CLOUD COMPUTING ARCHITECTURE LAB

NAME- SHUBHI DIXIT

BATCH- 05

SAP ID- 500094571

Experiment 5: Attaching EBS to Amazon EC2 Linux and Windows Instance.

Ques 1: Differentiate between EBS, EFS and S3 storage of AWS?

Ans:

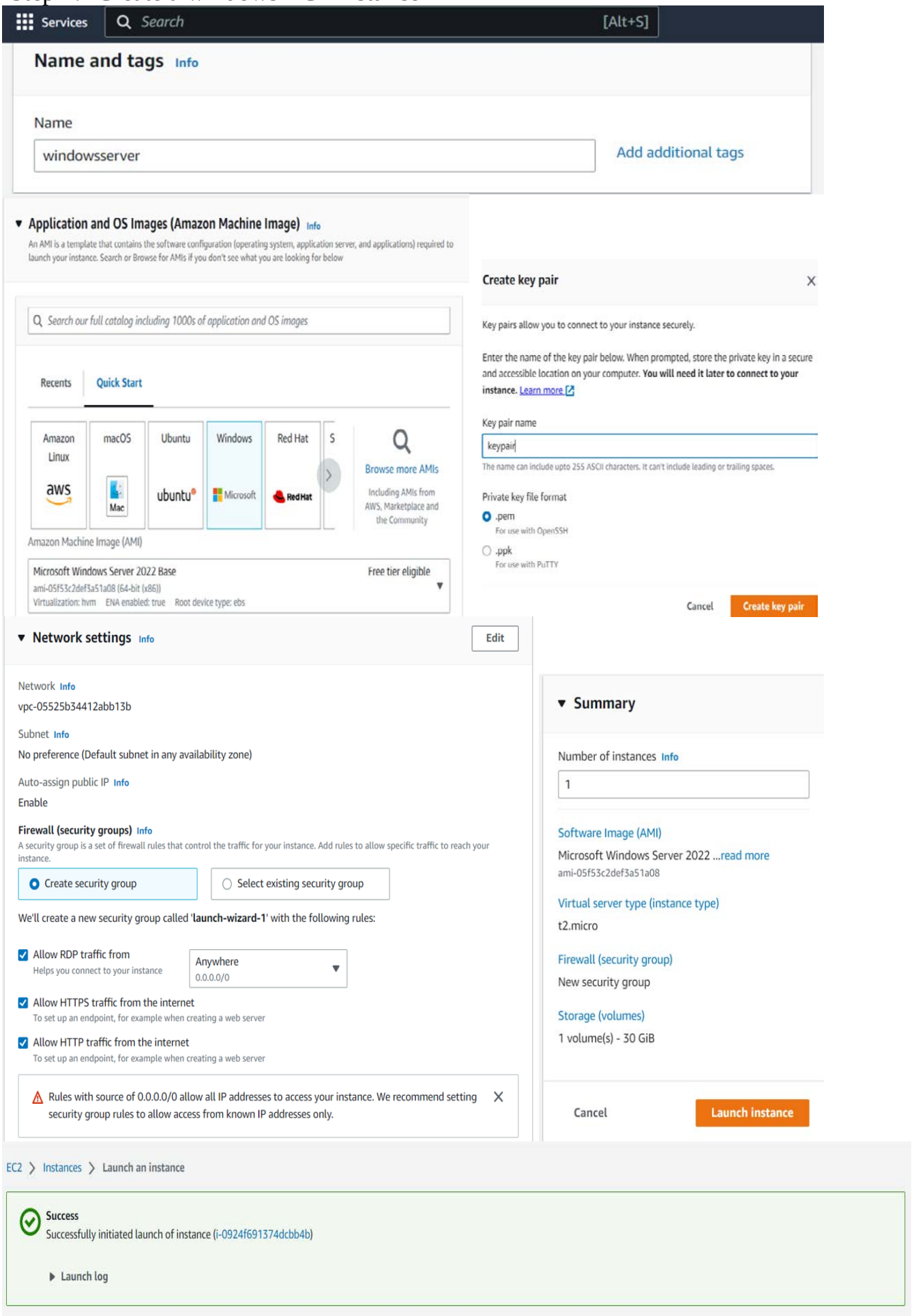
Storage Options	EBS	EFS	S3
Storage Type	Block storage for an EC2 instance.	File system storage for multiple EC2 instances.	Object storage.
Pricing	Pay for provisioning capacity	Pay as you use	Pay as you use
Features	High performance for workloads of a single EC2 instance.	Strong consistency, concurrent accessibility, and file locking features.	Can be accessible to any service or person
Use cases	Boot volumes, transactional and NoSQL databases, data warehousing & ETL.	Home directories, database backups, developer tools, container storage, big data analytics.	Web applications, content management, photos, videos, backups, big data.
Durability	Stored in a single AZ.	Stored across multiple AZs.	Multiple AZs.
Scalability	Scales both horizontally and vertically.	Grow and shrink as files are uploaded/deleted.	Limitless scalability.
Service endpoint	Within a VPC.	Within a VPC.	Within VPC, Without VPC (S3 URL).

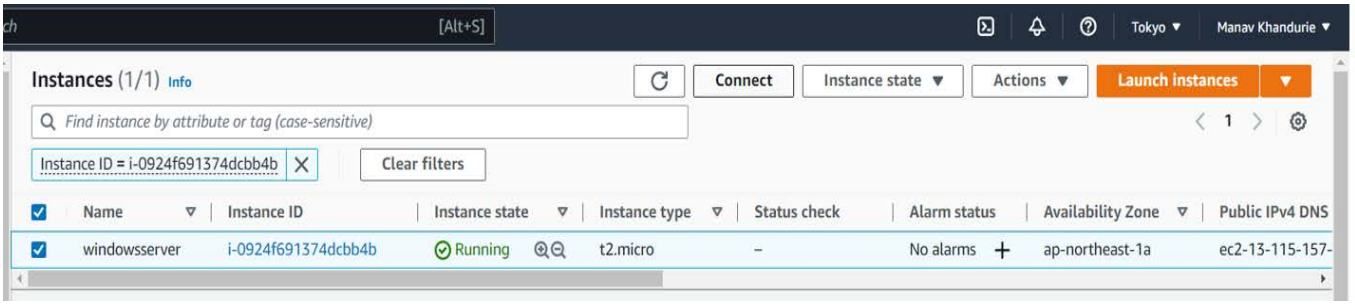
Ques 2: Write the uses cases where EBS (block storage) will be more useful than others.

Ans: Amazon Elastic Block Store (EBS) provides block level storage volumes for use with a single EC2 instance or multiple servers available in the same availability zone.

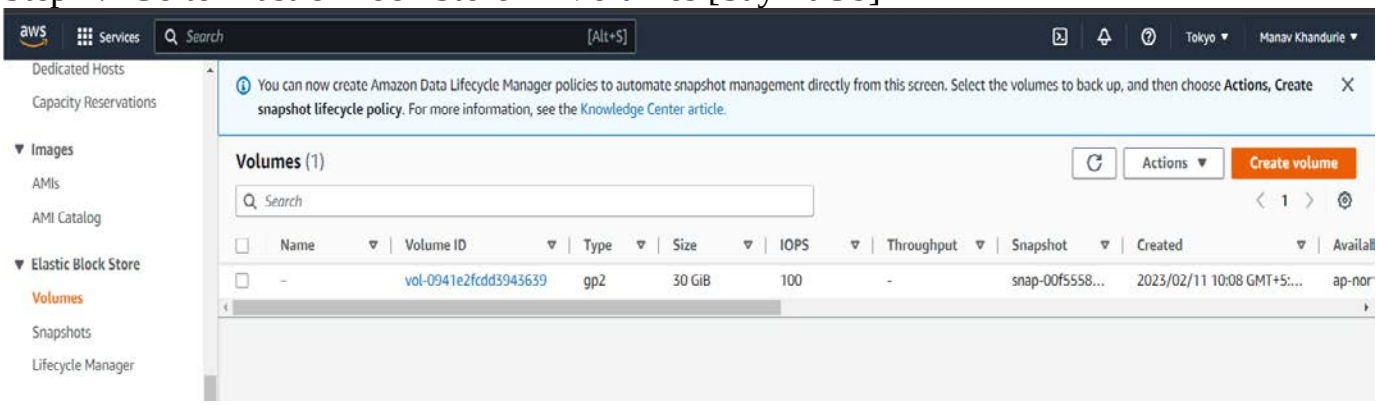
- EBS is best suited for scenarios where frequent IO operations (especially read and write) are required.
- If minimised disk latency is a need, EBS is best suited.
- EBS is also the best option for relational and NoSQL Databases.
- EBS is also comparatively a cost-effective storage.
- Where high availability (99.99%) is the demand, EBS is best suited.
- EBS supports data persistency because its data does not get deleted even if the EC2 instance is terminated.
- With EBS, data is well protected as point-in-time snapshots can be taken.
- It is also used for enterprise applications such as ERP systems, mail servers, middleware etc.

