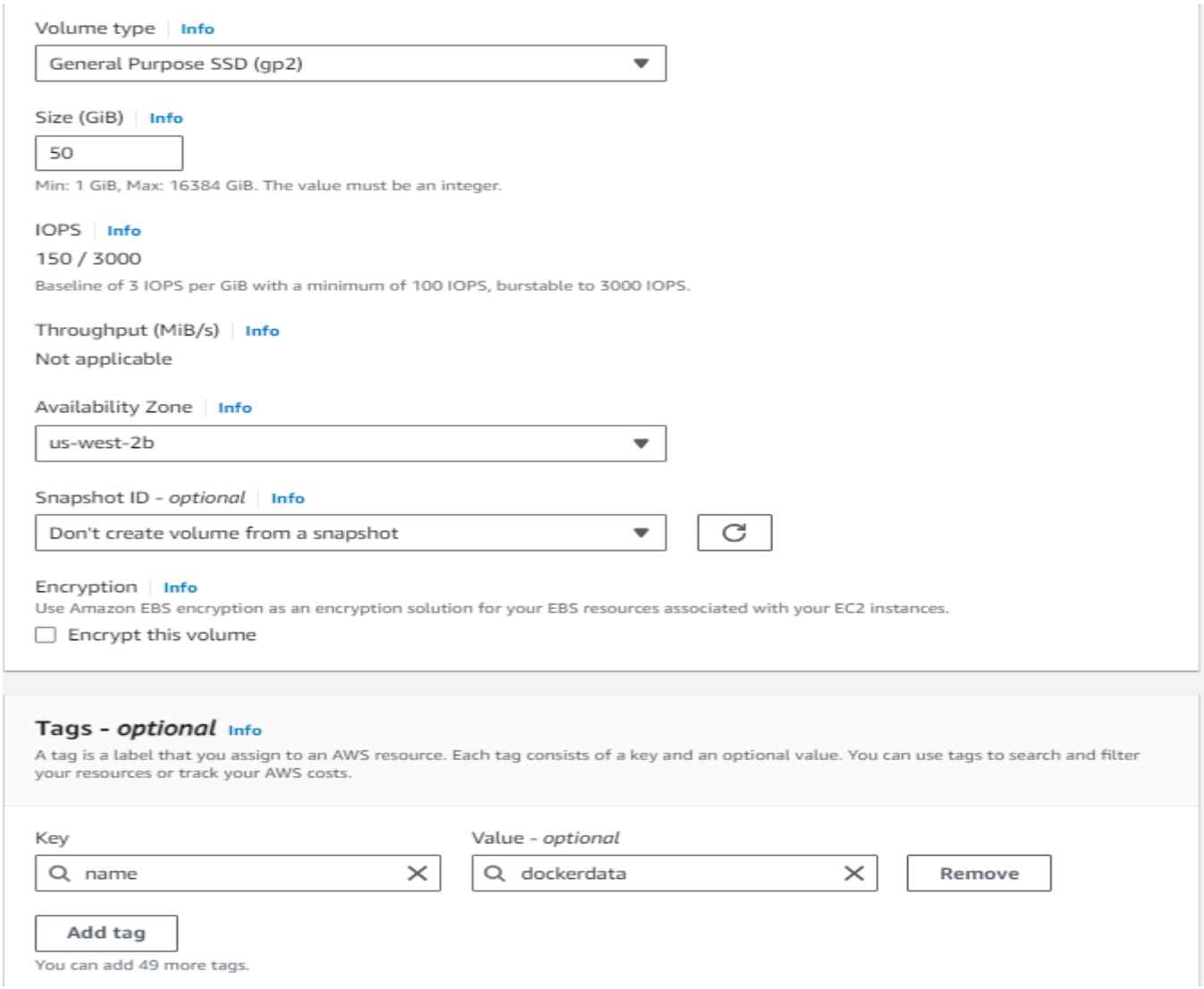
- Objective:
  - creating an EBS volume
  - attaching to our Ubuntu running Ec2 instance
  - mapping to docker to store images or data
- Task:
  - attach the EBS volume to Ec2 instance and then map it to dockerdata

Steps:

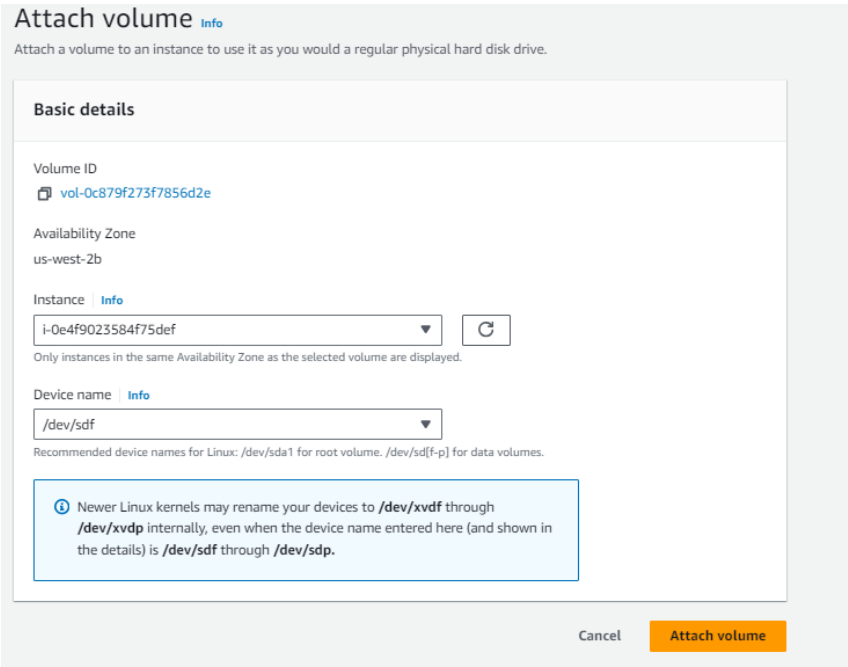
- create your Ubuntu based ec2 instance and install Docker on it
  - # sudo apt-get update
  - # sudo apt install docker -y
  - # sudo systemctl start docker
  - # sudo systemctl enable docker
  - # sudo systemctl status
  - # sudo df -h # to check disk free

```
root@ip-172-31-20-101:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  2.3G  4.5G  34% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  900K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
root@ip-172-31-20-101:/home/ubuntu#
```

- Create EBS volume as per your requirement(make sure ec2 and ebs vol are in same subnet)



- Once created it available to attach to your ec2 instance



- Now check if the attached vol is visible or not
  - # lsblk (in this case your will see xvdf EBS vol extra that we have attached)

```
root@ip-172-31-20-101:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  2.3G  4.5G  34% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  900K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
root@ip-172-31-20-101:/home/ubuntu# lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0 25.2M  1 loop /snap/amazon-ssm-agent/7993
loop1      7:1    0 55.7M  1 loop /snap/core18/2829
loop2      7:2    0 38.8M  1 loop /snap/snapd/21759
xvda     202:0    0    8G  0 disk
├─xvda1  202:1    0    7G  0 part /
├─xvda14 202:14   0    4M  0 part
├─xvda15 202:15   0  106M  0 part /boot/efi
└─xvda16 259:0    0  913M  0 part /boot
xvdf     202:80   0   50G  0 disk
root@ip-172-31-20-101:/home/ubuntu#
```

- Now run below command to format and map it to the folder
  - # fdisk /dev/xvdf
  - # lsblk
  - # mkfs.ext4 /dev/xvdf1
  - Copy the UUID generated (in this case UUID: 0a34a11f-87d3-40e1-b845-2af06791d14e)
  - Create the dir that you want to map to attached vol (# mkdir /dockerdata)
  - Add the UUID using #vim /etc/fstab
    - UUID=0a34a11f-87d3-40e1-b845-2af06791d14e /dockerdata ext4
  - # mount –a
  - # lsblk

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-104857599, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-104857599, default 104857599):

Created a new partition 1 of type 'Linux' and of size 50 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
root@ip-172-31-20-101:/home/ubuntu# lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0 25.2M  1 loop /snap/amazon-ssm-agent/7993
loop1      7:1    0 55.7M  1 loop /snap/core18/2829
loop2      7:2    0 38.8M  1 loop /snap/snapd/21759
xvda     202:0    0    8G  0 disk
├─xvda1  202:1    0    7G  0 part /
├─xvda14 202:14   0    4M  0 part
├─xvda15 202:15   0  106M  0 part /boot/efi
└─xvda16 259:0    0  913M  0 part /boot
xvdf     202:80   0   50G  0 disk
└─xvdf1  202:81   0   50G  0 part
```

