

Course: DevOps
Module: AWS & Services
Topic: EBS & EFS
Trainer Name: Mr. Madhukar sir
Name: Billipati Sai Teja
Batch no: 115
Assignment no: 08
Date of submission: 11th – Feb – 2024
Mail-ID: (BILLIPATISAITEJA@GMAIL.COM)

1.To create one EBS (Elastic Block Store) volume and attach it to 3 instances

Sign in to the AWS Management Console.

click on "Security Groups".

Click the "Create Security Group" button.

In the "Create Security Group" wizard, configure the security group settings:

Enter a name for the security group in the "Security group name" field.

Optionally, enter a description for the security group in the "Description" field.

Select the VPC (Virtual Private Cloud) where you want to create the security group (in this it was in default mood).

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar navigation includes Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups), CloudShell, and Feedback. The main content area displays the details of a security group named "sg-061695668af16da0 - efs-ebs-1". The "Details" section shows the security group name, ID (sg-061695668af16da0), description (ssh), owner (358105589371), inbound rules count (1 Permission entry), and outbound rules count (1 Permission entry). Below the details are tabs for Inbound rules, Outbound rules, and Tags. A context menu titled "Actions" is open, listing: Edit inbound rules, Edit outbound rules, Manage tags, Copy to new security group, Delete security groups, and a copy link option (Obj90d721d6a47fce1). The bottom of the screen includes the standard AWS footer with links to CloudShell, Feedback, and various legal and support information.

In the "Inbound Rules" section, define the inbound traffic rules for the security group. You can add rules based on your application's requirements, such as allowing SSH (port 22) or HTTP (port 80) traffic. Click the "Add Rule" button to add rules.

The screenshot shows the "Edit inbound rules" page for the security group "sg-061695668af16da0 - efs-ebs-1". The page title is "ModifyInboundSecurityGroupRules" and the URL is "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-061695668af16da0". The left sidebar and top navigation bar are identical to the previous screenshot. The main content area is titled "Edit inbound rules" and contains the following table:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0c700ce2d6c58c655	SSH	TCP	22	Cu... ▾	0.0.0.0/0 X

Below the table is a "Delete" button. At the bottom of the page is an "Add rule" button. The footer includes the standard AWS footer with links to CloudShell, Feedback, and various legal and support information.

Click the "Create" button to create the security group.

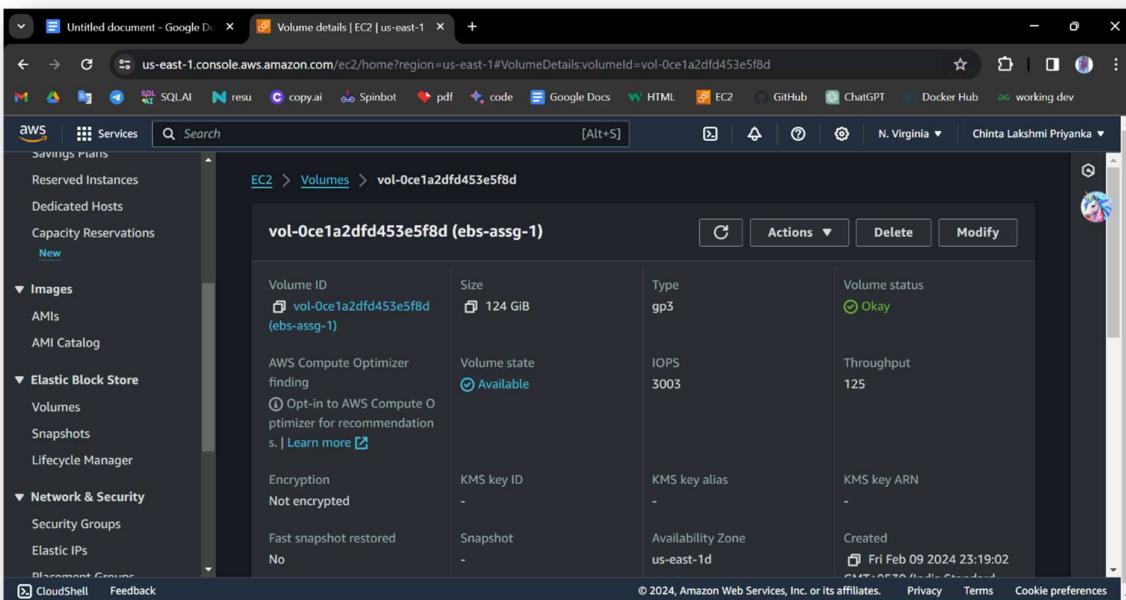
Now create ebs

1. Open the Amazon EC2 console at

[https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)

2. In the navigation pane, click on "Volumes".

3. Click the "Create Volume" button.



4. In the "Create Volume" wizard, configure the volume settings:

- Select the desired Volume Type, Size, Availability Zone, and other settings according to your requirements.

5. Click the "Create" button to create the EBS volume.

6. Once the volume is created, note down the Volume ID.

The screenshot shows the AWS EC2 Instances page. The navigation pane on the left is expanded, showing 'Instances' selected under 'EC2'. The main content area displays a table of 21 instances. The columns include Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Five instances are listed as 'Running': 'ebs-3' (i-0874416e5b4f4fd9), 'vpc-ass-1' (i-00cb7e4fe4c3743cd), 'sample-vpc-1' (i-0e7407e32f2e294ee), 'ebs -1' (i-0484a249943802762), and 'ebs -2' (i-01fefc37f4a679635). The 'Actions' button is highlighted in orange at the top right of the table.