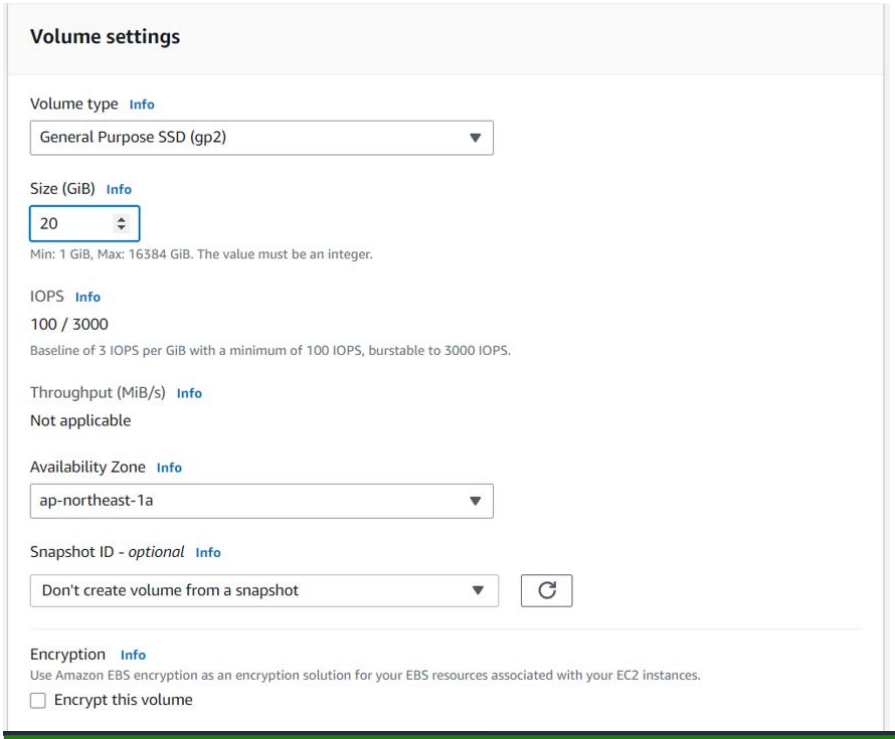
Step 1:- Create a windows EC2 instance

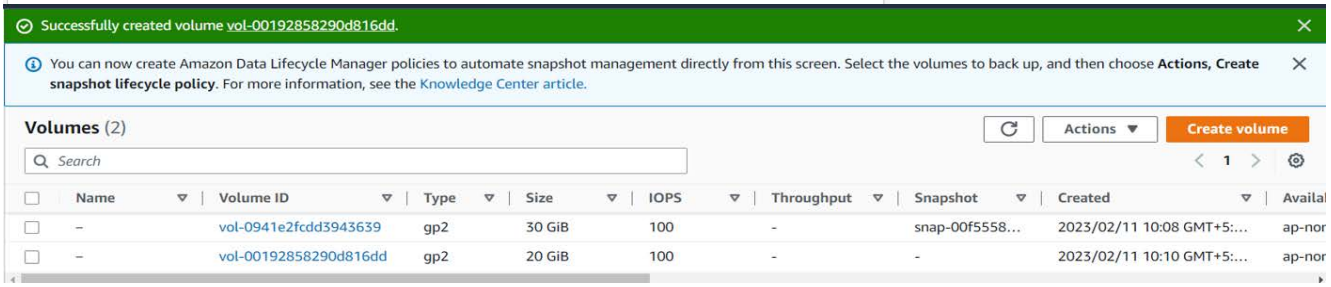




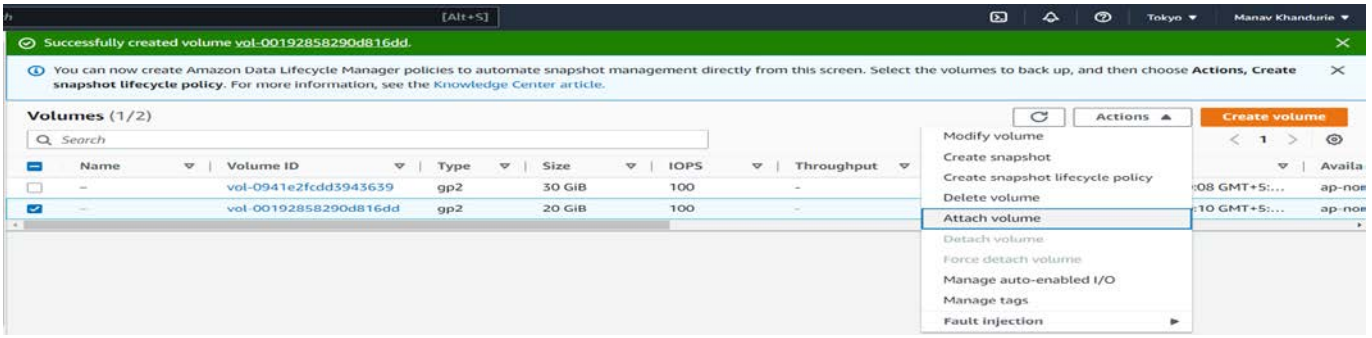
Step 2:- Go to Elastic Block Store -> Volumes [Say 20Gb]

