EBS Volume Replication and Expansion

Problem Statement:

You work for XYZ Corporation. Your corporation is working on an application and they require secured web servers on Linux to launch the application.

You have been asked to:

1. Create an Instance in the us-east-1 (N. Virginia) region with Linux OS and manage the requirement of web servers of your company using AMI

2. Replicate the instance in the us-west-2 (Oregon) region

3. Build two EBS volumes and attach them to the instance in the us-east-1 (N. Virginia) region

4. Delete one volume after detaching it and extend the size of the other volume

5. Take a backup of this EBS volume

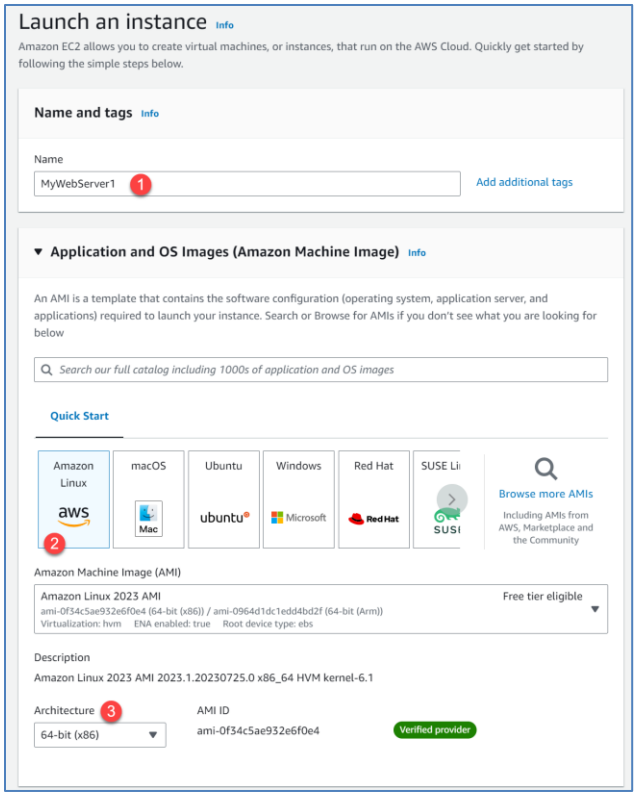
Answer:

Login to AWS Management console

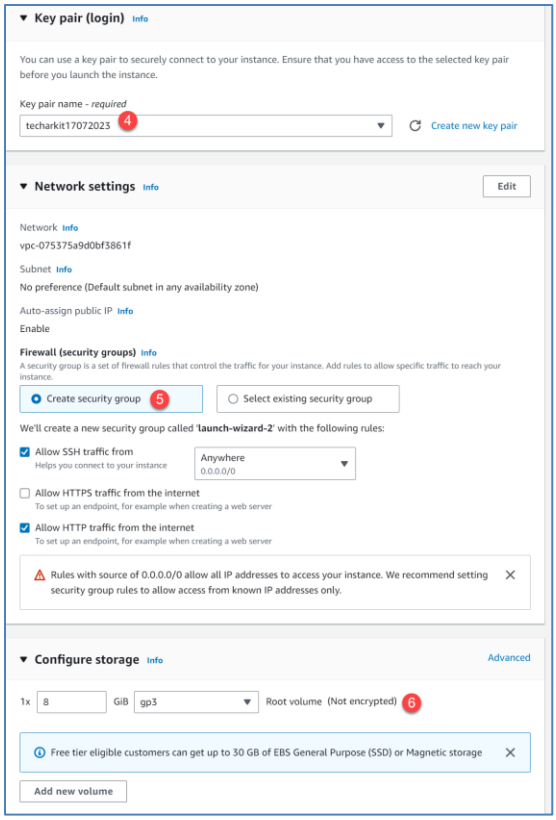
https://aws.amazon.com/console/

Go to EC2 service → Instances → Launch instances →

Created by TechArkit Youtube Channel. Follow us for more videos and Documents:
https://www.youtube.com/@TechArkit/videos