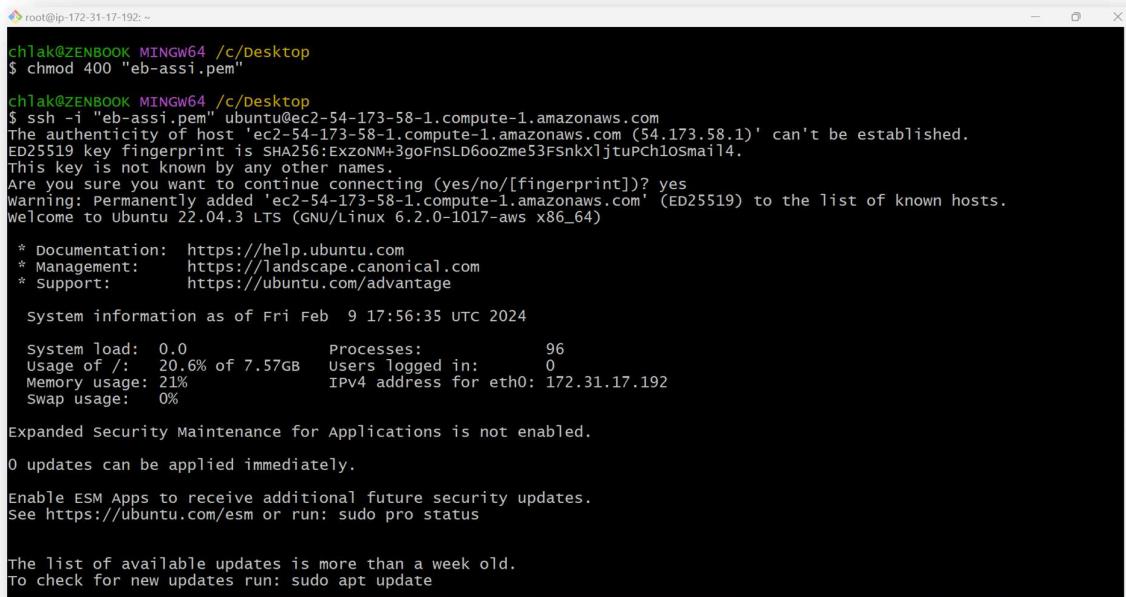
7. Now go to the EC2 instance and create 3 instances
8. In the navigation pane, click on "Instances" and select the instances you want to attach the EBS volume to.
9. Click on the "Actions" button and select "Attach Volume".
10. Attach the first instance to ebs volume.

The screenshot shows the 'Attach volume' page for instance 'i-0137d93b1434a2d20'. The URL in the browser is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AttachVolume;volumeId=vol-0ce1a2dfd453e5f8d'. The page has a breadcrumb navigation: EC2 > Volumes > vol-0ce1a2dfd453e5f8d > Attach volume. The main form is titled 'Attach volume' and contains a 'Basic details' section. It shows the Volume ID as 'vol-0ce1a2dfd453e5f8d (ebs-assg-1)', the Availability Zone as 'us-east-1', and the Instance dropdown set to 'i-0137d93b1434a2d20'. The 'Device name' field is also visible. A note at the bottom says 'Only instances in the same Availability Zone as the selected volume are displayed.'

Now, Connect the instance

To create a file system on an EBS (Elastic Block Store) volume after connecting it to an instance, you can follow these steps:

1. Connect to the instance that has the EBS volume attached using SSH or any other remote access method.



```
root@ip-172-31-17-192: ~
ch1ak@ZENBOOK MINGW64 ~/c/Desktop
$ chmod 400 "eb-assi.pem"

ch1ak@ZENBOOK MINGW64 ~/c/Desktop
$ ssh -i "eb-assi.pem" ubuntu@ec2-54-173-58-1.compute-1.amazonaws.com
The authenticity of host 'ec2-54-173-58-1.compute-1.amazonaws.com (54.173.58.1)' can't be established.
ED25519 key fingerprint is SHA256:ExzoNM+3goFnSLD6ooZme53FsnkXljtUPchiosmail14.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-173-58-1.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Fri Feb  9 17:56:35 UTC 2024

System load:  0.0          Processes:           96
Usage of /:   20.6% of 7.57GB  Users logged in:      0
Memory usage: 21%
Swap usage:   0%
                                         IPv4 address for eth0: 172.31.17.192

Expanded security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

2. Once connected to the instance, you need to identify the device name of the EBS volume that you want to create a file system on. **lsblk**

This command will display a list of available block devices, including the attached EBS volume. It will show the device name (e.g., /dev/xvdf, /dev/sdf) of the volume you want to use for creating the file system.

To create file system:

```
mkfs -s /dev/xvdf
```

To create directory:

```
mkdir -p aws/v1&v2
```

4. After the file system is created, you may need to mount it to a directory so that you can access and use it.

```
mount /dev/xvdf aws/v1
```

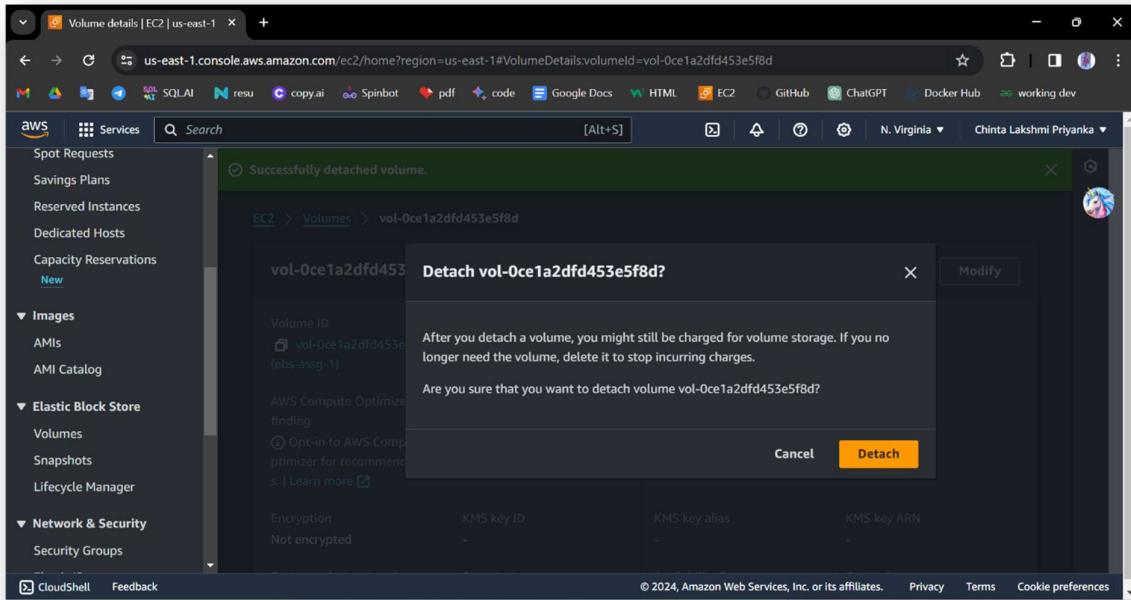
5. You can verify the mounted file system by using the **df -h** command, which lists the file systems and their mount points.

The screenshot shows a terminal window with the following session:

```
[root@ip-172-31-17-192: ~]
[root@ip-172-31-17-192: ~]# mkfs -t xfs /dev/xvdf
meta-data=/dev/xvdf isize=512 agcount=4, agsize=8126464 blks
          = sectsz=512 attr=2, projid32bit=1
          = crc=1 finobt=1, sparse=1, rmapbt=0
data      = reflink=1 bigtime=0 inobtcount=0
          = bsize=4096 blocks=32505856, imaxpct=25
          = sunit=0 swidth=0 blks
naming   =version 2 bsize=4096 ascii-ci=0, ftype=1
log       =internal log bsize=4096 blocks=15872, version=2
          = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
[root@ip-172-31-17-192: ~]# mkdir -p aws/v1&v2
[1] 1994
v2: command not found
[1]+  done                      mkdir -p aws/v1
root@ip-172-31-17-192: ~]# cd aws/
root@ip-172-31-17-192:~/aws# ls
v1
root@ip-172-31-17-192:~/aws# cd ..
root@ip-172-31-17-192: ~]# mount /dev/xvdf aws/v1
root@ip-172-31-17-192: ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       7.6G  1.6G  6.0G  21% /
tmpfs           475M    0  475M   0% /dev/shm
tmpfs           190M  852K  190M   1% /run
tmpfs            5.0M    0  5.0M   0% /run/lock
/dev/xvda1s     105M  6.1M  99M   6% /boot/efi
tmpfs           95M  4.0K  95M   1% /run/user/1000
/dev/xvdf       124G  918M 124G   1% /root/aws/v1
root@ip-172-31-17-192: ~]
```

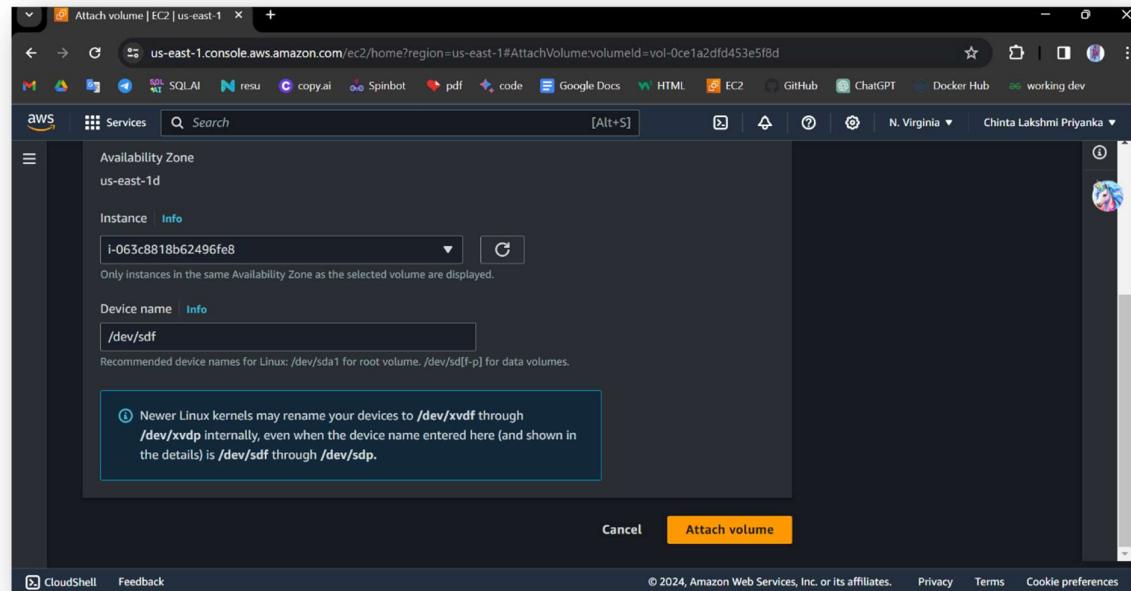
df -h

Now, you should have successfully created a file system on the EBS volume and mounted it to the desired directory on the instance.



Now detach the volume and attach it to another ebs-2 instance

Attach to the second volume.



Now check whether the file system is created or not.

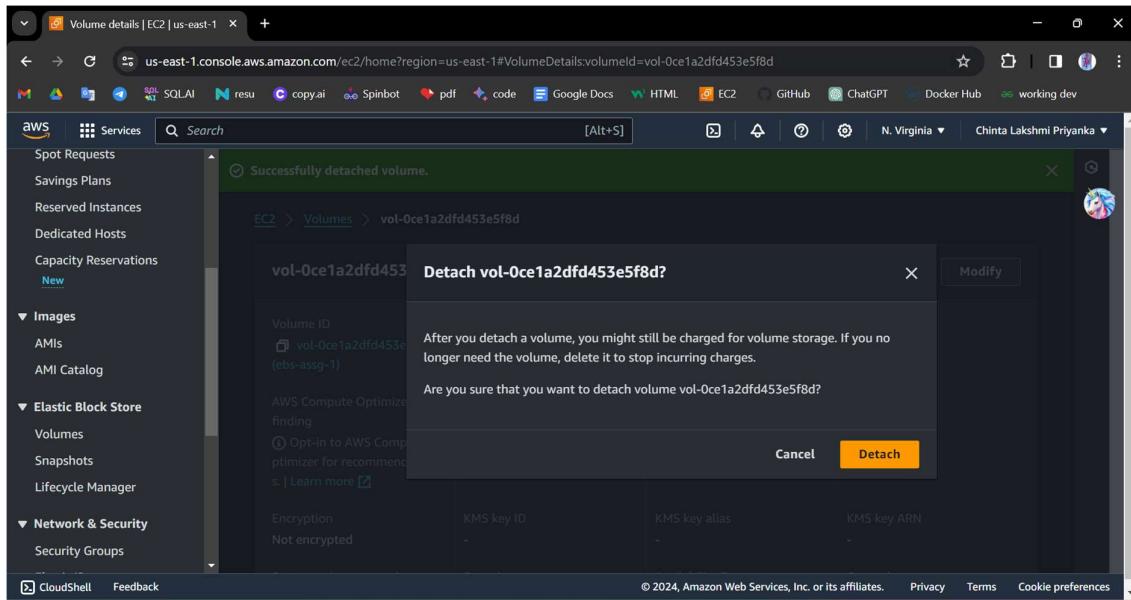
```
root@ip-172-31-23-68:~  
Are you sure you want to continue connecting (yes/no/[fingerprint])?  
? yes  
Warning: Permanently added 'ec2-54-174-82-129.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Sat Feb 10 04:28:41 UTC 2024  
System load: 0.00830078125 Processes: 100  
Usage of /: 20.6% of 7.57GB Users logged in: 0  