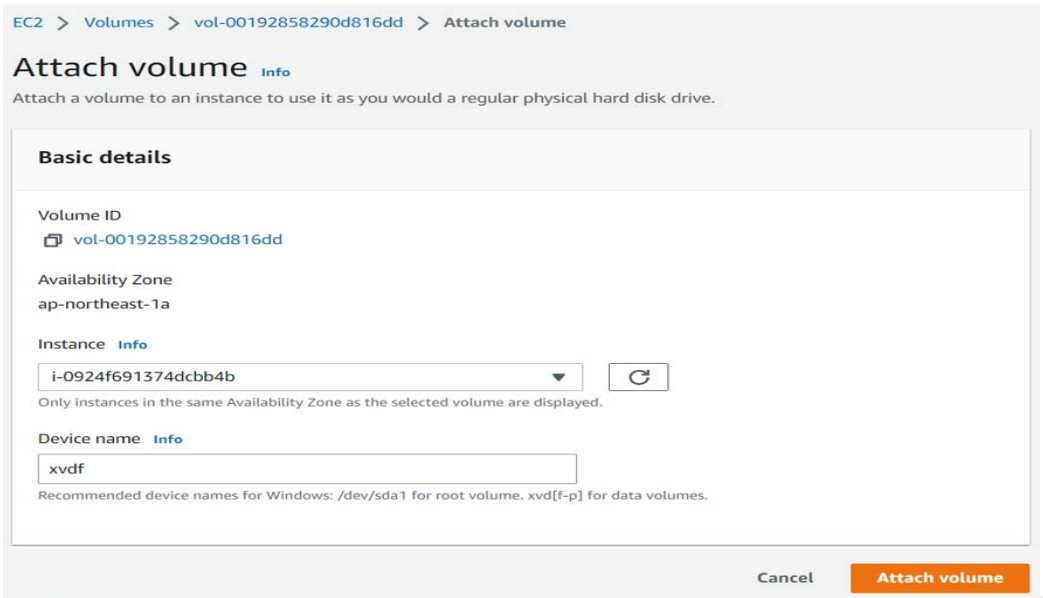


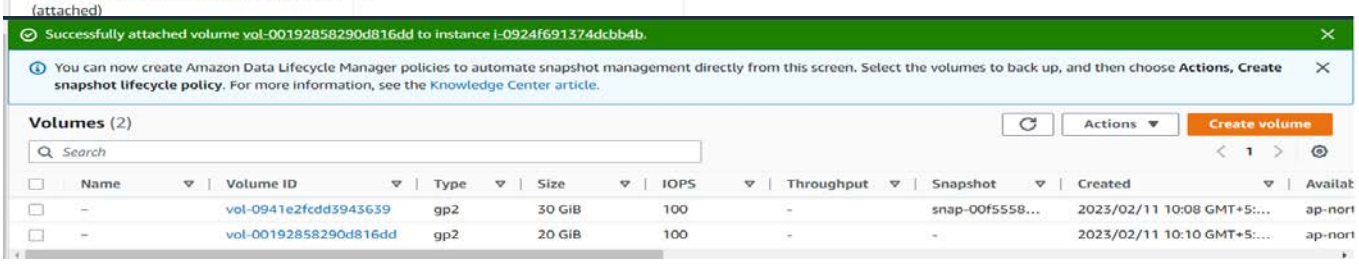




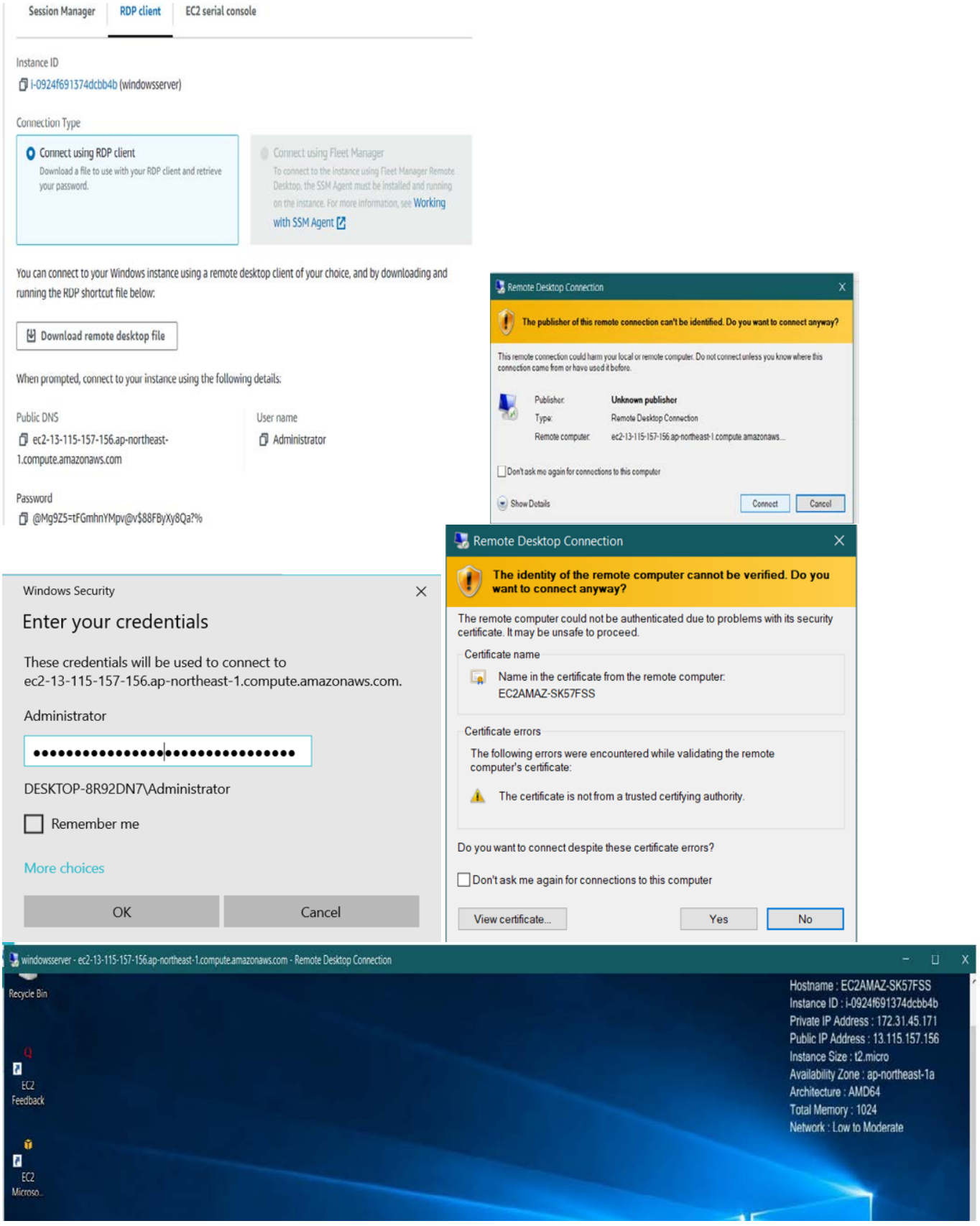
Step 3:- Select the volume created in step 2 and click on actions-> Attach volume and select the instance running



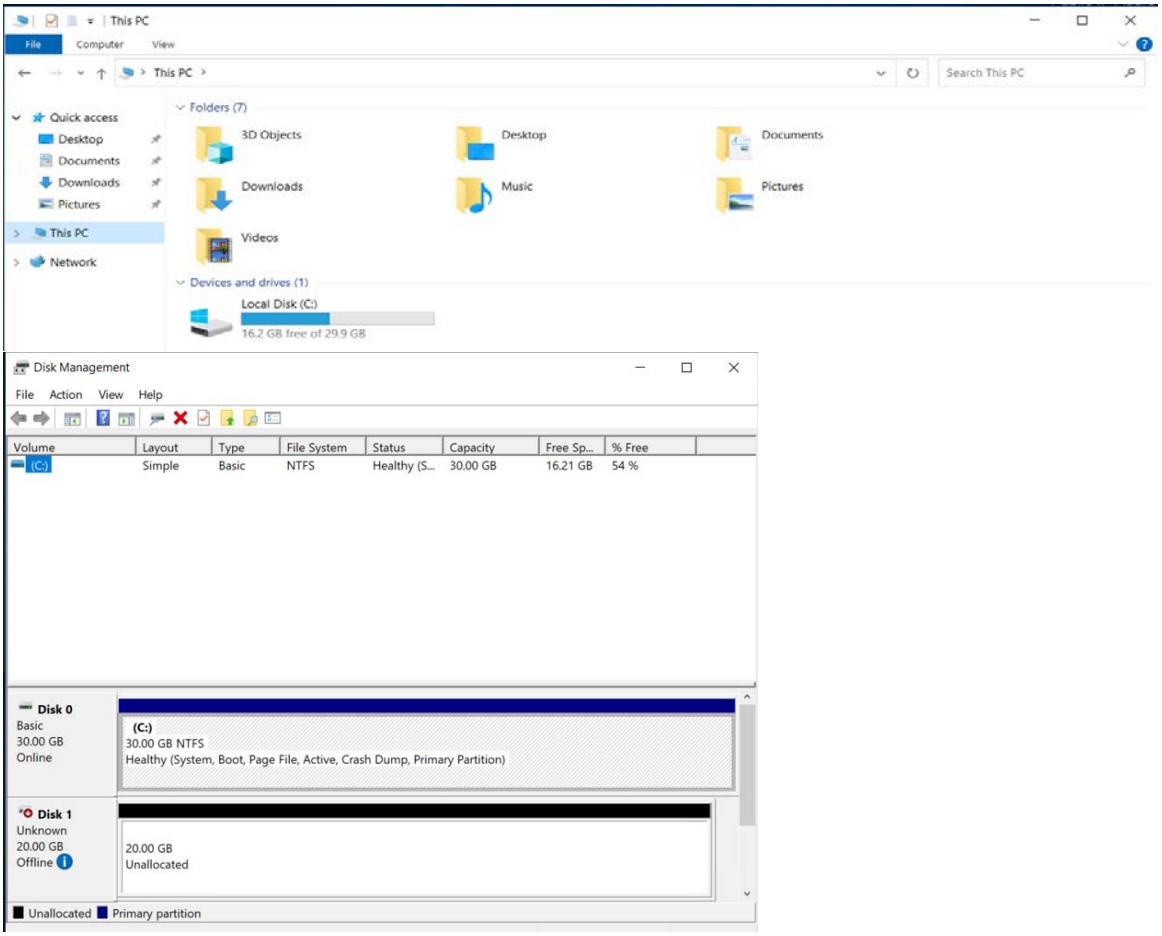




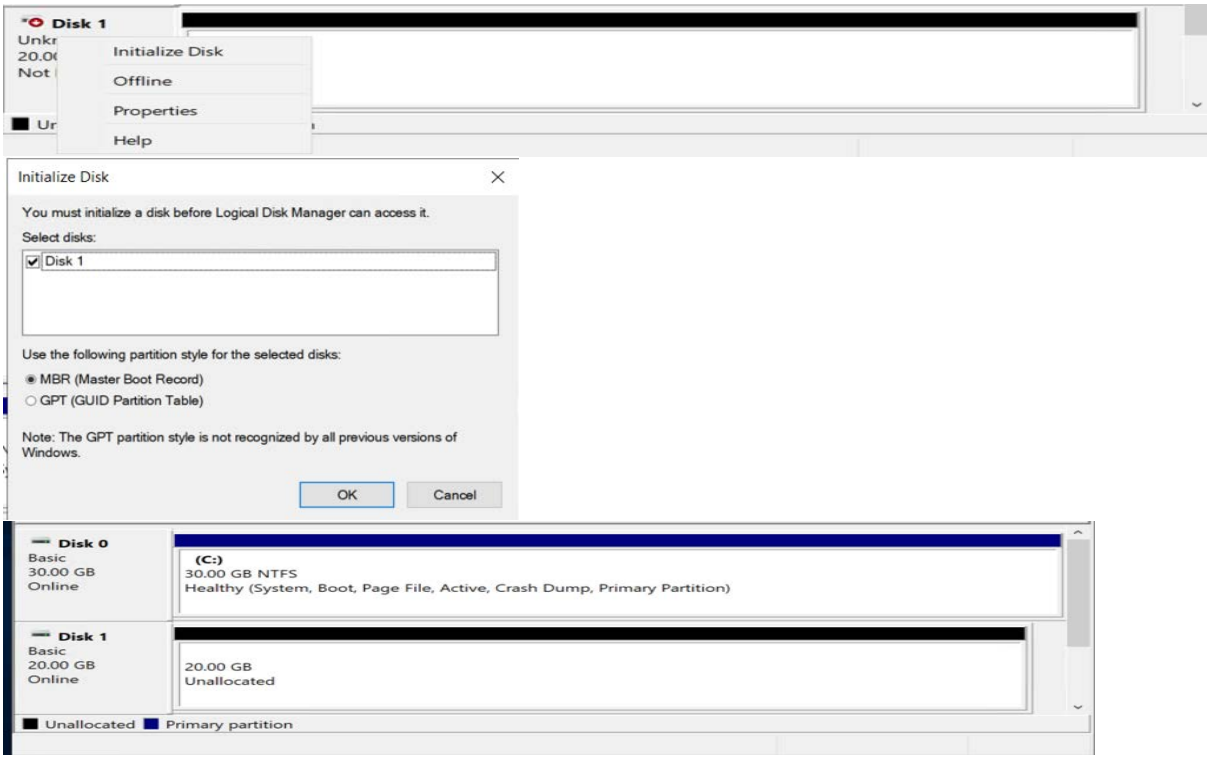
Step 4:- Connect to the instance created in step 1