1. Provide Instance a name
2. Select Operating system
3. Select Architecture

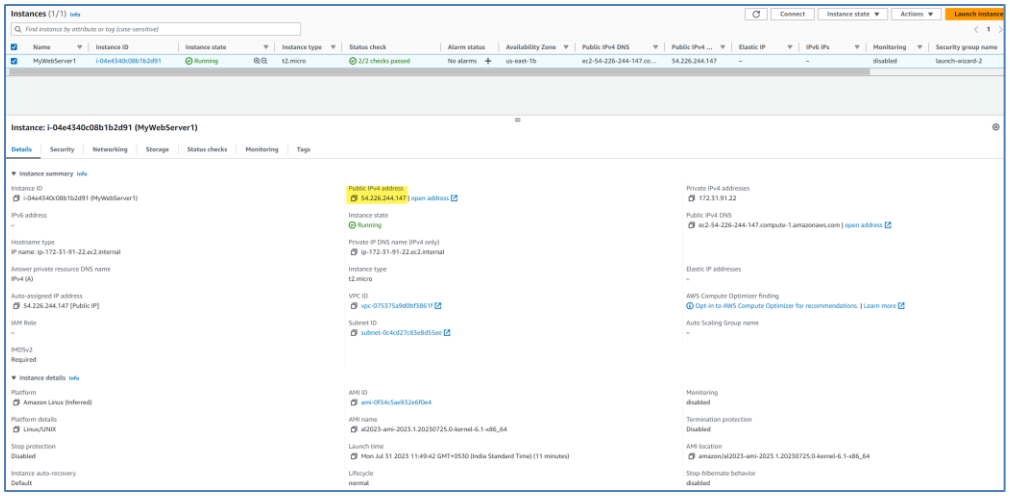


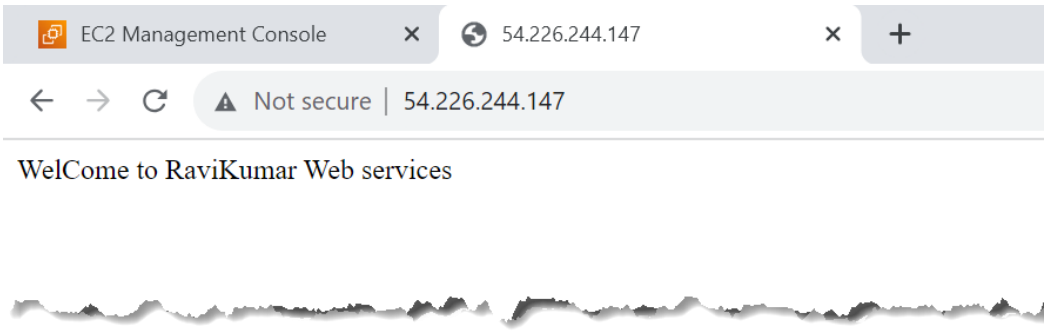
4. Select Key pair
5. Create Security Group and allow SSH and HTTP from anywhere
6. EBS volume (If require add another)

Advanced settings add user data as above to install and start web server.

Click on “Launch instance”

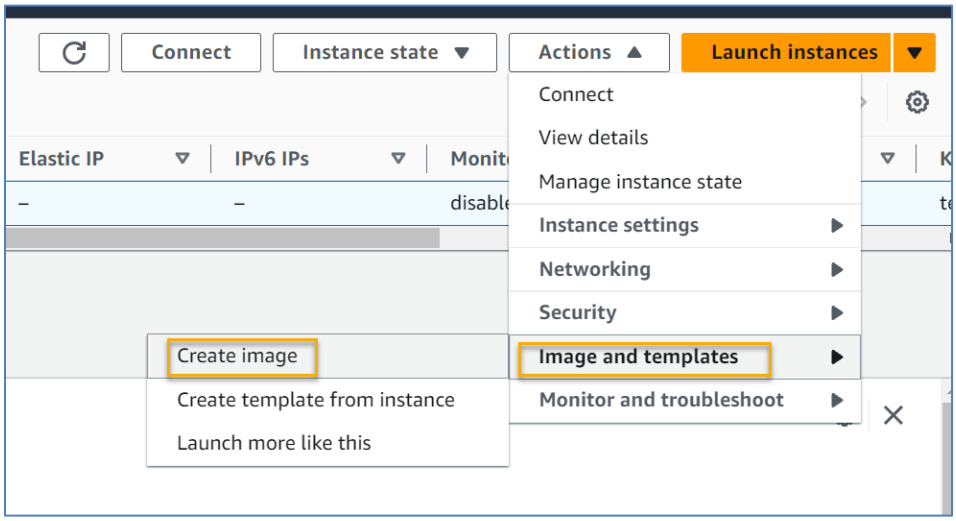
Instance has been launched in N. Virginia region successfully.

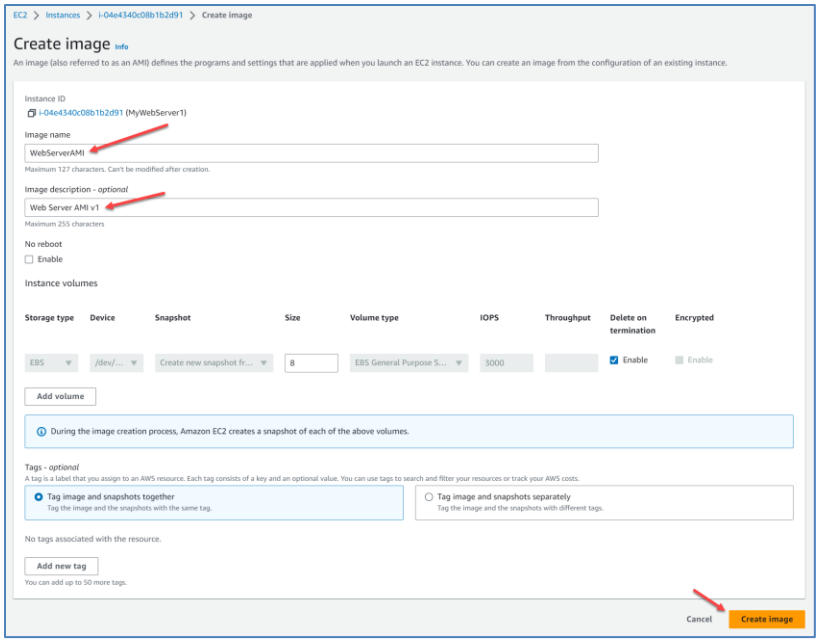




Web server is accessible from public facing.

Select the instance you wanted to copy to another region then **Actions → Image and templates → Create Image**



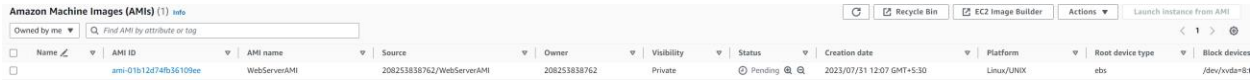


Provide Image name

Image Description

Note: Do not select No Reboot option if you wanted to have consistent AMI. No Reboot option will create consistent issues some times.