Memory usage: 20% IPv4 address for eth0: 172.31.23.68  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-23-68:~$ sudo -i  
root@ip-172-31-23-68:~#
```

And check whether the file system is created or not.

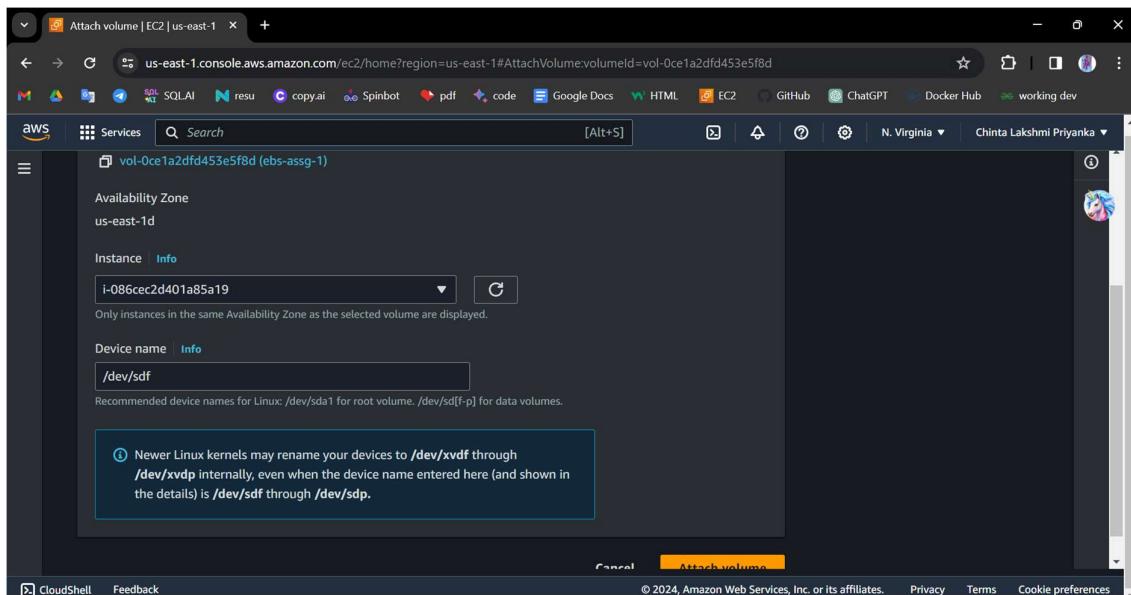
```
root@ip-172-31-23-68:~  
To check for new updates run: sudo apt update  
Last login: Sat Feb 10 06:11:01 2024 from 103.88.236.42  
ubuntu@ip-172-31-23-68:~$ sudo -i  
root@ip-172-31-23-68:~# df -h  
Filesystem      size  Used Avail Use% Mounted on  
/dev/root       7.6G  1.8G  5.8G  24% /  
tmpfs          475M    0   475M  0% /dev/shm  
tmpfs          190M  848K  190M  1% /run  
tmpfs          5.0M    0   5.0M  0% /run/lock  
/dev/xvda15     105M  6.1M   99M  6% /boot/efi  
tmpfs          95M   4.0K   95M  1% /run/user/1000  
root@ip-172-31-23-68:~# lsblk  
Command 'lsblk' not found, did you mean:  
  command 'lsbin' from deb util-linux (2.37.2-4ubuntu3)  
Try: apt install <deb name>  
root@ip-172-31-23-68:~# lsblk  
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS  
loop0       7:0    0   24.9M  1 loop /snap/amazon-ssm-agent/7628  
loop1       7:1    0   55.7M  1 loop /snap/core18/2812  
loop2       7:2    0   63.5M  1 loop /snap/core20/2015  
loop3       7:3    0   63.9M  1 loop /snap/core20/2105  
loop4       7:4    0  111.9M  1 loop /snap/lxd/24322  
loop5       7:5    0   87M   1 loop /snap/lxd/27037  
loop6       7:6    0   40.9M  1 loop /snap/snappyd/20290  
loop7       7:7    0   40.4M  1 loop /snap/snappyd/20671  
xvda      202:0    0   8G   0 disk  
└─xvda1    202:1    0   7.9G  0 part /  
└─xvda14   202:14   0   4M   0 part  
└─xvda15   202:15   0  106M  0 part /boot/efi  
xvdf      202:80   0  124G  0 disk  
root@ip-172-31-23-68:~# file -s /dev/xvdf  
/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)  
root@ip-172-31-23-68:~#
```

It was created. It shows that the already file exists.

Let's detach this volume and attach it to another to the third volume.



Now attach it to the third volume.



Now connect the instance and check whether the file exists or not.