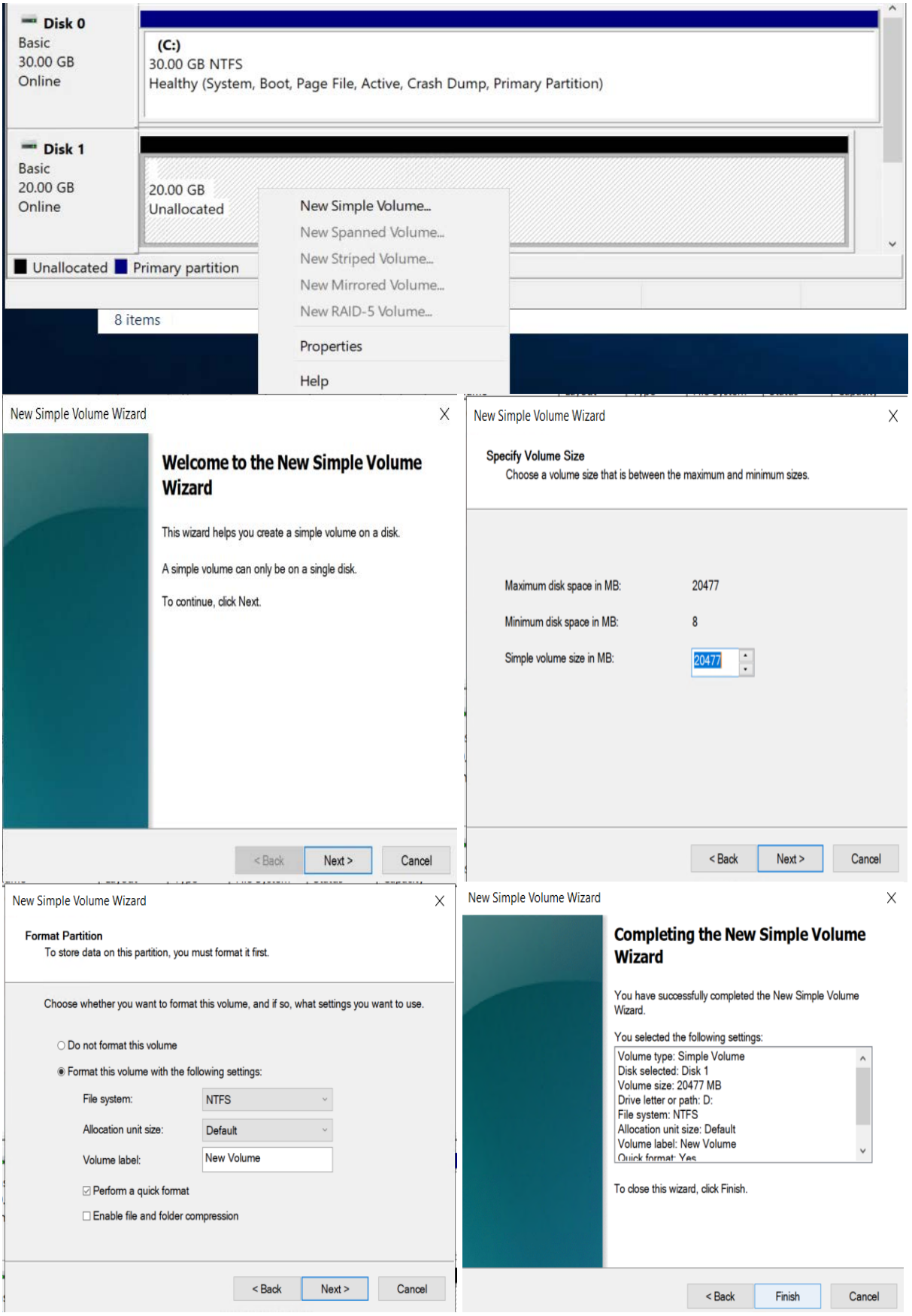
Step 5:- Initially the attached storage is not visible in control panel , to initialize the disk we goto the disk manager (win + x + k)



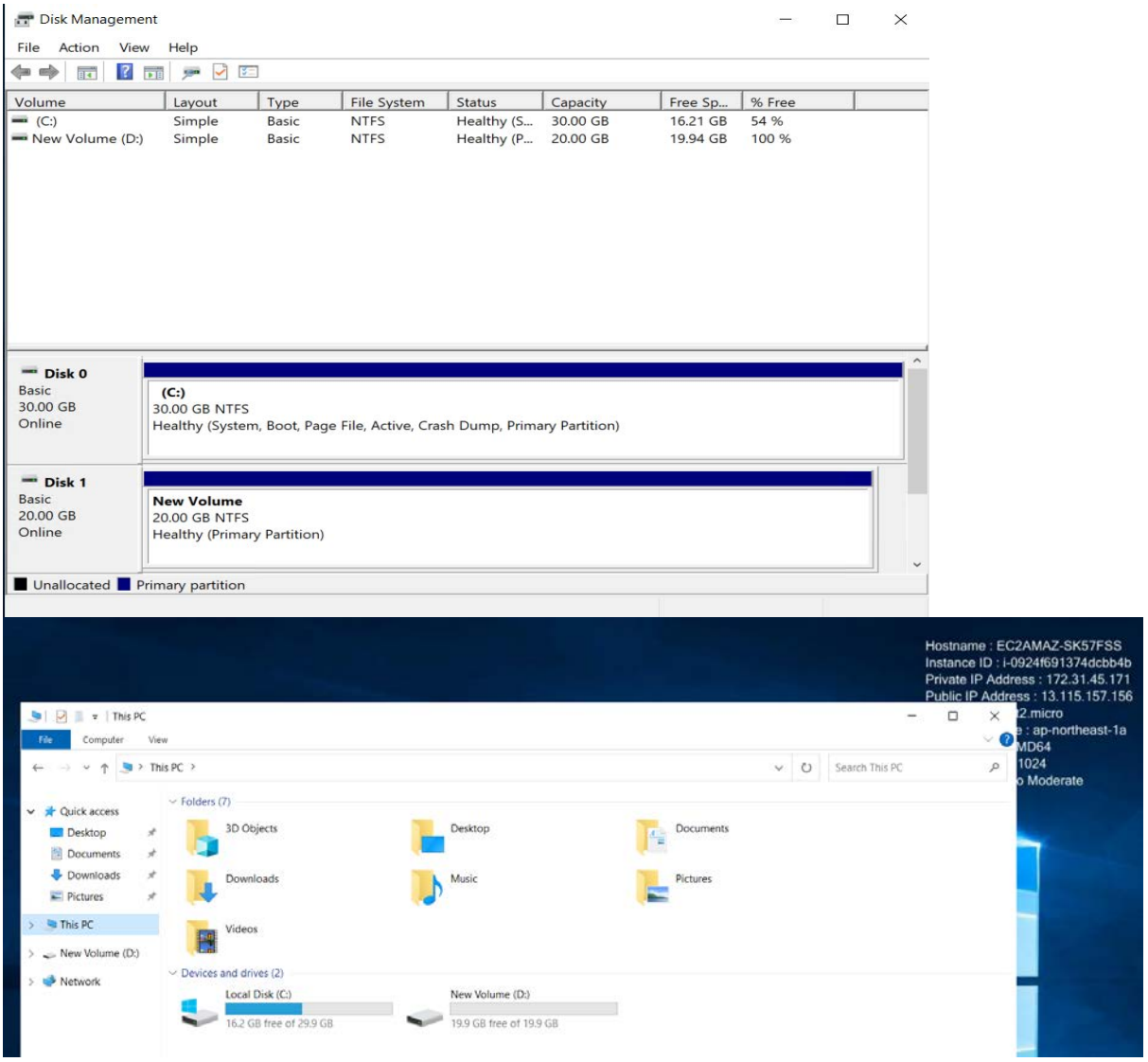
Step 6:- Right click on the unallocated disk and click on initialize the disk



