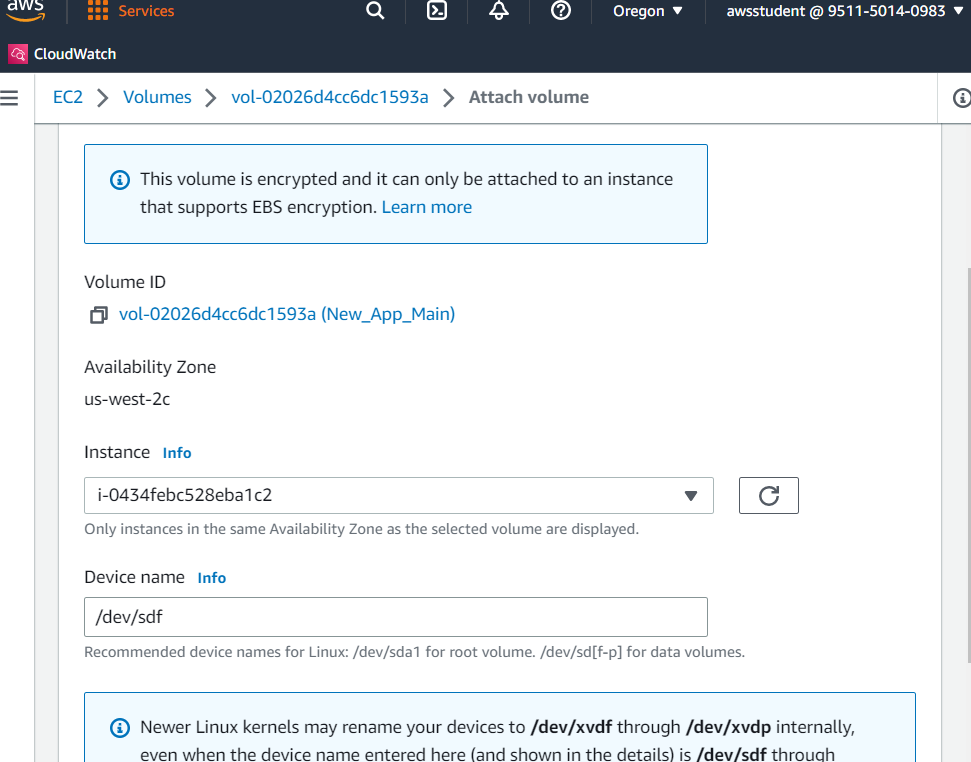
The new EBS volume is in the list as New\_App\_Main and it’s ready to be attached to EC2 instance, that way the instance can use the volume as block storage