```

root@ip-172-31-29-234:~ Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Sat Feb 10 05:35:56 2024 from 103.88.236.42
ubuntu@ip-172-31-29-234:~$ sudo -i
root@ip-172-31-29-234:~# df -h
Filesystem      size  used Avail Use% Mounted on
/dev/root       7.6G  2.3G  5.3G  31% /
tmpfs           475M   0    475M  0% /dev/shm
tmpfs           190M  864K  190M  1% /run
tmpfs            5.0M   0    5.0M  0% /run/lock
/dev/xvda15     105M  6.1M  99M  6% /boot/efi
tmpfs           95M   4.0K  95M  1% /run/user/1000
root@ip-172-31-29-234:~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0 24.9M  1 loop /snap/amazon-ssm-agent/7628
loop1    7:1    0 55.7M  1 loop /snap/core18/2812
loop2    7:2    0 63.5M  1 loop /snap/core20/2015
loop3    7:3    0 63.9M  1 loop /snap/core20/2105
loop4    7:4    0 111.9M 1 loop /snap/lxd/24322
loop5    7:5    0   87M  1 loop /snap/lxd/27037
loop6    7:6    0  40.9M  1 loop /snap/snapd/20290
loop7    7:7    0  40.4M  1 loop /snap/snapd/20671
xvda   202:0    0   8G  0 disk
└─xvda1  202:1    0   7.9G 0 part /
└─xvda14 202:14   0   4M  0 part
└─xvda15 202:15   0 106M 0 part /boot/efi
xvdf   202:80   0 124G 0 disk
root@ip-172-31-29-234:~# file -s /dev/xvdf
/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
root@ip-172-31-29-234:~#

```

It shows that the already file exists.

2. create efs and attach ebs to 3 different instances in 3 different availability zones.

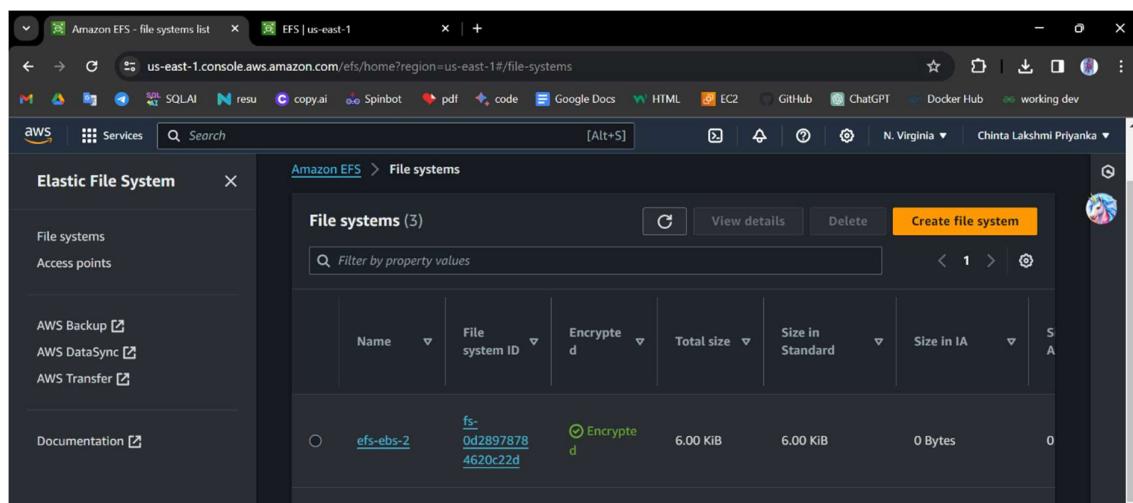
To create an EFS (Elastic File System) and attach an EBS (Elastic Block Store) volume to 3 different instances in 3 different availability zones, Sign in to the AWS Management Console.

1. Open the Amazon EFS console at

<https://console.aws.amazon.com/efs/>.

2. Click the "Create file system" button.
3. In the "Create file system" wizard, configure the file system settings:
 - Choose the desired VPC and specify the appropriate settings for your application(in this the vpc in default mode).
 - Select the availability zones for each mount target you want to create by checking the corresponding checkboxes.

- Configure the appropriate file system settings for throughput and performance.
4. Click the "Create file system" button to create the EFS file system.