Click on “Create Image”



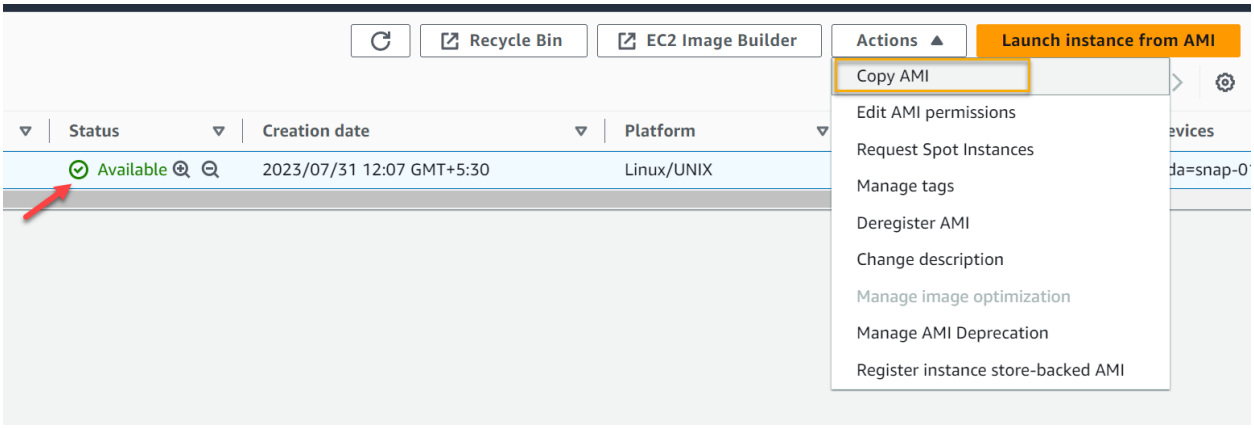
Wait for a few minutes to create the AMI.

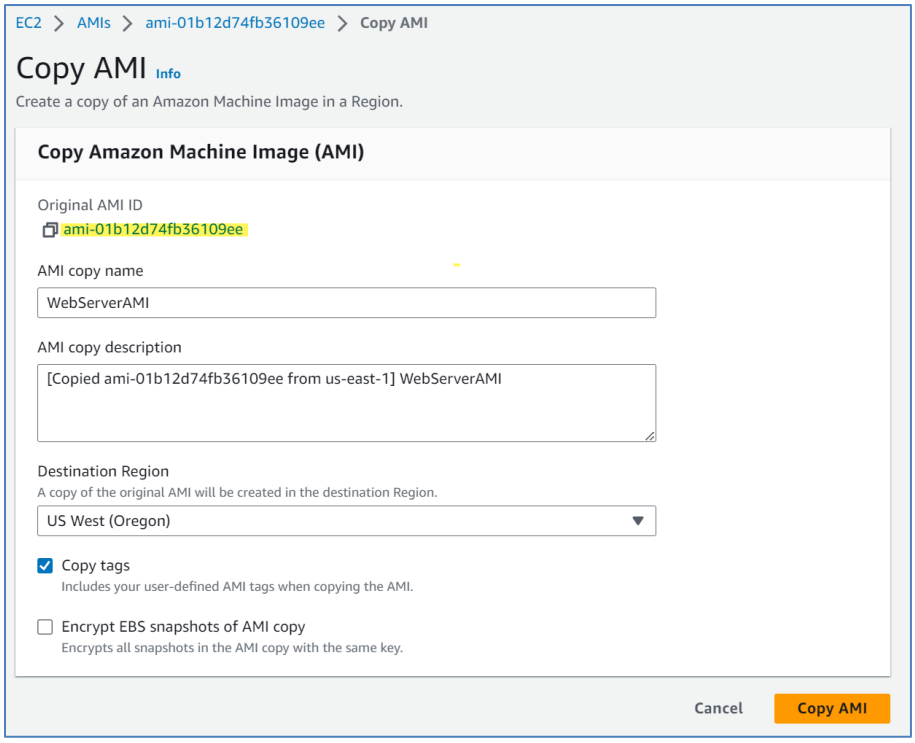
Note: AMI Creation takes time depending on instance data size.

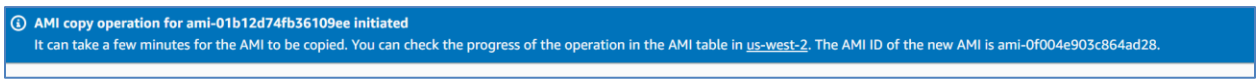
Once the AMI state is available then select the AMI

EC2 Dashboard → Images → AMIs → Select the AMI → Actions → Copy AMI

Created by TechArkit Youtube Channel. Follow us for more videos and Documents:
https://www.youtube.com/@TechArkit/videos