Task 1.3

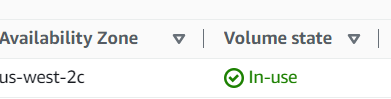
Attach EBS volume to an EC2 instance



enter the provided settings and attach volume

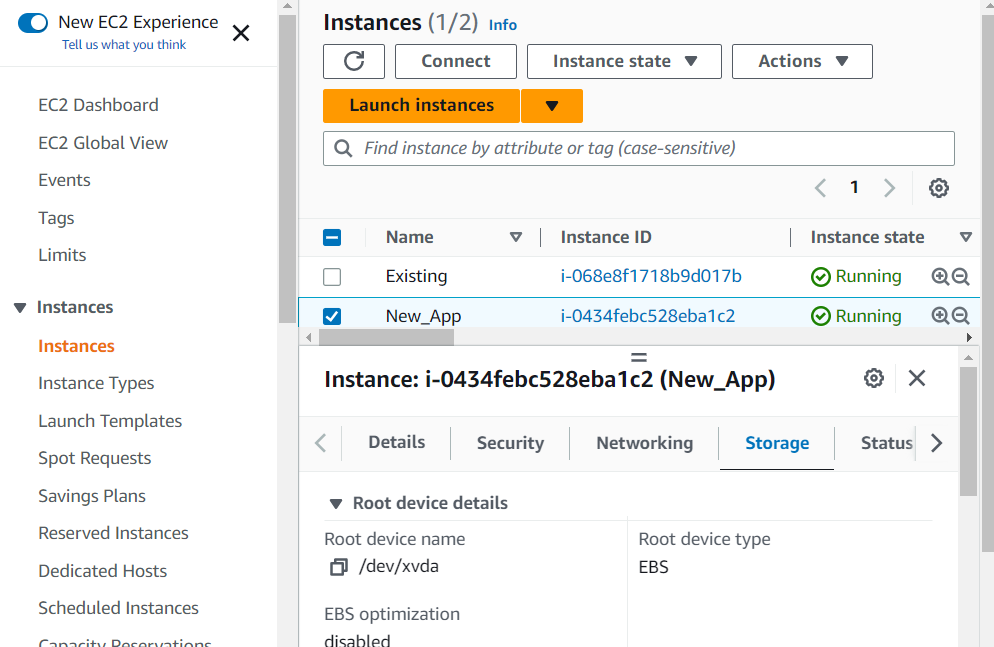


We can see that the state of the volume changed from Available to In-use



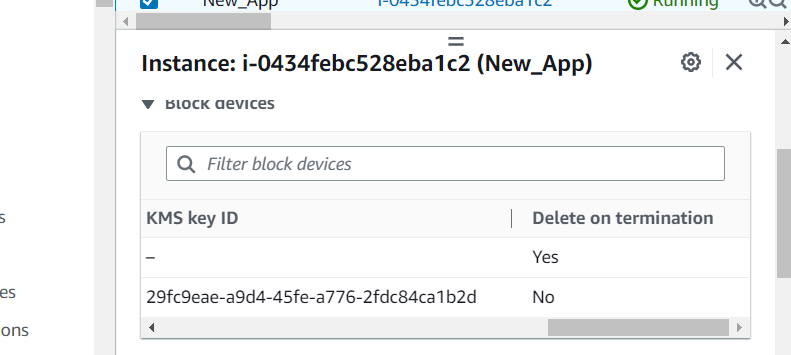
Next, examine the two volumes that are attached to EC2 instance by the name New\_App

In instances choose New\_app > storage tab



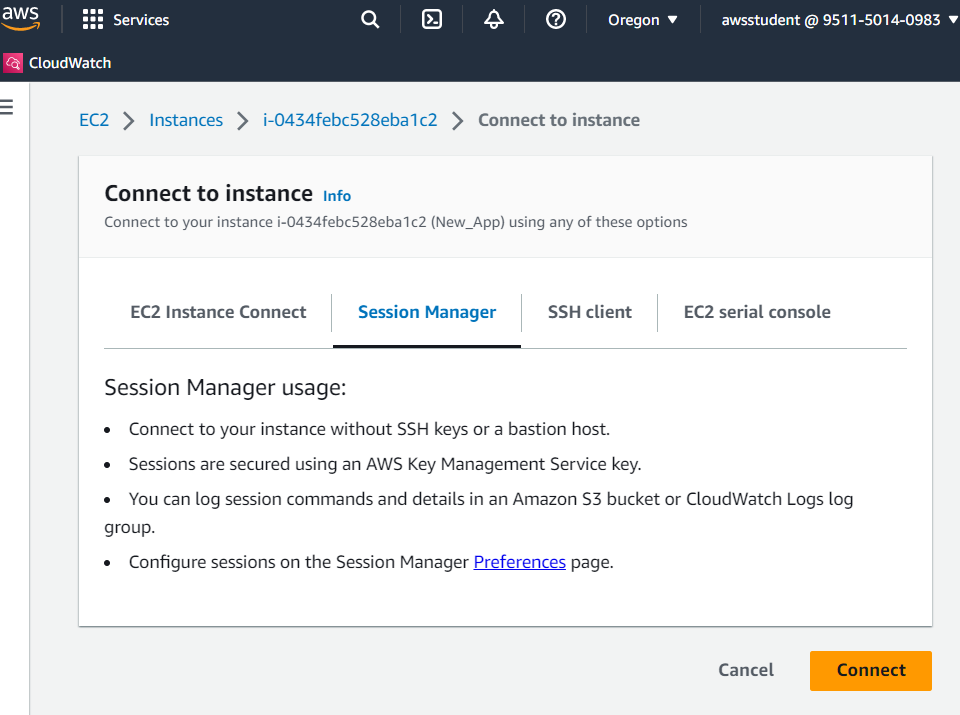
Scroll to Block devices and we can see that delete on termination is YES for /dev/xvda (in the first line) which means that the instance will be permanently deleted if the EC2 instance is terminated

And the delete termination is NO for the device name /dev/sdf which means that EBS volume will persist when the EC2 instance is terminated