```
root@ip-172-31-20-101:/home/ubuntu# mkfs.ext4 /dev/xvdf1
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 13106944 4k blocks and 3276800 inodes
Filesystem UUID: 0a34a11f-87d3-40e1-b845-2af06791d14e
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
root@ip-172-31-20-101:/# mkdir /dockerdata
root@ip-172-31-20-101:/# ls
bin  bin.usr-is-merged  boot  dev  dockerdata  etc
root@ip-172-31-20-101:/# pwd
/
root@ip-172-31-20-101:/#
```

```
root@ip-172-31-20-101:/# cat /etc/fstab
LABEL=cloudimg-rootfs   /        ext4   discard,commit=30,errors=remount-ro     0 1
LABEL=BOOT      /boot   ext4    defaults        0 2
LABEL=UEFI      /boot/efi       vfat    umask=0077      0 1
UUID=0a34a11f-87d3-40e1-b845-2af06791d14e /dockerdata ext4
root@ip-172-31-20-101:/#
```

```
root@ip-172-31-20-101:/# mount -a
mount: (hint) your fstab has been modified, but systemd still uses
       the old version; use 'systemctl daemon-reload' to reload.
root@ip-172-31-20-101:/# df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/root        6.8G  2.3G  4.5G  34% /
tmpfs            479M     0  479M   0% /dev/shm
tmpfs            192M  904K  191M   1% /run
tmpfs            5.0M     0  5.0M   0% /run/lock
/dev/xvda16      881M   76M  744M  10% /boot
/dev/xvda15      105M  6.1M   99M   6% /boot/efi
tmpfs             96M   12K   96M   1% /run/user/1000
/dev/xvdf1        49G   24K   47G   1% /dockerdata
```

- After doing above task now do configuration change in docker
  - #sudo systemctl stop docker.service
  - #sudo systemctl stop docker.socket
  - #sudo vim /lib/systemd/system/docker.service
  - Edit highlighted part:
    - ExecStart=/usr/bin/dockerd **--data-root /dockerdata** -H fd:// -- containerd=/run/containerd. sock
  - sudo rsync -aqxP /var/lib/docker/ /dockerdata
  - sudo systemctl daemon-reload && sudo systemctl start docker
  - sudo systemctl status docker --no-pager
  - ps aux | grep -i docker | grep -v grep

Once executed successful then you are able to see the below

```
root@ip-172-31-20-101:/# ps aux | grep -i docker | grep -v grep
root        7212  0.0  7.4 1830416 73444 ?       Ssl  14:24   0:00 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
root@ip-172-31-20-101:/#
```

- check it
  - docker run --rm -d --name app1 -p 8000:80 kiran2361993/troubleshootingtools:v1
  - df -h

```
root@ip-172-31-20-101:/# docker run --rm -d --name app1 -p 8000:80  kiran2361993/troubleshootingtools:v1
Unable to find image 'kiran2361993/troubleshootingtools:v1' locally
v1: Pulling from kiran2361993/troubleshootingtools
171857c49d0f: Pull complete
419640447d26: Pull complete
61e52f862619: Pull complete
78a19a6c959a: Pull complete
60967de56c4c: Pull complete
56a967db6e41: Pull complete
c8b1454d39d1: Pull complete
88eba6dadadd: Pull complete
f3bc3aa8b626: Pull complete
70c3609ec292: Pull complete
cbd12a390544: Pull complete
8db040f432b1: Pull complete
4db74f7c22ef: Pull complete
c11eb9b9792b: Pull complete
Digest: sha256:b30ef49139a7191c59b26d23c7564c930ecd119d23781ddb902c7256a6db82fc
Status: Downloaded newer image for kiran2361993/troubleshootingtools:v1
66b53e8cc94bc61053f8abdcfd86c86d812ef68027279b4a9d81d24e73c8a37a
root@ip-172-31-20-101:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  3.9G  2.9G  58% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  988K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
/dev/xvdf1       49G  328K   47G   1% /dockerdata
overlay         6.8G  3.9G  2.9G  58% /var/lib/docker/overlay2/1c1341045c10800b0199b69cf3ec2297849f37509fdb885718c215281bc98a4b/merged
root@ip-172-31-20-101:/# /dockerdata/
bash: /dockerdata/: Is a directory
root@ip-172-31-20-101:/# cd /dockerdata/
root@ip-172-31-20-101:/dockerdata# ls
buildkit  containers  engine-id  image  lost+found  network  overlay2  plugins  runtimes  swarm  tmp  volumes
root@ip-172-31-20-101:/dockerdata#
```