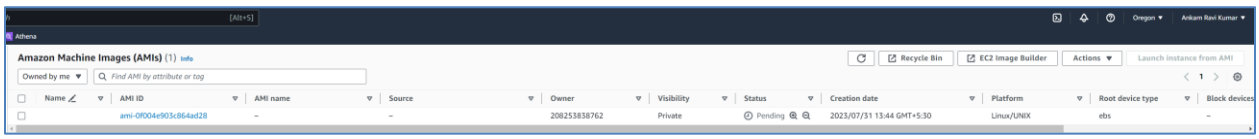


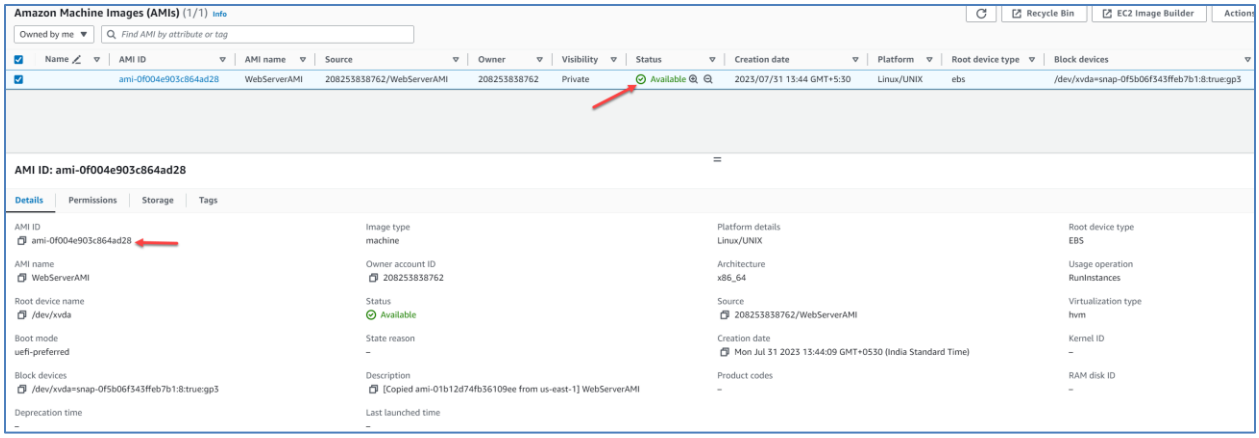




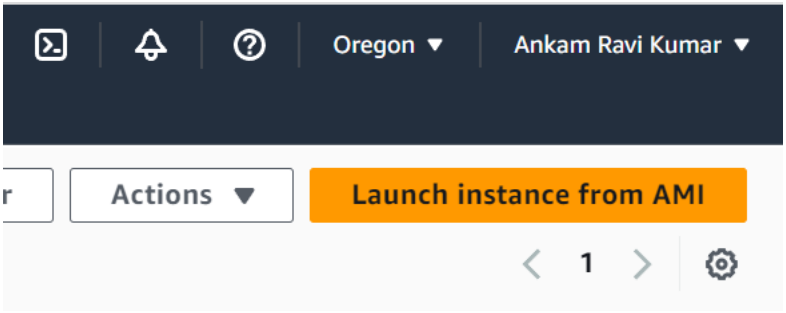
Amazon Machine Image is still pending

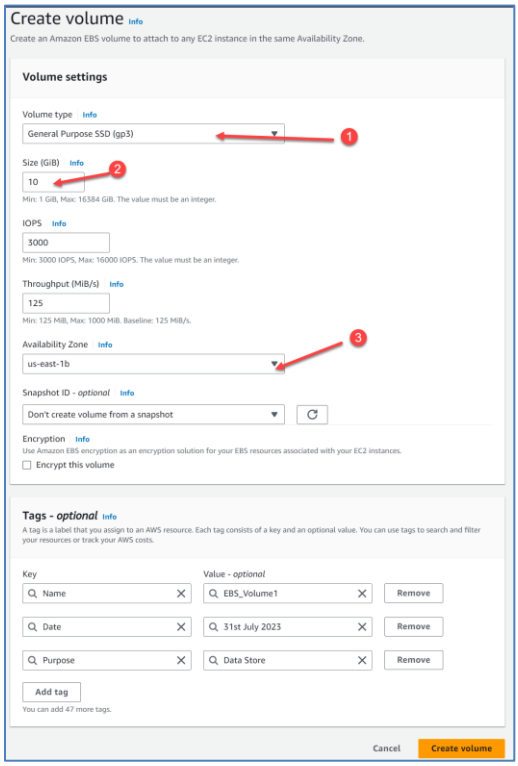
Wait for a few minutes to complete the copy.

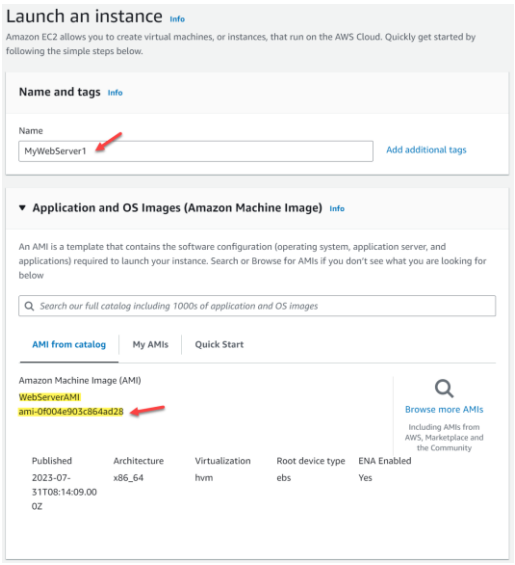




Now AMI is available in Oregon region, Lets spin the EC2 instance using the backup AMI.







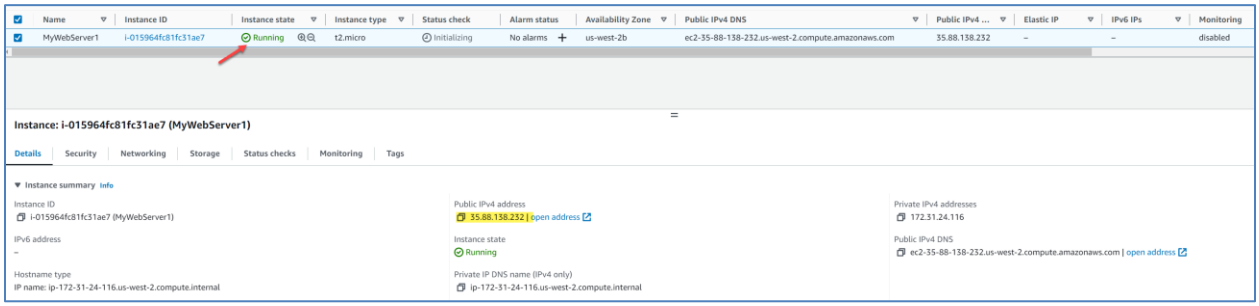
Provide new EC2 instance name

Validate your launching from the correct AMI

Select the Key pair and Security Group

Allow SSH and HTTP from anywhere to access the EC2 instance as well as browse the Webserver

Click "**Launch instance**"



Grab the instance public IP Address and check web server is working fine.