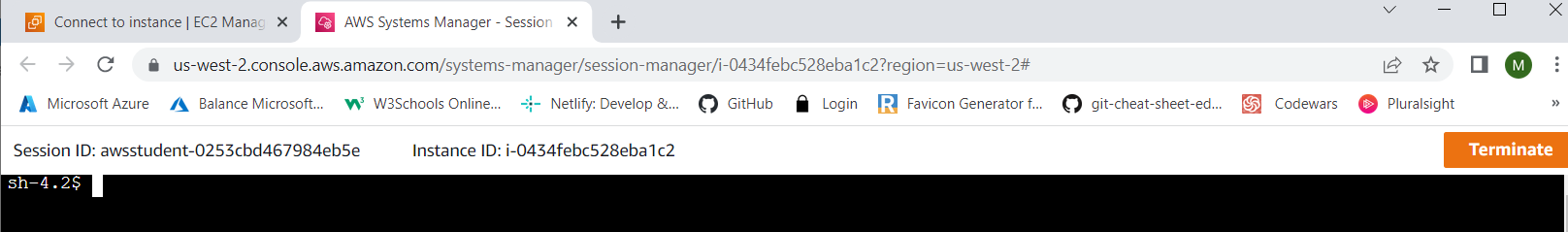


Task 2 Create and configure a file system on an attached EBS volume

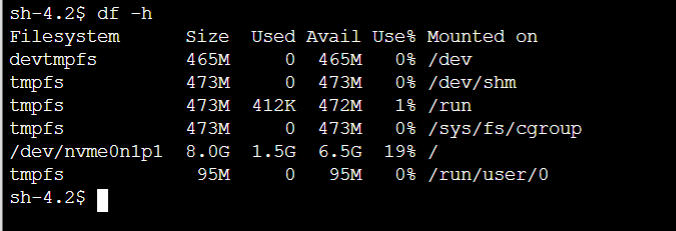
Connect to instance



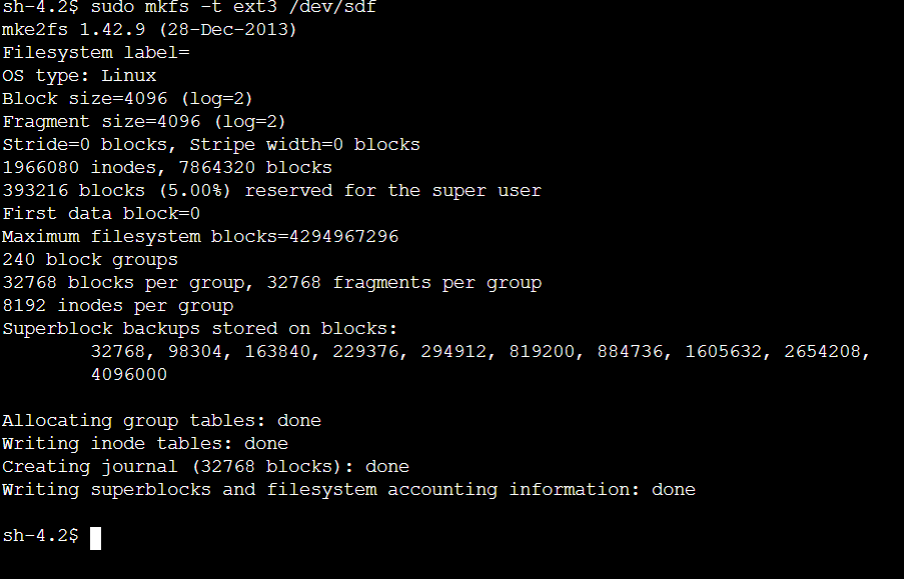
New tab will open with running terminal session



Run df –h (disk free) to view the storage that is available on the instance



Sudo mkfs –t ext /dev/sdf to create anext3 linux file system in the EBS volume

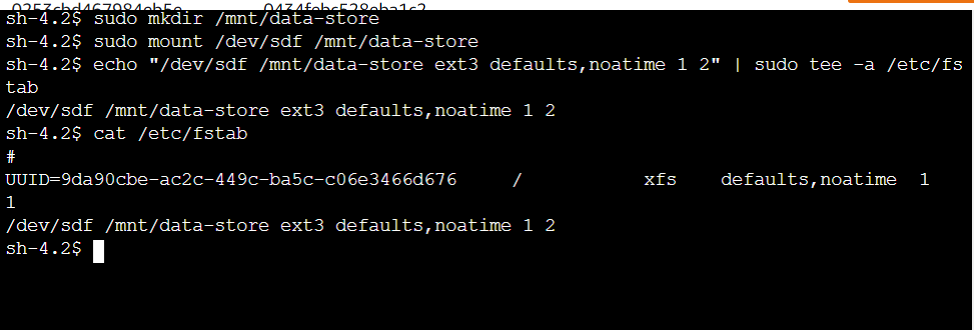


Sudo mkdir /mnt/data-store to create a new directory for mounting the new storage volume