The screenshot shows the AWS EFS console with the title bar "Amazon EFS - file systems list" and "EFS | us-east-1". The URL is "us-east-1.console.aws.amazon.com/efs/home?region=us-east-1#/file-systems". The left sidebar has "Elastic File System" selected, with options for "File systems", "Access points", and links to "AWS Backup", "AWS DataSync", "AWS Transfer", and "Documentation". The main area is titled "File systems (3)" and contains a table with columns: Name, File system ID, Encrypted, Total size, Size in Standard, Size in IA, and S. A. The table shows one row for "efs-ebs-2" with the following details: Name: efs-ebs-2, File system ID: fs-0d28978784620c22d, Encrypted: Yes, Total size: 6.00 KiB, Size in Standard: 6.00 KiB, Size in IA: 0 Bytes, and S. A.: 0. There is a "Create file system" button at the top right of the table.

5. Once the file system is created, select it, and note down the file system ID.

Now create 3 instances connect them to the instances and create file systems in different zones.

The screenshot shows the AWS EC2 Instances page. The left sidebar has 'Instances' selected under 'Instances'. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
efs-ebs-2	i-08a58583b6ff13988	Running	t2.micro	Initializing	View alarms
efs-ebs-1	i-03a85a7ea7d41c654	Running	t2.micro	2/2 checks passed	View alarms

A modal window titled 'Select an instance' is open at the bottom.

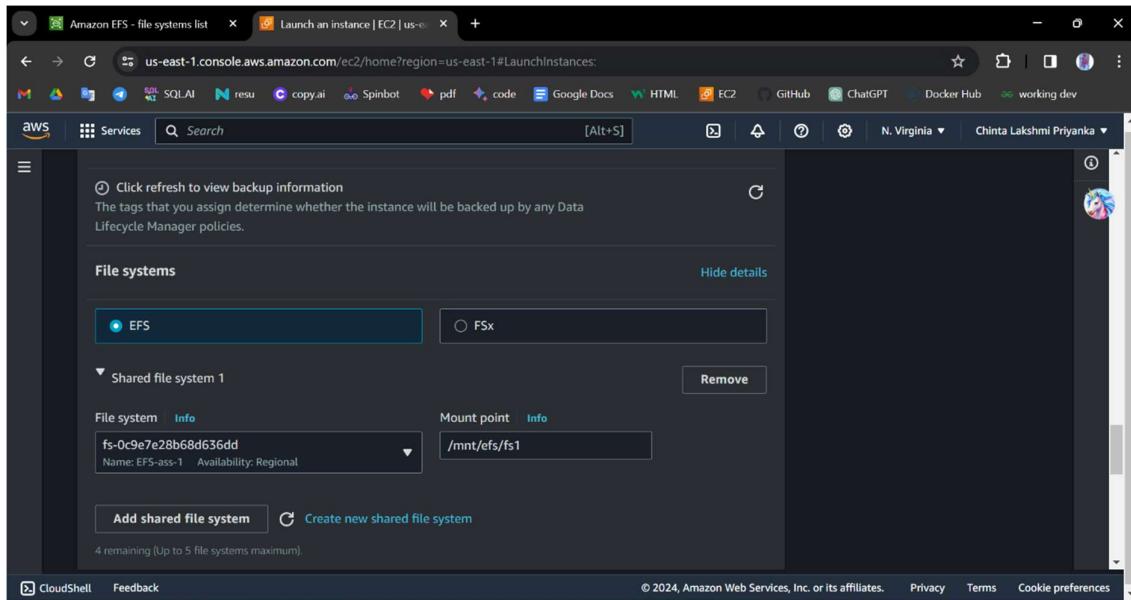
Connect to the instances.

The screenshot shows the 'Launch an instance' wizard. Step 1: Name and tags. The 'Name' field contains 'efs-ebs-1'. Below it, there's a section for 'Application and OS Images (Amazon Machine Image)'. A note says: 'An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.'

Now choose the subnet in the availability zone as us-east-1d.

And then edit network settings. In that choose the subnet add the security group and add the file system we created in the past

Note the mount point:/mnt/efs/fs1



Launch the instance and connect to the instance

Now create a file system

Connect to the instance using sudo -i

Cd /mnt/efs/fs1

Vi file

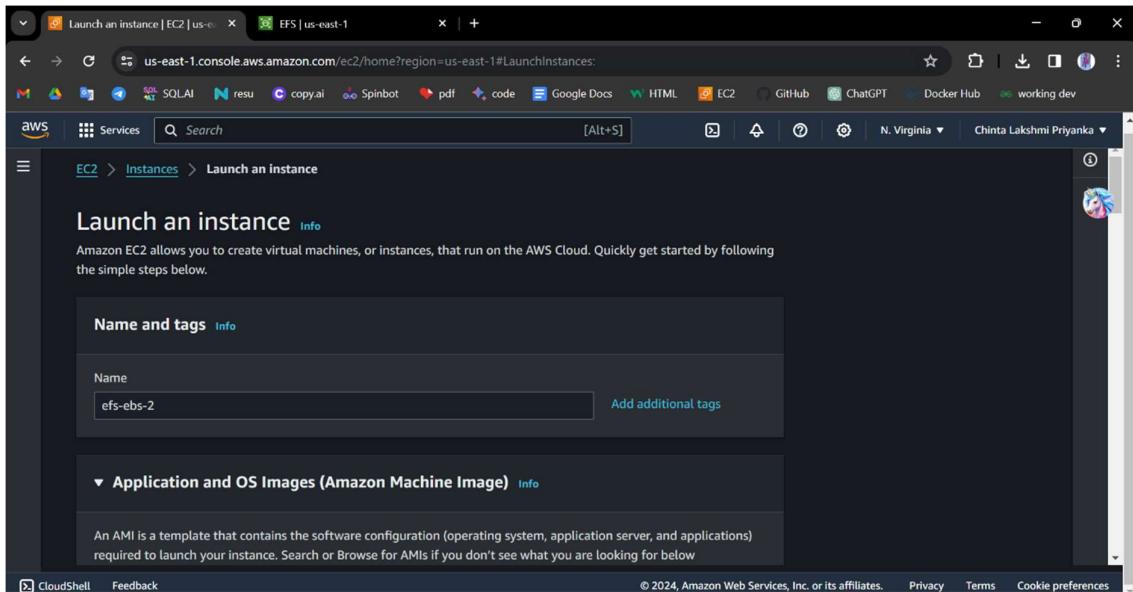
Now create a file in fs1 and it is reflected in the second efs instance

The screenshot shows a terminal window within the AWS CloudShell interface. The user is connected via EC2 Instance Connect to an EC2 instance in the us-east-1 region. The terminal shows a root shell session on a mounted EFS volume. The user runs several commands to update the system and check the contents of the EFS directory. The terminal also displays the instance ID (i-0721eace03ab62ae0) and its public and private IP addresses.