Successfully copied the EX2 instance from N.Virginia to Oregon

Step 3: Create EBS volumes

To attach the EBS volumes first we need to create two EBS volumes and follow the steps

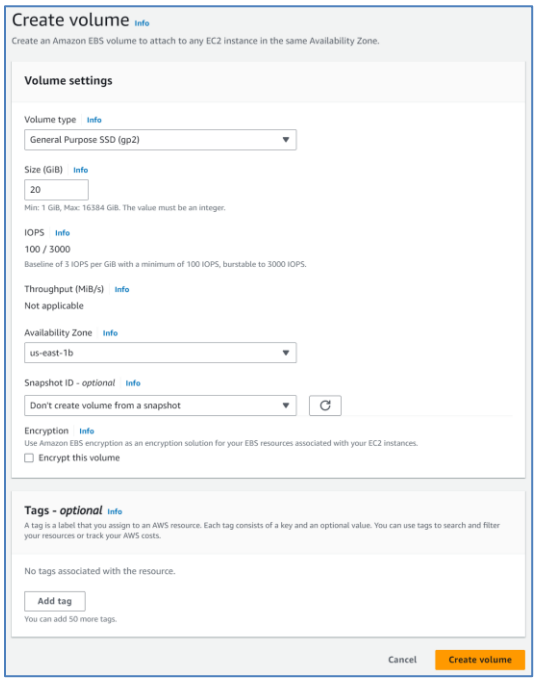
Note: Please check the instance AZ details before going to create volume.

We have to create the volume in the same AZ where instance is running.

EC2 → Elastic Block Store → Volumes → Create volume

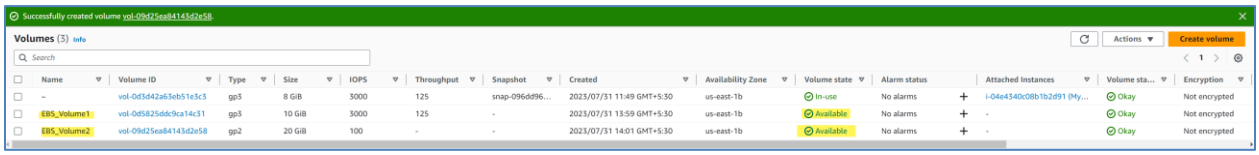
1. Select the volume type (This is based on application required performance (I/O))
2. Provide Size
3. Availability Zone if you select wrong AZ you can't attach to instance

Click "Create volume"



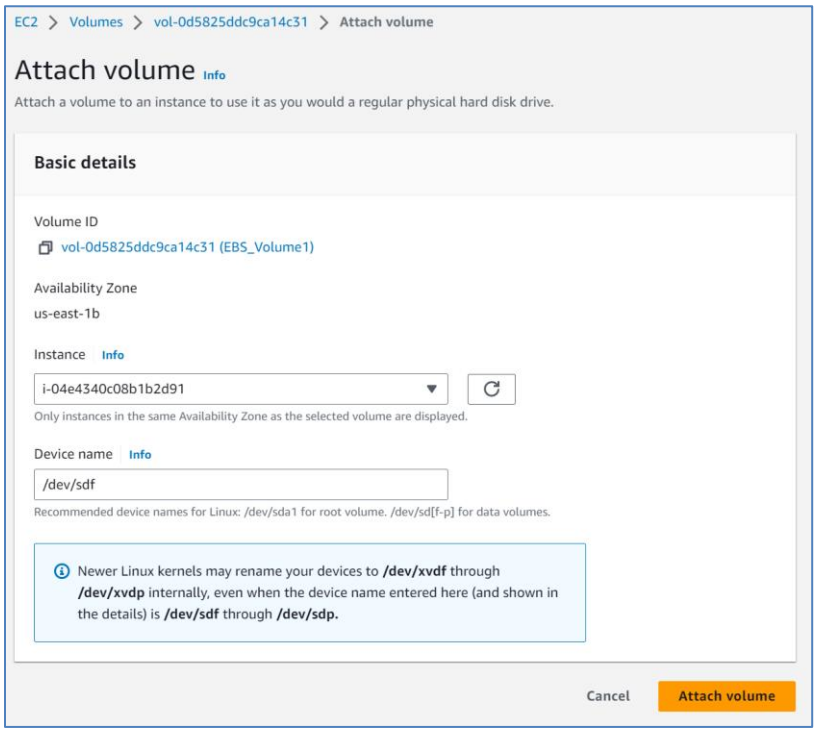
Similarly create another volume this time 20GB in size

Wait for a few seconds to EBS volume to come in Available status



Now select the volume and attach to the instance

Select EBS volume → Actions → Attach volume



Click "**Attach volume**"

Once you attach the volume state becomes In-use

In similar way attach another EBS volume.