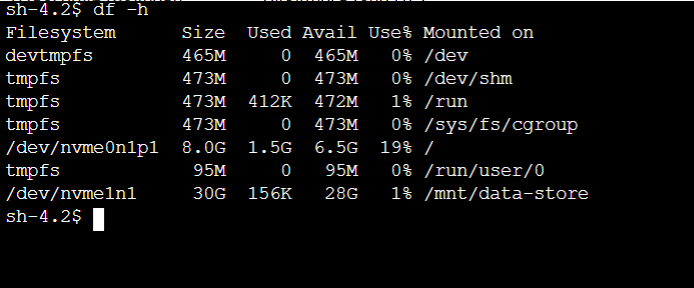
Sudo mount /dev/sdf /mnt/data/store to mount the new storage volume

echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab to add the below line to the end of /etc/fstab file

cat /etc/fstab to view that file

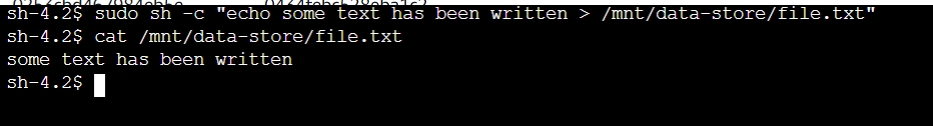


df –h to view the available storage



sudo sh -c "echo some text has been written > /mnt/data-store/file.txt" to create a new text file on the mounted volume

cat /mnt/data-store/file.txt to view the contents of the text file

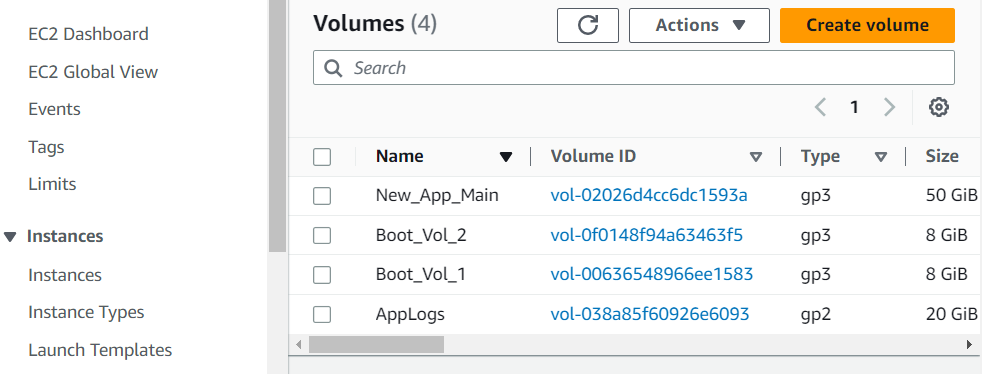


Task 3

Modify the EBS volume size and expand the file system on the volume

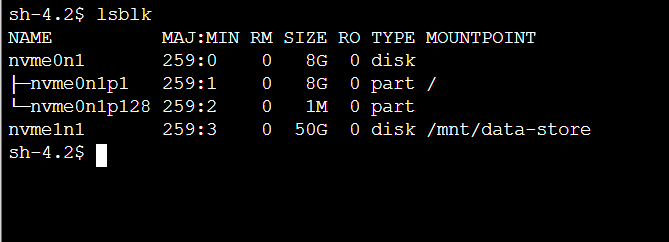
Task 3.1 Modify the size of an existing EBS volume

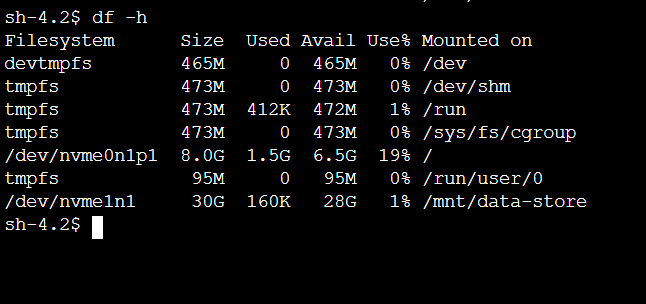
Instances>Elastic block store > volumes > new\_App\_Main and update the size to 50gb

Upgrade to 50   


Task 3.2 Expand the volume of your file system

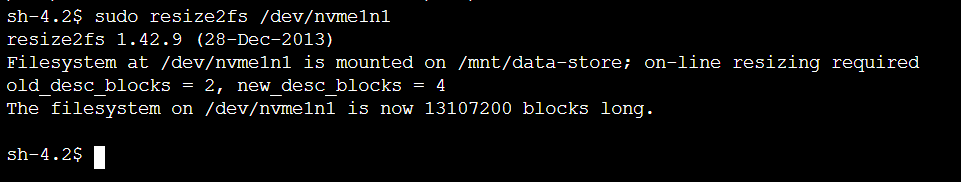
In previously opened terminal run lsblk to display the information about the block devices that are attached to the instance