```
~/m/
[ec2-user@ip-172-31-23-176 ~]$ sudo -i
[root@ip-172-31-23-176 ~]# yum update -
Last metadata expiration check: 0:01:36 ago on Sat Feb 10 14:44:42 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-23-176 ~]# cd /mnt
[root@ip-172-31-23-176 mnt]# ls
efs
[root@ip-172-31-23-176 mnt]# cd efs/
[root@ip-172-31-23-176 efs]# ls
fs1
[root@ip-172-31-23-176 efs]# cd fs1/
[root@ip-172-31-23-176 fs1]# vi file1
[root@ip-172-31-23-176 fs1]# ls
file1
[root@ip-172-31-23-176 fs1]#
```

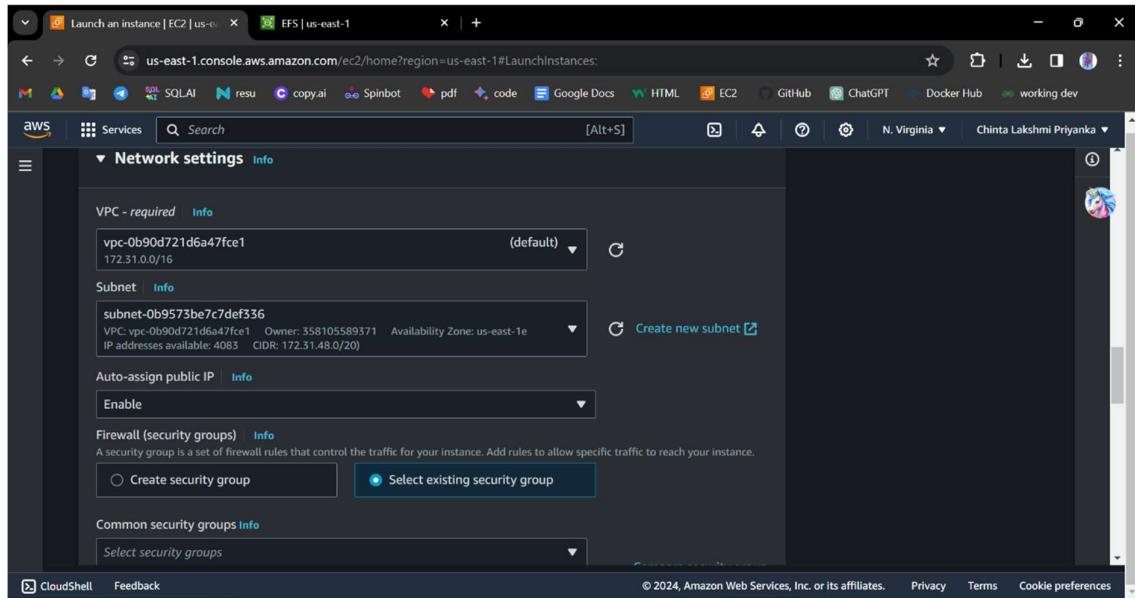
i-0721eace03ab62ae0 (efs-bs-1)
Public IPs: 54.235.17.53 Private IPs: 172.31.23.176

Now create another instance and connect the instance.

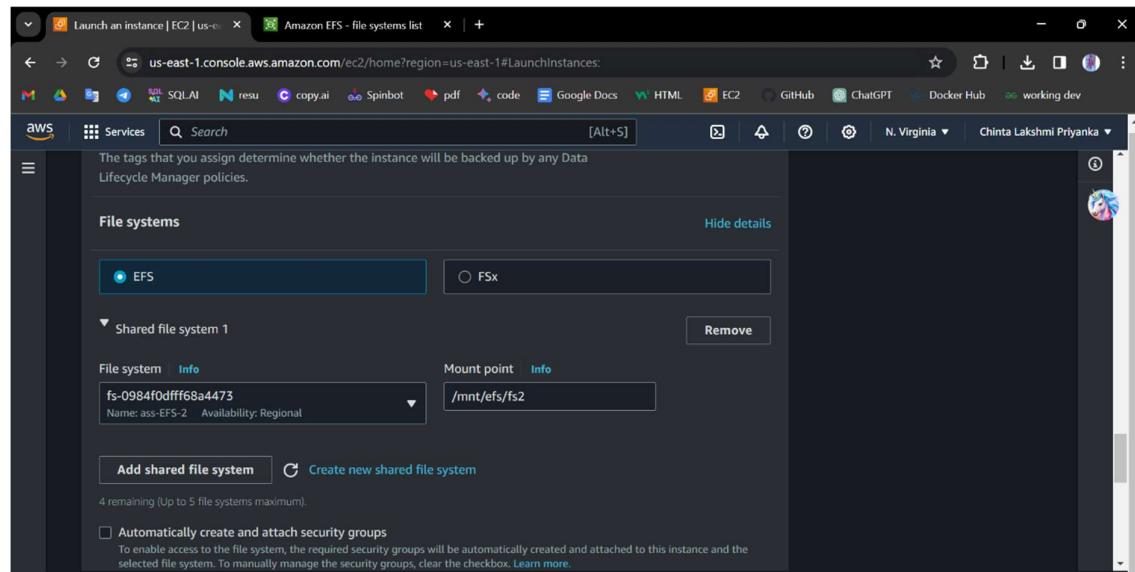


Add existing security group and edit network settings

And choose the availability subnet.



Now add the shared file system



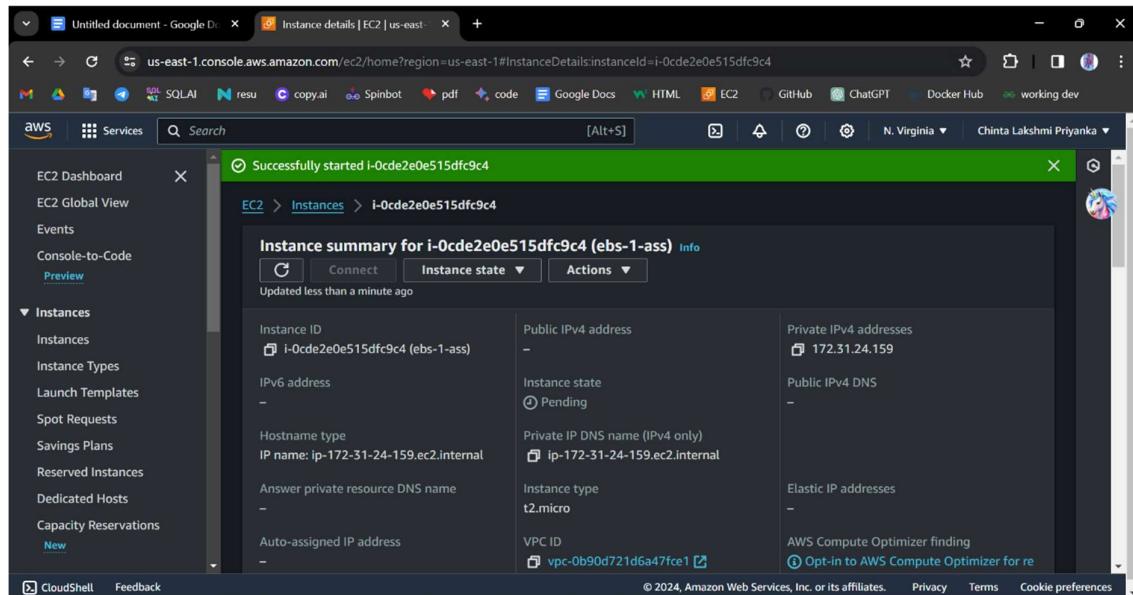
Same as the shown these 2 instances. we have to create a third instance like this.

Now we have to create 3 ebs instances and check whether the file exists in