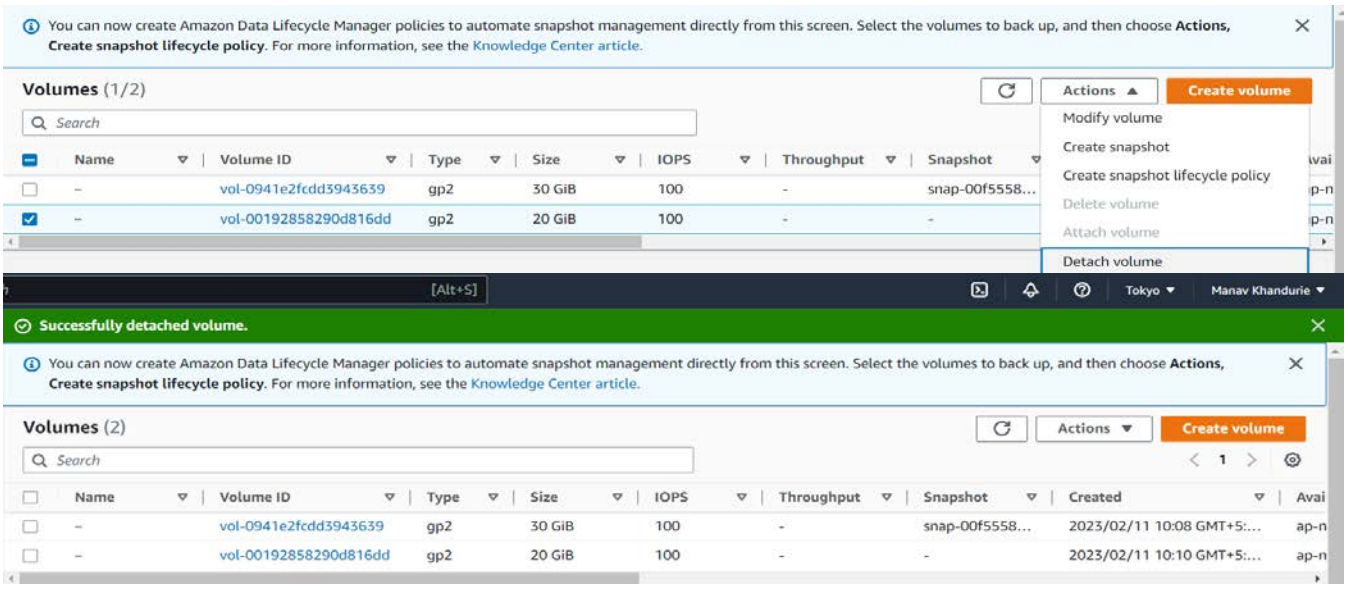
Step 7:- Right click on the unallocated disk and click on “new simple volume” and finish the Volume wizard