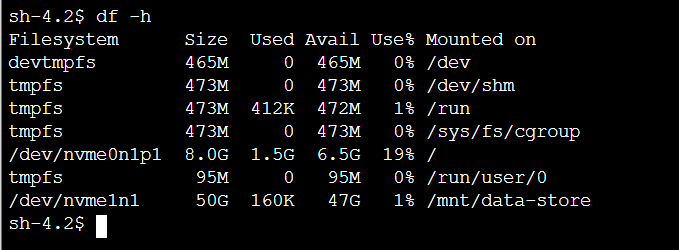
Df – h to check the size of the file system for each volume 

the size of the file system is 30gb and we need to increase the volume size

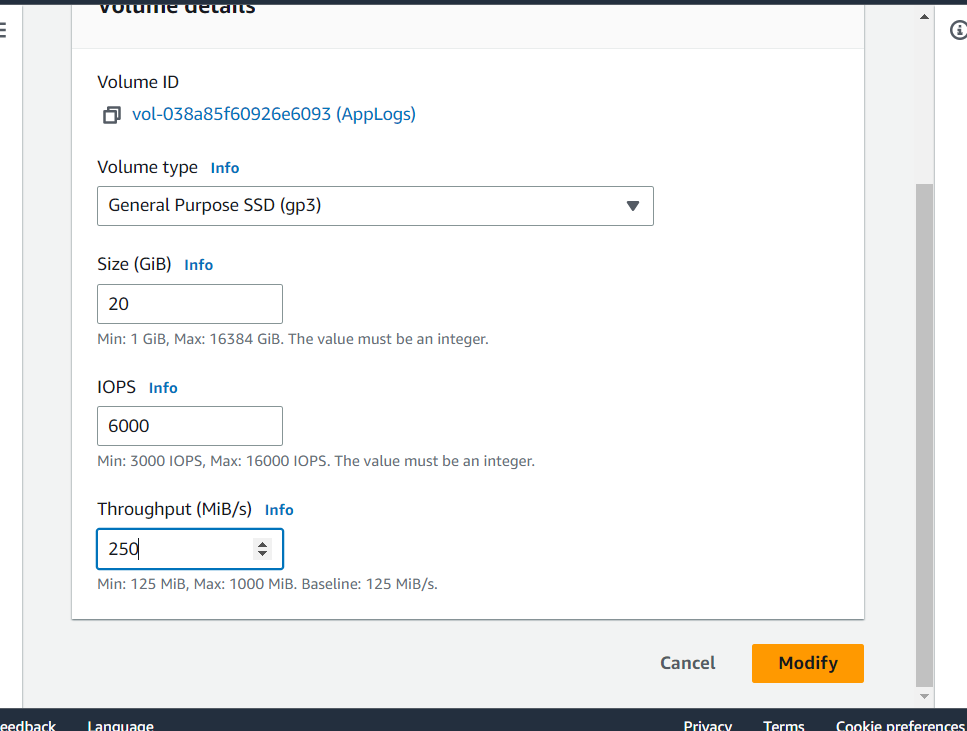
sudo resize2fs /dev/nvme1n1 will help us to extend to the available capacity