Login to EC2 instance

```
[ec2-user@ip-172-31-91-22 ~]$ sudo -s
[root@ip-172-31-91-22 ec2-user]# 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  10G  0 disk
xvdg      202:96   0  20G  0 disk
[root@ip-172-31-91-22 ec2-user]# mkdir /volume1 /volume2
[root@ip-172-31-91-22 ec2-user]# mkfs.xfs /dev/xvdf
meta-data=/dev/xvdf              isize=512    agcount=4, agsize=655360 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=2621440, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-172-31-91-22 ec2-user]# mkfs.xfs /dev/xvdg
meta-data=/dev/xvdg              isize=512    agcount=4, agsize=1310720 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=5242880, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-172-31-91-22 ec2-user]# mount /dev/xvdf /volume1
[root@ip-172-31-91-22 ec2-user]# mount /dev/xvdg /volume2
[root@ip-172-31-91-22 ec2-user]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  2.8M  188M   2% /run
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs           475M     0  475M   0% /tmp
tmpfs            95M     0   95M   0% /run/user/1000
/dev/xvdf        10G  104M  9.9G   2% /volume1
/dev/xvdg        20G  175M   20G   1% /volume2
[root@ip-172-31-91-22 ec2-user]#
```

`sudo -s` to switch to root user to do the activity

`lsblk` to identify the disk paths

`mkdir /volume1 /volume2`

`mkfs.xfs /dev/xvdf` #To make file system in xvdf disk

`mkfs.xfs /dev/xvdg` #To make file system in xvdg disk

`mount /dev/xvdf /volume1` #To mount the disk

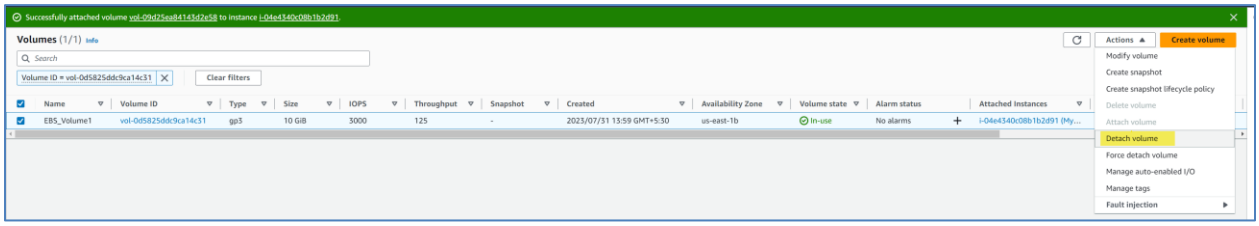
`df -h` # To check the mount point status