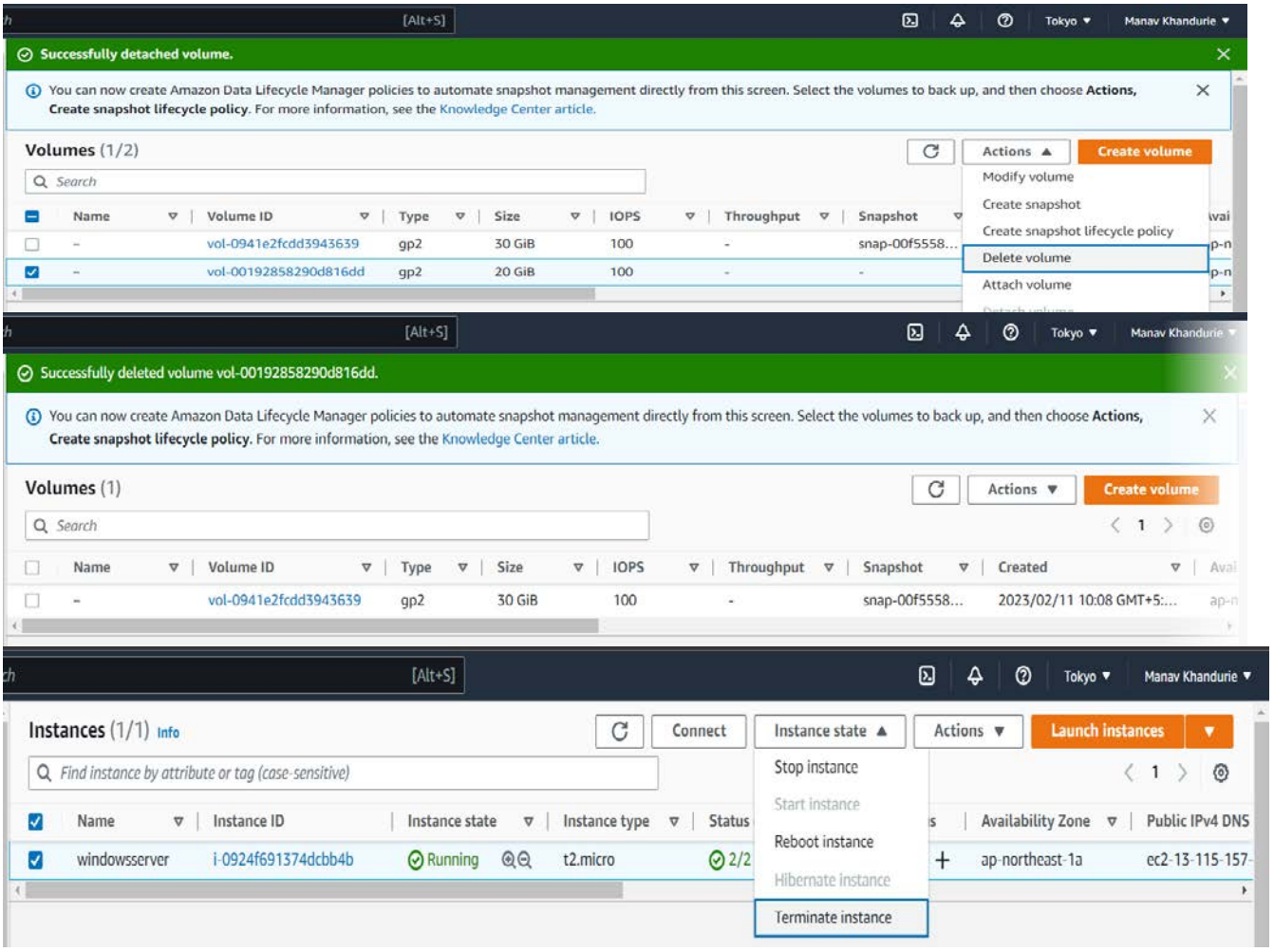


Step 8:- Click on finish and the new volume will get allocated



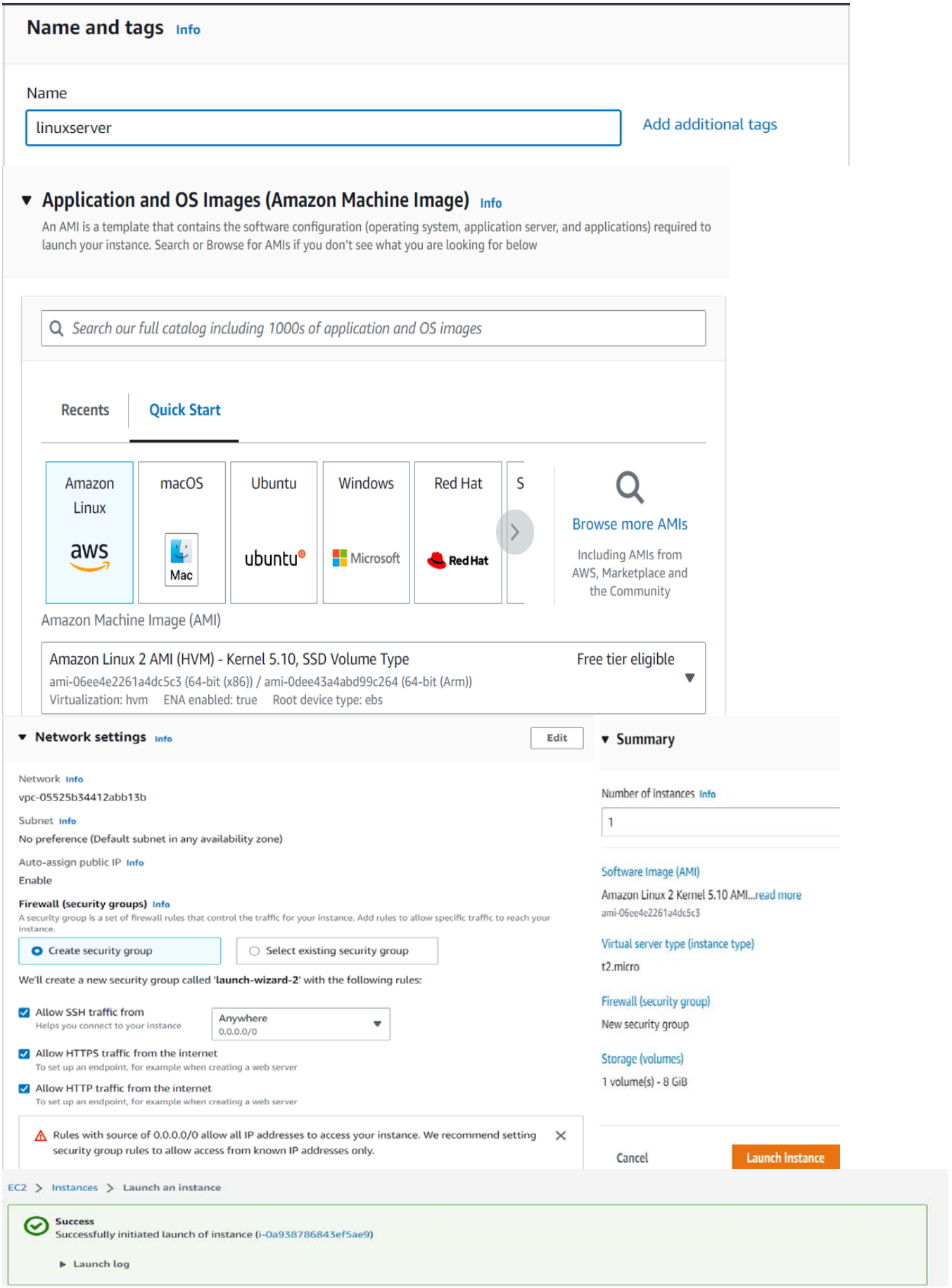
Step 9:- Terminate the instances and detach and delete the EBS volume

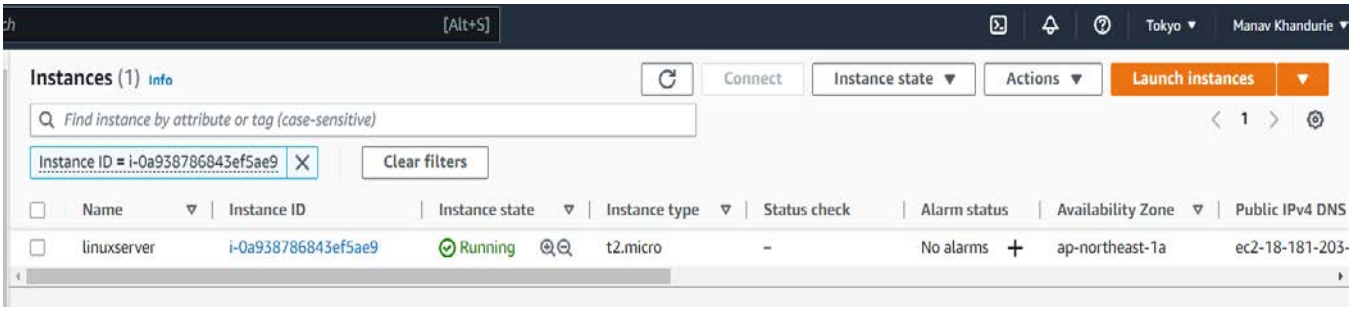



[Alt+S]
Tokyo
Manav Khandurie
Successfully detached volume.
You can now create Amazon Data Lifecycle Manager policies to automate snapshot management directly from this screen. Select the volumes to back up, and then choose Actions, Create snapshot lifecycle policy. For more information, see the Knowledge Center article.
Volumes (1/2)
Search
Actions
Create volume
Modify volume
Create snapshot
Create snapshot lifecycle policy
Delete volume
Attach volume
Name
Volume ID
Type
Size
IOPS
Throughput
Snapshot
vol-0941e2fcdd3943639
gp2
30 GiB
100
snap-00f5558...
vol-00192858290d816dd
20 GiB
Successfully deleted volume vol-00192858290d816dd.
Volumes (1)
Created
2023/02/11 10:08 GMT+5:...
Instances (1/1) Info
Find instance by attribute or tag (case-sensitive)
Connect
Instance state
Launch instances
Stop instance
Start instance
Reboot instance
Hibernate instance
Terminate instance
Instance ID
Instance state
Instance type
Status
Availability Zone
Public IPv4 DNS
windowsserver
i-0924f691374dcbb4b
Running
t2.micro
2/2
ap-northeast-1a
ec2-13-115-157-