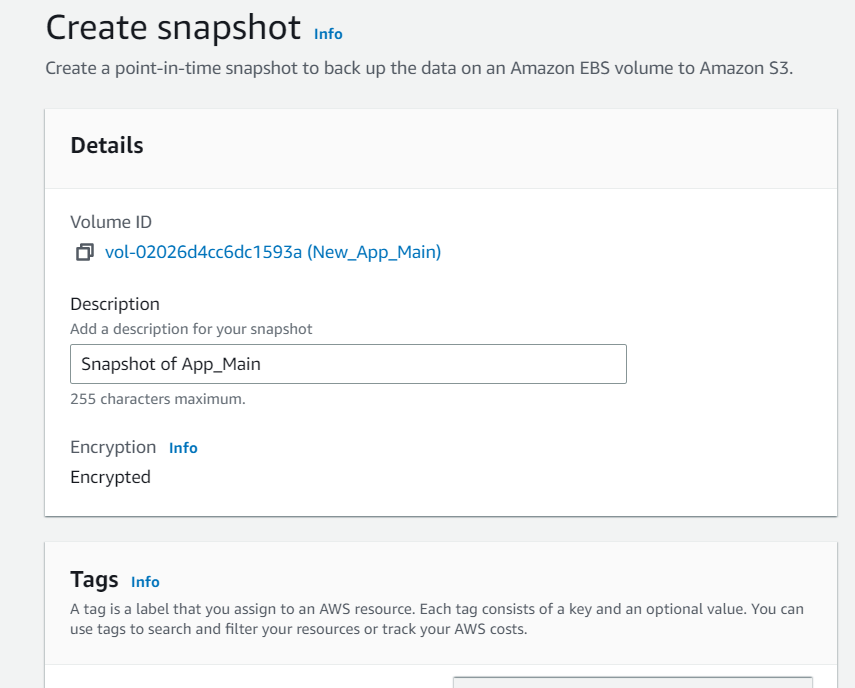
df –h check the size again



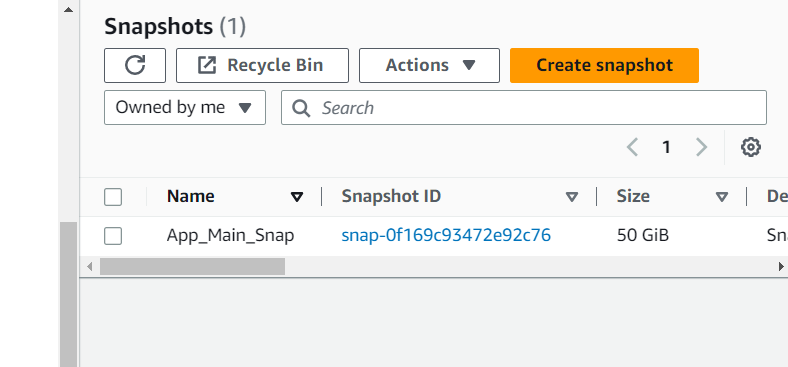
**Task 4 :** Modify the EBS volume type and provisioned performance for an existing application   
EC2 > Elastic block store > volumes > appLogs and modify with the provided settings



Task 5 : Configure a snapshot for an existing EBS volume

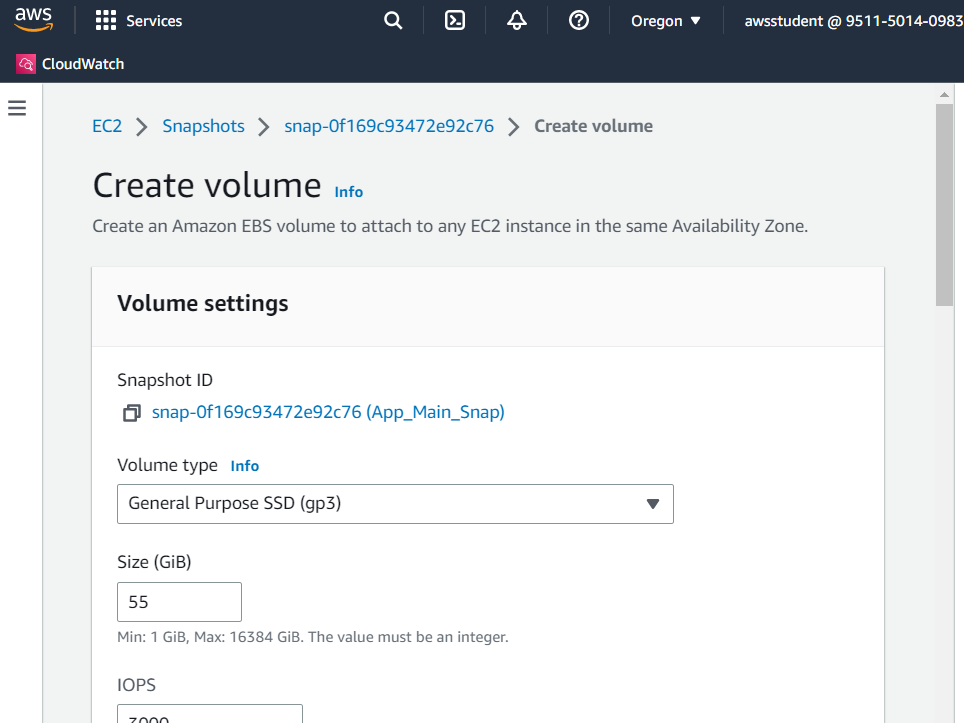
Volumes> New\_App\_Main > actions and create a snapshot with a tag