```
[ec2-user@ip-172-31-53-79 ~]$ sudo -i  
[root@ip-172-31-53-79 ~]# yum update -y  
Last metadata expiration check: 0:01:43 ago on Sat Feb 10 15:19:07 2024.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[root@ip-172-31-53-79 ~]# cd /mnt  
[root@ip-172-31-53-79 mnt]# ls  
efs  
[root@ip-172-31-53-79 mnt]# cd efs/  
[root@ip-172-31-53-79 efs]# ls  
fs2  
[root@ip-172-31-53-79 efs]# cd fs2/  
[root@ip-172-31-53-79 fs2]# ls  
file  
[root@ip-172-31-53-79 fs2]#  
  
i-01f58905536ab904c (efs-ebs-2)  
PublicIPs: 100.25.220.30 PrivateIPs: 172.31.53.79
```

Now launch the first instance and connect the instance.

Now connect the instance



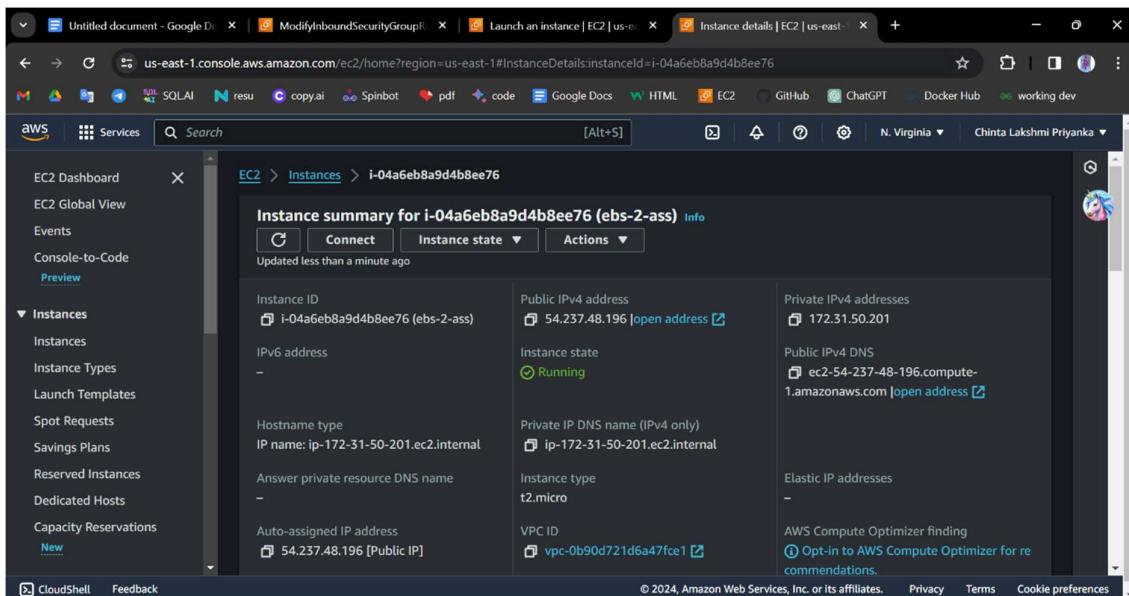
A screenshot of an AWS CloudShell terminal window. The terminal shows a root shell on an EC2 instance. The user navigates to a directory named 'priya' and lists its contents. Then, they change to a sub-directory 'chinta' and list its contents. Finally, they mount a new volume from Amazon EBS and check the disk usage again. The terminal window also displays the instance ID and its public and private IP addresses.

```
priya@ip-172-31-24-159:~# cd priya/
priya@ip-172-31-24-159:~/priya# ls
chinta
priya@ip-172-31-24-159:~/priya# cd chinta/
priya@ip-172-31-24-159:~/priya/chinta# ls
priya@ip-172-31-24-159:~/priya/chinta# cd ..
priya@ip-172-31-24-159:~/priya# mount /dev/xvdg priya/chinta
priya@ip-172-31-24-159:~# df -h
Filesystem      Size   Used  Avail Use% Mounted on
/dev/root       7.6G   6.0G  21% /
tmpfs          475M     0  475M  0% /dev/shm
tmpfs          190M  852K  190M  1% /run
tmpfs          5.0M     0  5.0M  0% /run/lock
/dev/xvda15    105M   6.1M  99M  6% /boot/efi
tmpfs          95M   4.0K  95M  1% /run/user/1000
/dev/xvdg      101G  754M  101G  1% /root/priya/chinta
priya@ip-172-31-24-159:~#
```

i-0cd2e0e515dfc9c4 (ebs-1-ass)
PublicIPs: 54.242.78.195 PrivateIPs: 172.31.24.159

In this we have created a file system Now check if it was reflected to another or not.

Create a second instance



Now we have to check if there is any existing file in the system

Using file -s /dev/xvdf . If it does not exist it will go to the next step if not

We have to create a file system.

```
realtime =none          extsz=4096  blocks=0, rtextents=0
[root@ip-172-31-62-122 ~]# 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  2.9M 188M  2% /run
/dev/xvda1     8.0G  1.6G  6.5G 19% /
tmpfs          475M   0  475M  0% /tmp
/dev/xvda128   10M  1.3M  8.7M 13% /boot/efi
tmpfs          95M   0  95M  0% /run/user/1000
[root@ip-172-31-62-122 ~]# 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 /boot/efi
xvdf  202:80   0  102G 0 disk
[root@ip-172-31-62-122 ~]# i-0f2ef41644808c8df (ebs -2-ass)
PublicIPs: 100.25.132.133 PrivateIPs: 172.31.62.122
```

In this, the file was not reflected. We have to create a file system

Now as the 2 instances we have to create a third instance

And check whether the file system is created or not

It was not created.

CONCLUSION: Only in EFS did we have to create a file system and it was reflected in another instance. But in the EBS it was not reflected in another instance we had to create a new file system.

----- End -----