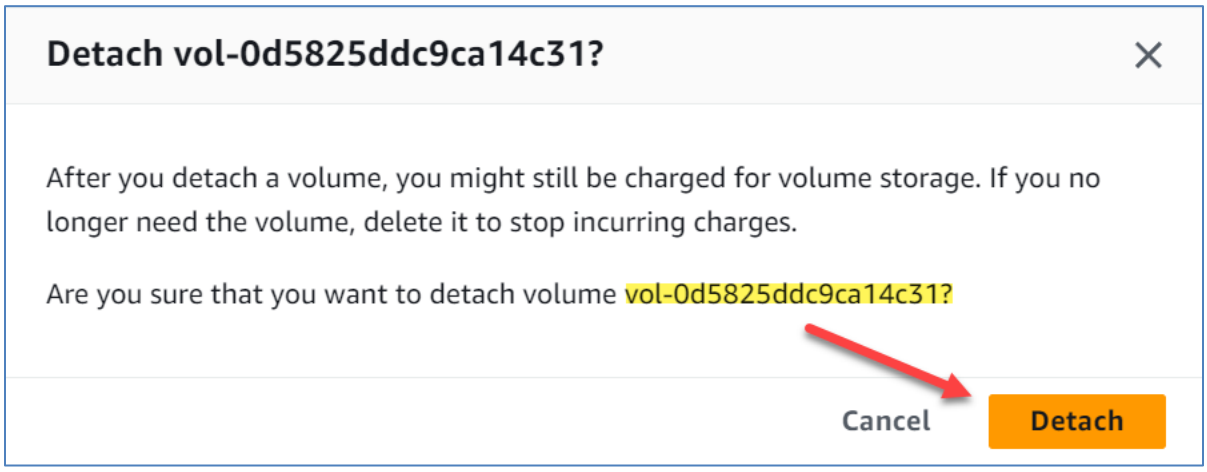
Remove Disk from EC2 instance

Un-mount the disk using a below command

`umount -l /volume1`

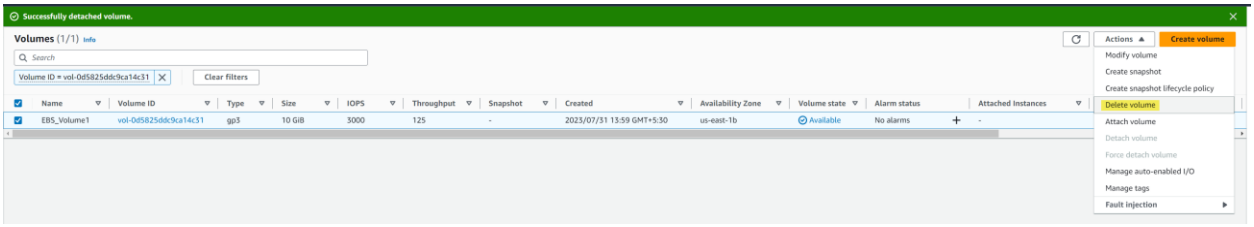
Now detach volume from the instance

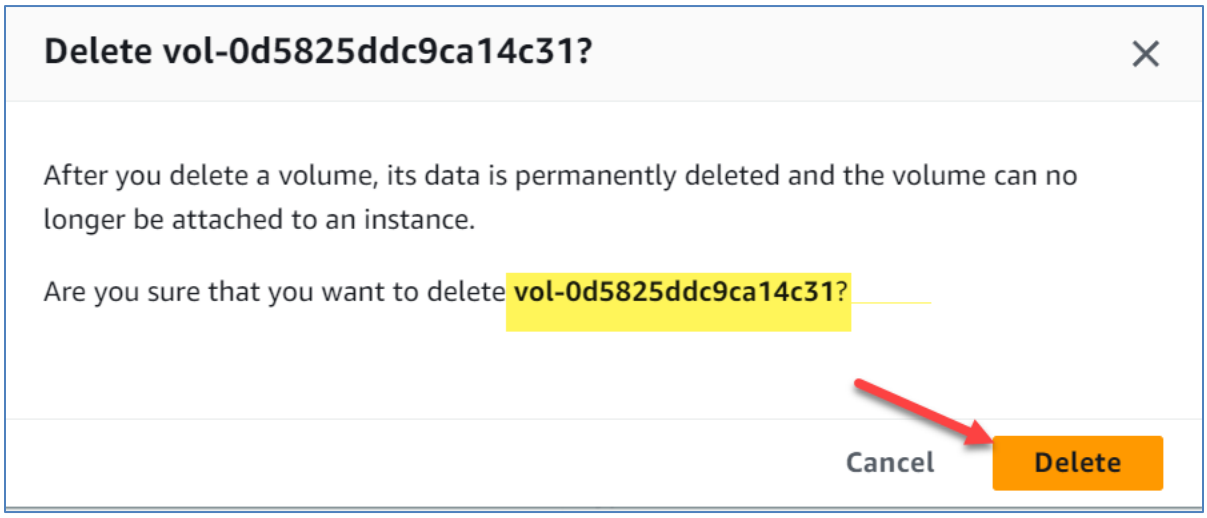




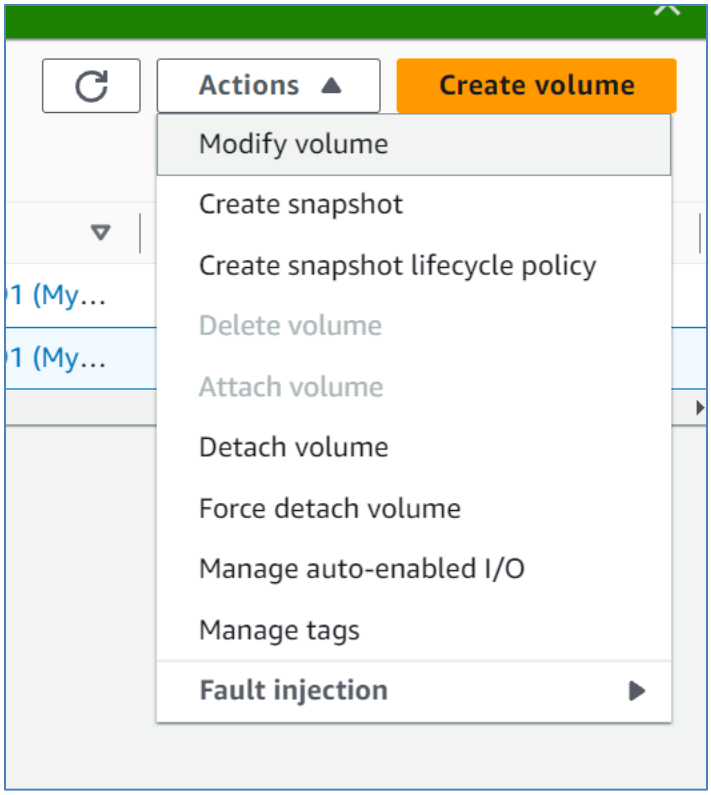
Validate the volume ID before you click on detach

Created by TechArkit Youtube Channel. Follow us for more videos and Documents:
https://www.youtube.com/@TechArkit/videos