Go back to elastic block store > snapshots and check if new snapshot was created

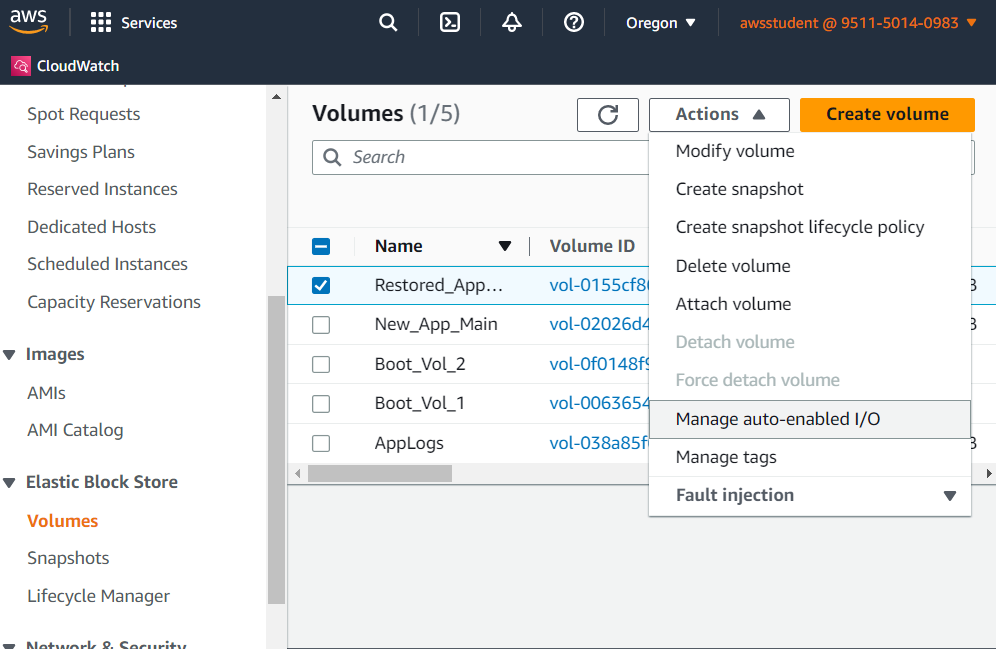


Task 6 : Restore an EBS volume from an existing snapshot

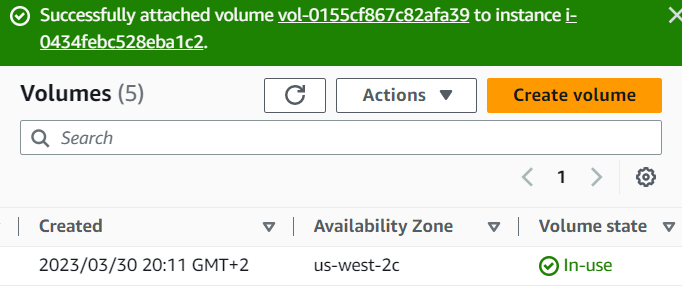
Ec2 > snapshots > create volume



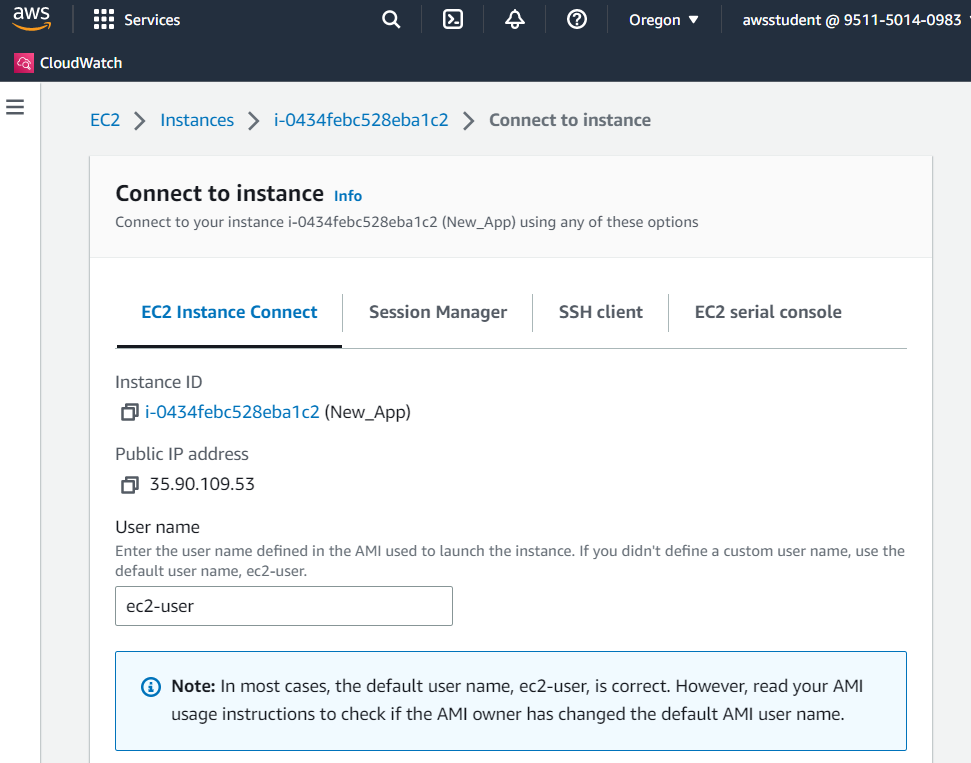
Elastic block store > volumes > Restored\_app\_main and attach volume from New\_app instance