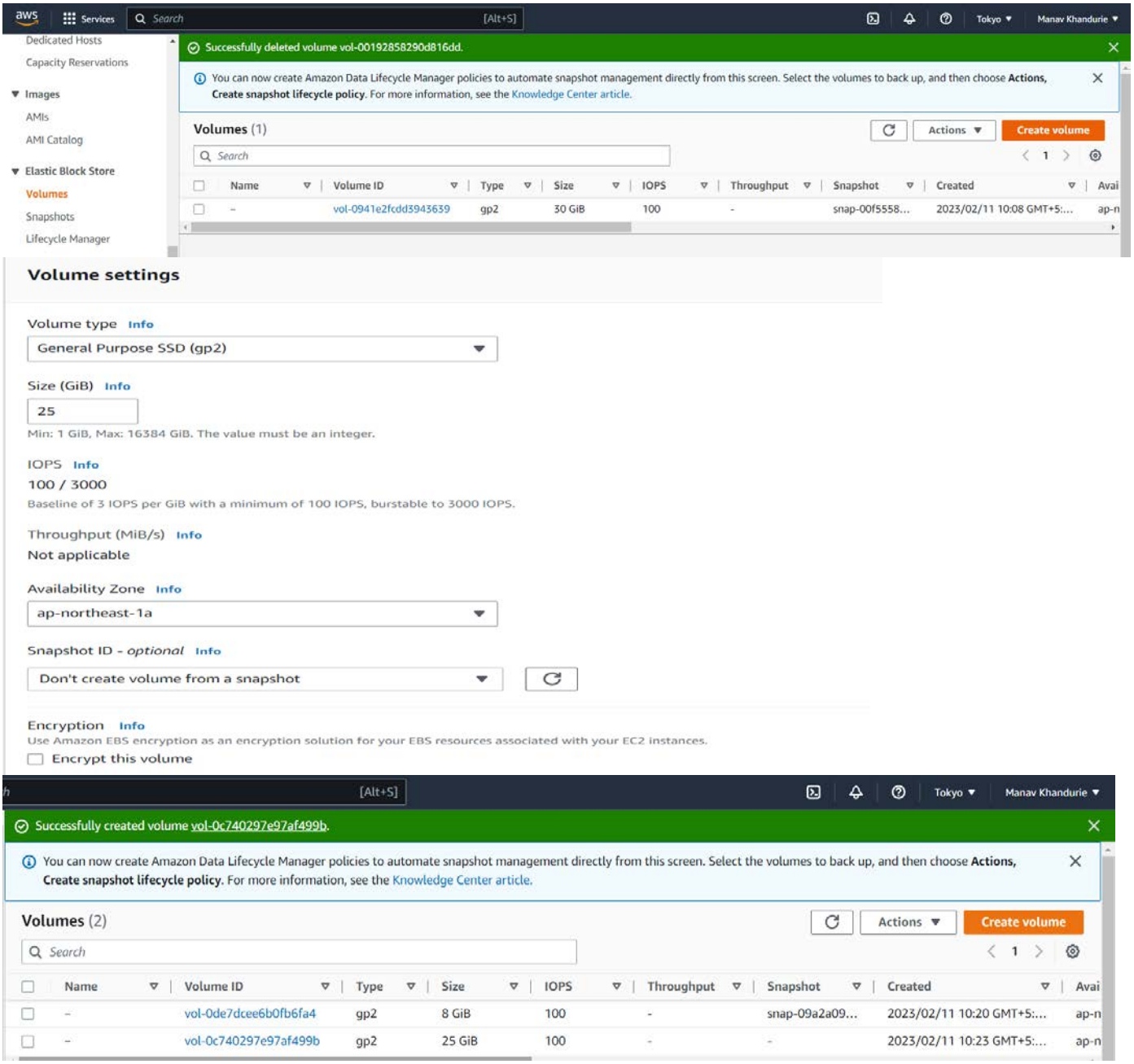
FOR LINUX SERVER-

Step 1:- Create a Linux EC2 instance

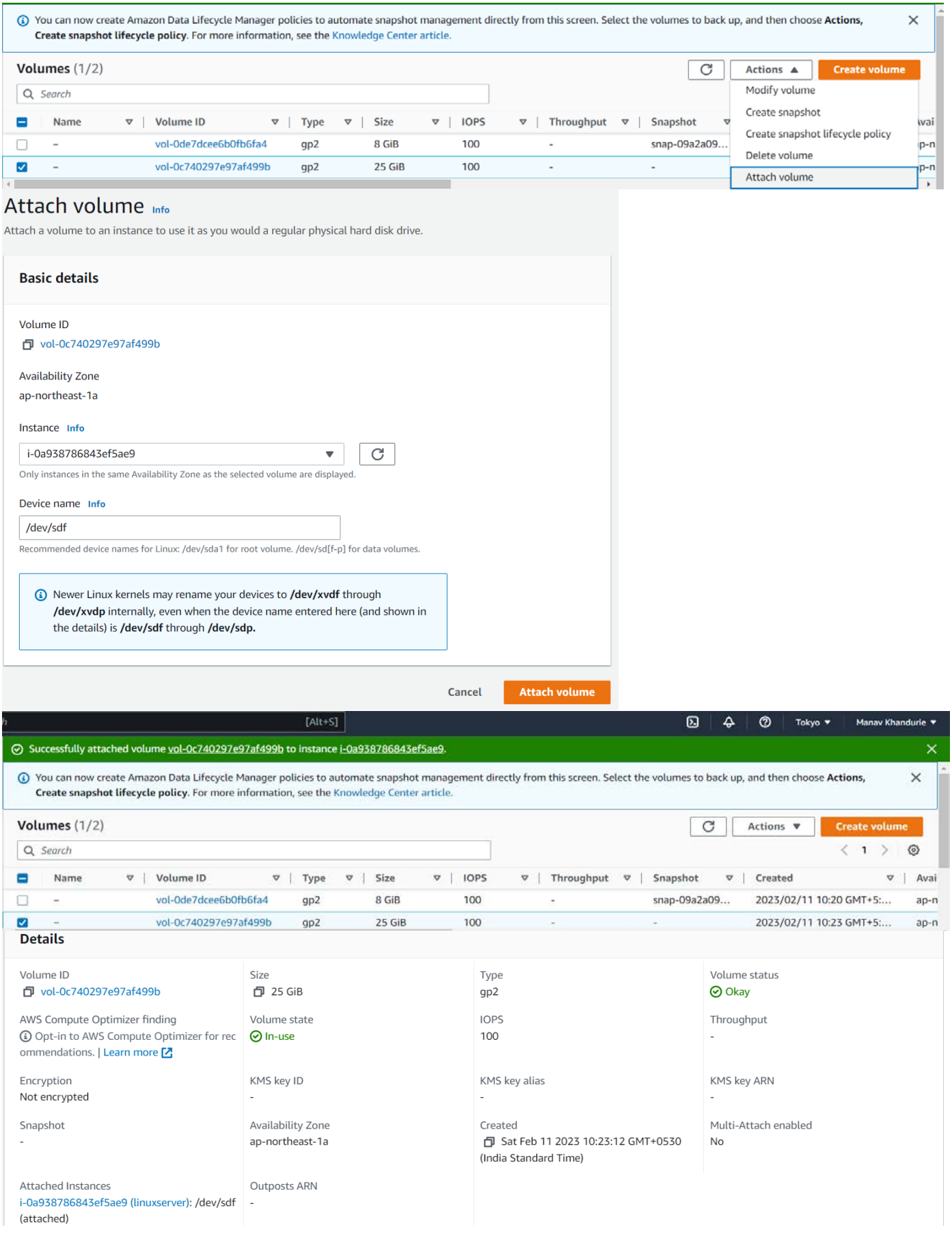




Step 2:- Go to Elastic block Store -> Volumes



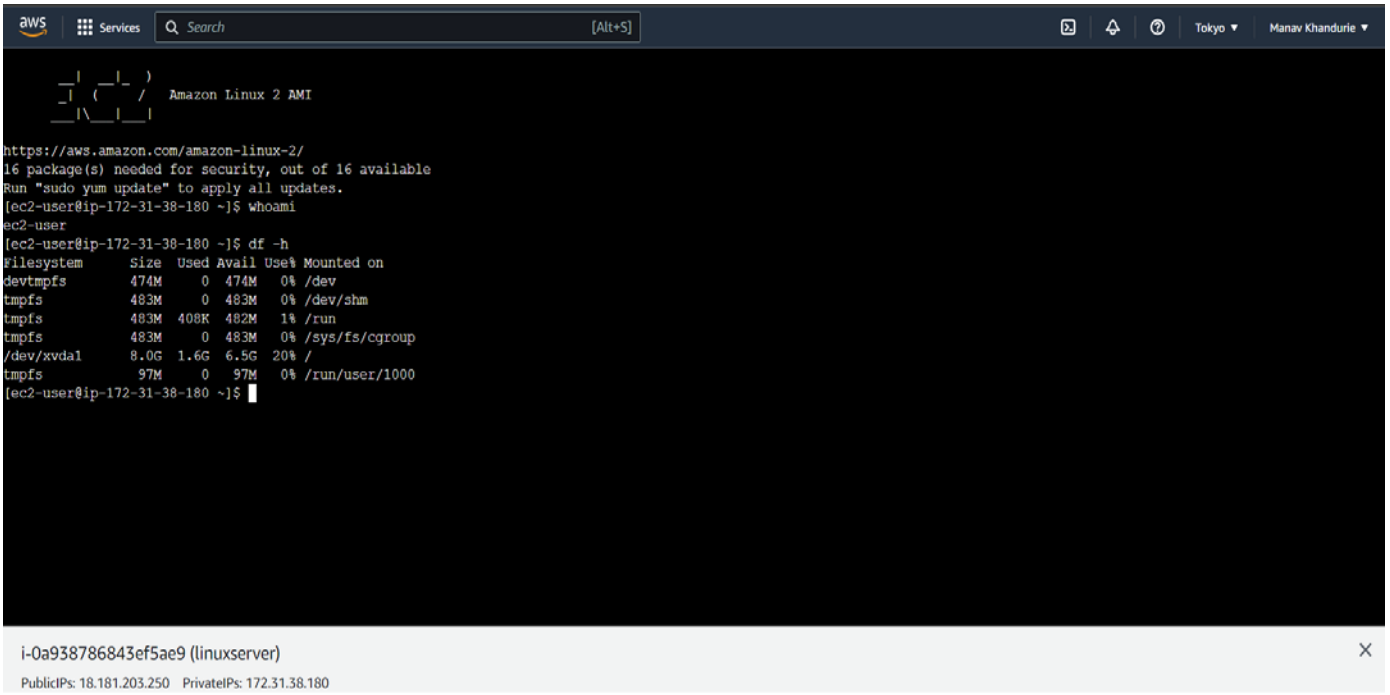
Step 3:- Attach the volume created in step 2 to the instance created in step 1



Step 4:- Connect to the instance created in step 1 and type the command ->

df -h

- ➔ df -> df shows the amount of free space that is left on a file system
- ➔ h -> Its an attribute that displays output in human readable form(in Gbs)



Step 5:- Use the command ->

lsblk

➔ lsblk ->lists information about all available or the specified block devices.

```
https://aws.amazon.com/amazon-linux-2/
16 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-38-180 ~]$ whoami
ec2-user
[ec2-user@ip-172-31-38-180 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           483M     0  483M   0% /dev/shm
tmpfs           483M  408K  482M   1% /run
tmpfs           483M     0  483M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs            97M     0   97M   0% /run/user/1000
[ec2-user@ip-172-31-38-180 ~]$ sudo su
[root@ip-172-31-38-180 ec2-user]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdf    202:80   0  25G  0 disk
```

Step 6:- Use the command sudo su to run commands as root user

sudo su

➔ sudo su -> allows you to run programs as another user, by default the root user

```
[ec2-user@ip-172-31-38-180 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           483M     0  483M   0% /dev/shm
tmpfs           483M  408K  482M   1% /run
tmpfs           483M     0  483M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs            97M     0   97M   0% /run/user/1000
[ec2-user@ip-172-31-38-180 ~]$ sudo su
[root@ip-172-31-38-180 ec2-user]# lsblk
```

Step 7:- Use the command

file -s /dev/xvdf

➔ file-> determines the actual type of a file, no matter what its extension is.