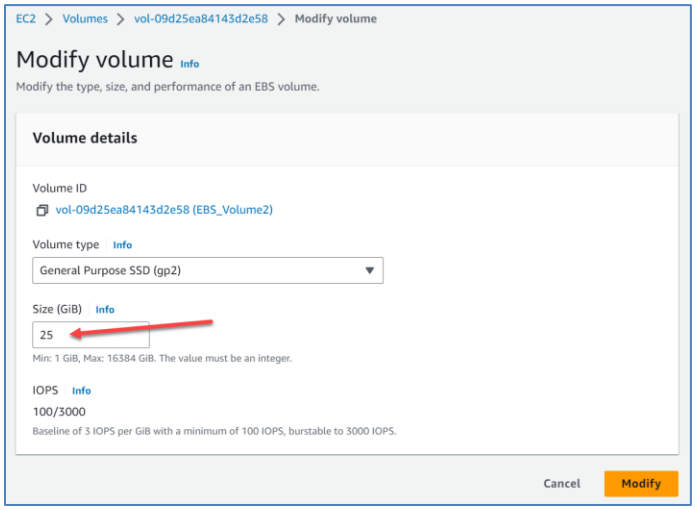




Expand the second volume

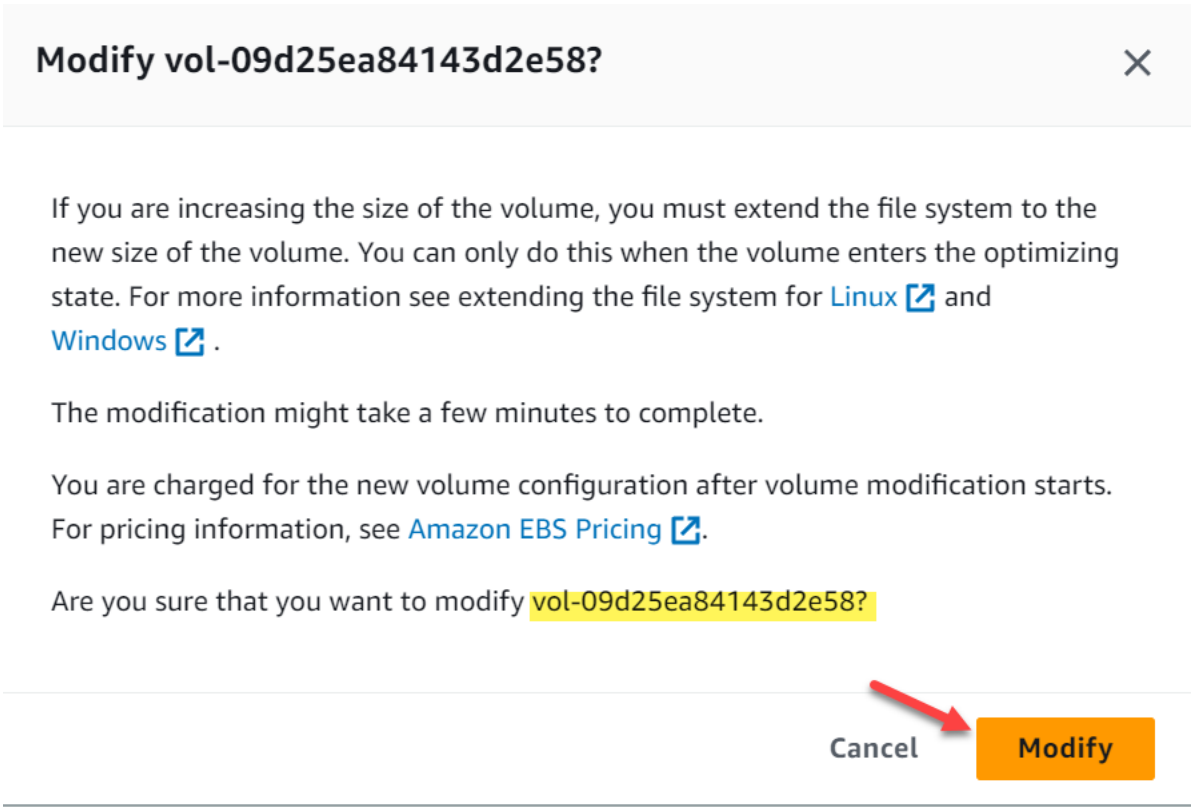


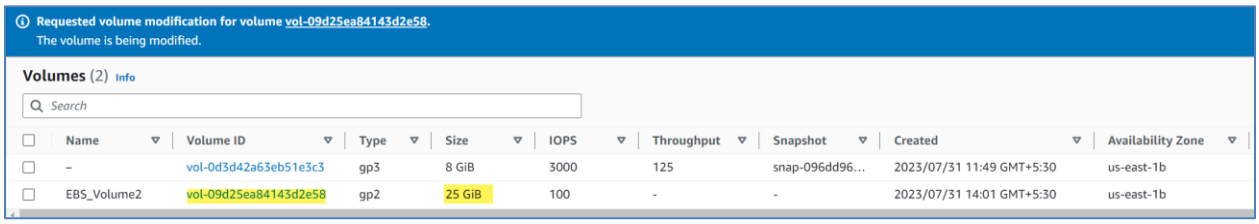
Select the volume Actions → Modify volume →



Update the size details from 20 to 25 which become 25GB

Click Modify





Volume is now become 25GB, expanding the same from Linux OS.

```
[root@ip-172-31-91-22 ec2-user]# 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
xvdg      202:96   0  25G  0 disk /volume2
[root@ip-172-31-91-22 ec2-user]# xfs_growfs /volume2
meta-data=/dev/xvdg              isize=512    agcount=4, agsize=1310720 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=5242880, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 5242880 to 6553600
[root@ip-172-31-91-22 ec2-user]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  2.8M  188M   2% /run
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs           475M     0  475M   0% /tmp
tmpfs            95M     0   95M   0% /run/user/1000
/dev/xvdg        25G  211M   25G   1% /volume2
[root@ip-172-31-91-22 ec2-user]#
```

Login to the EC2 instance then run above commands

`lsblk`

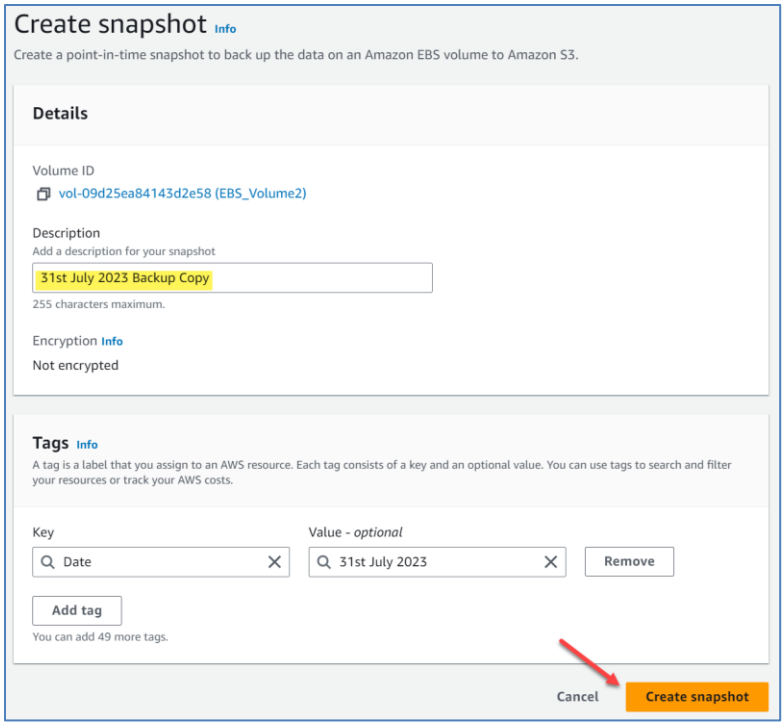
`xfs_growfs /volum2` #since it is a XFS file system, to grow we have to run this command

OR

`resize2fs` #for ext4, ext3 file systems

EBS Volume Backup

Go to Elastic Block Store → Volumes → Select volume → Actions → Create Snapshot



Provide description to refer later point of time.

Add Tags for easy identification

Click "**Create snapshot**"