Attach volume and the volume state will change to “in-use”



instance connect

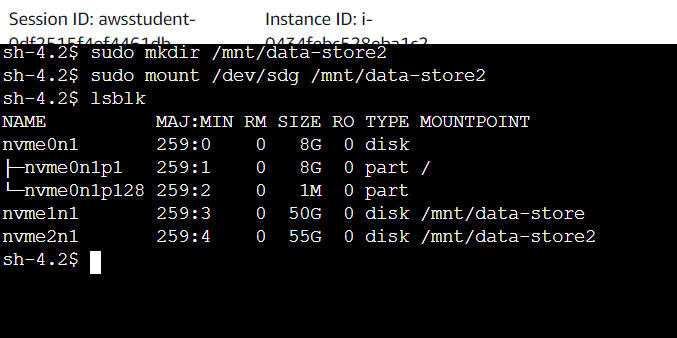


back to the terminal run

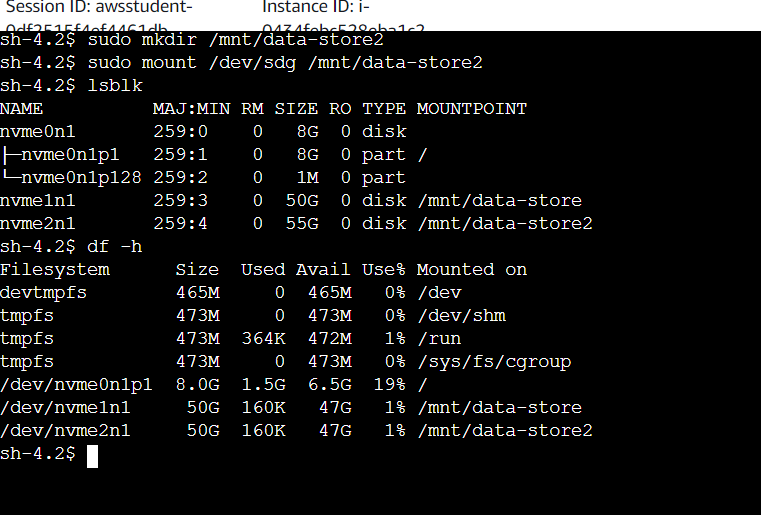
sudo mkdir /mnt/data-store2 to create a directory for mounting the restored storage volume

sudo mount /dev/sdg /mnt/data-store2 to mount the new volume

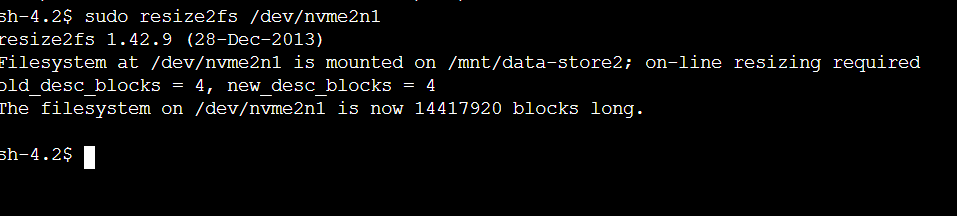
lsblk to check the block devices that are attached to the instance



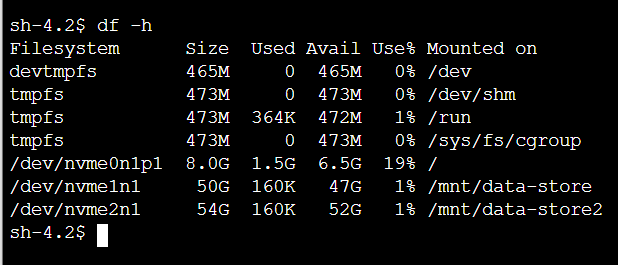
df –h to check the size again



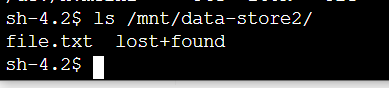
sudo resize2fs /dev/nvme2n1 to extend the file system and use the additional capacity that is available



Df – h check size



ls /mnt/data-store2/ to list the contents of the mounted volume



cat /mnt/data-store2/file.txt to check the contents of this file