➔ -s -> “s” indicates the file has the setuid bit set.

```
[root@ip-172-31-38-180 ec2-user]# file -s /dev/xvdf
/dev/xvdf: data
[root@ip-172-31-38-180 ec2-user]#
```

Step 8:- Use the command

mkfs -t xfs /dev/xvdf

➔ Mkfs -> makes a new file system on a specified device [everyting in linux is file]

➔ XFS is a high-performing, journaling Linux file system

➔ “-t” tag formats the display of output on the terminal screen

```
[root@ip-172-31-38-180 ec2-user]# mkfs -t xfs /dev/xvdf
meta-data=/dev/xvdf              isize=512    agcount=4, agsize=1638400 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0 inobtcount=0
data     =                       bsize=4096   blocks=6553600, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=3200, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-172-31-38-180 ec2-user]#
```

i-0a938786843ef5ae9 (linuxserver)

PublicIPs: 18.181.203.250 PrivateIPs: 172.31.38.180

Step 9:- Use the command

cd ~

➔ Cd command changes directory

➔ ~ is used to exit the directory

➔ pwd returns the full path name

```
[root@ip-172-31-38-180 ec2-user]# ls
[root@ip-172-31-38-180 ec2-user]# mkdir test
[root@ip-172-31-38-180 ec2-user]# ls
test
[root@ip-172-31-38-180 ec2-user]# cd test
[root@ip-172-31-38-180 test]# pwd
/home/ec2-user/test
[root@ip-172-31-38-180 test]# cd ~
[root@ip-172-31-38-180 ~]# mkdir /apps/my-data/
```

Step 10:- Use the command

mkdir -p /apps/volume/new-volume

- ➔ mkdir allows the user to create directories
- ➔ -p will create parent directory first, if it doesn't exist

```
[root@ip-172-31-38-180 ~]# mkdir /apps/my-data/
mkdir: cannot create directory '/apps/my-data/': No such file or directory
[root@ip-172-31-38-180 ~]#  mkdir -p /apps/volume/new-volume
[root@ip-172-31-38-180 ~]# ls
```

Step 11:- Use the command and then use the df command

mount /dev/xvdf /apps/volume/new-volume

- ➔ mount command is used to mount the filesystem found on a device to big tree structure(Linux filesystem) rooted at '/'
- ➔ its goes by syntax , mount device dir

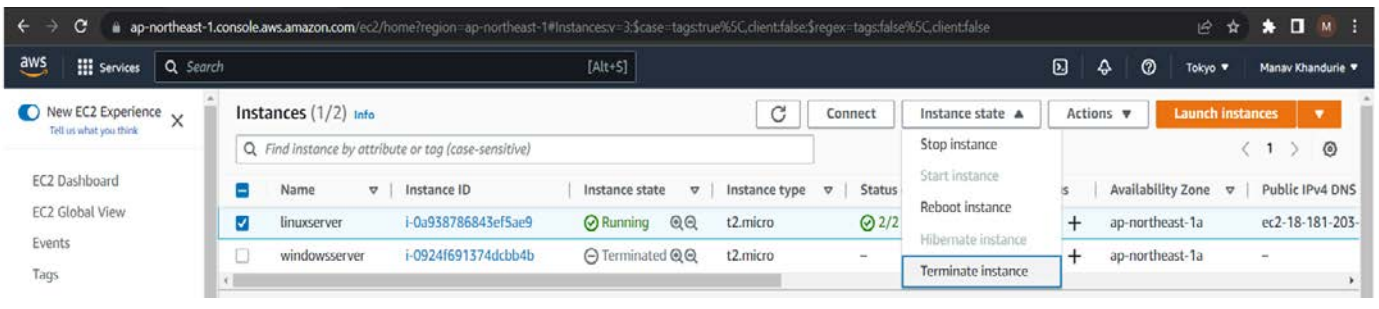
```
[root@ip-172-31-38-180 ~]#  mount /dev/xvdf /apps/volume/new-volume
[root@ip-172-31-38-180 ~]#  df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           483M     0  483M   0% /dev/shm
tmpfs           483M  468K  482M   1% /run
tmpfs           483M     0  483M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs            97M     0   97M   0% /run/user/1000
tmpfs            97M     0   97M   0% /run/user/0
/dev/xvdf        25G  211M   25G   1% /apps/volume/new-volume
[root@ip-172-31-38-180 ~]#
```

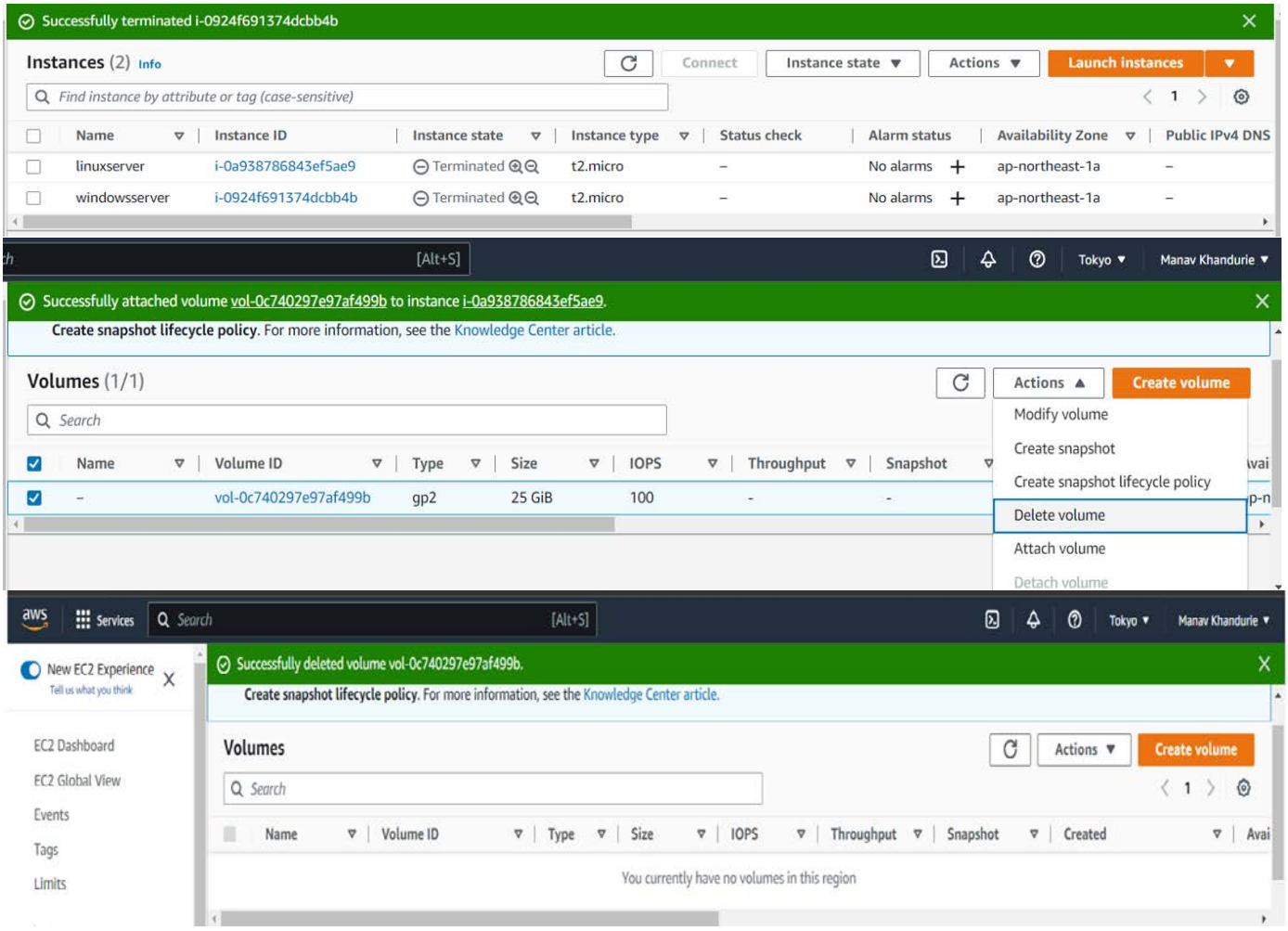
i-0a938786843ef5ae9 (linuxserver)

PublicIPs: 18.181.203.250 PrivateIPs: 172.31.38.180

Step 12:- The new volume that was created in step 3 is attached and online as shown above

Step 13:- Terminate the instances and delete and detach the volumes




Successfully terminated i-0924f691374dcbb4b
Instances (2) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
linuxserver
i-0a938786843ef5ae9
Terminated
t2.micro
–
No alarms
ap-northeast-1a
windowsserver
i-0924f691374dcbb4b
Terminated
t2.micro
–
No alarms
ap-northeast-1a
[Alt+S]
Tokyo
Manav Khandurie
Successfully attached volume vol-0c740297e97af499b to instance i-0a938786843ef5ae9.
Create snapshot lifecycle policy. For more information, see the Knowledge Center article.
Volumes (1/1)
Actions
Create volume
Search
Modify volume
Create snapshot
Create snapshot lifecycle policy
Delete volume
Attach volume
Detach volume
Name
Volume ID
Type
Size
IOPS
Throughput
Snapshot
vol-0c740297e97af499b
gp2
25 GiB
100
aws
Services
Search
[Alt+S]
Tokyo
Manav Khandurie
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Successfully deleted volume vol-0c740297e97af499b.
Create snapshot lifecycle policy. For more information, see the Knowledge Center article.
Volumes
Actions
Create volume
Search
Name
Volume ID
Type
Size
IOPS
Throughput
Snapshot
Created
You currently have no volumes in this region