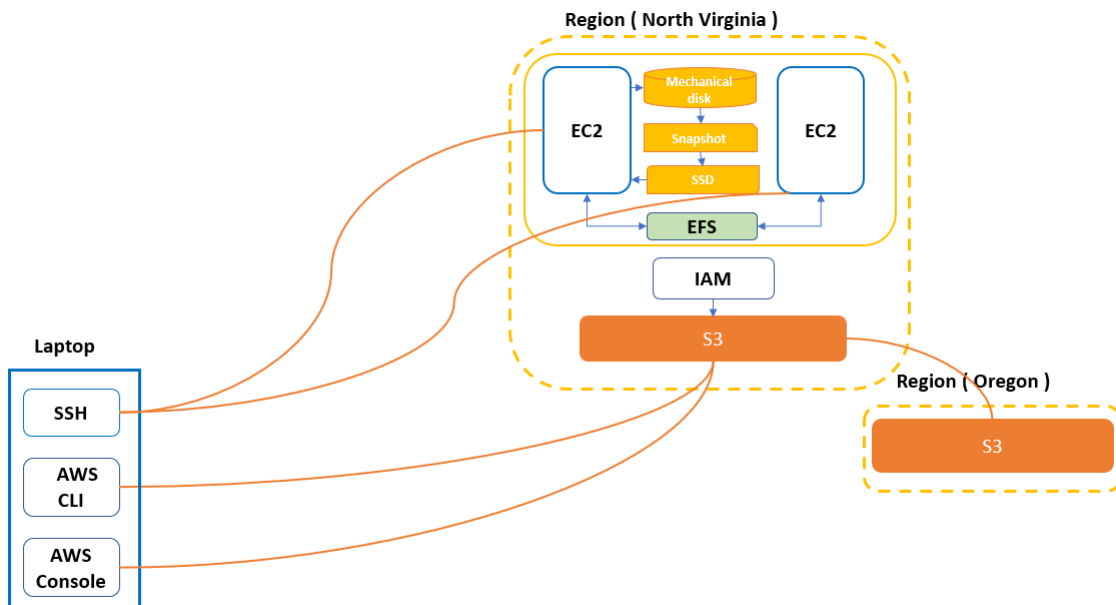


## EBS | S3 | EFS | AWS CLI | IAM

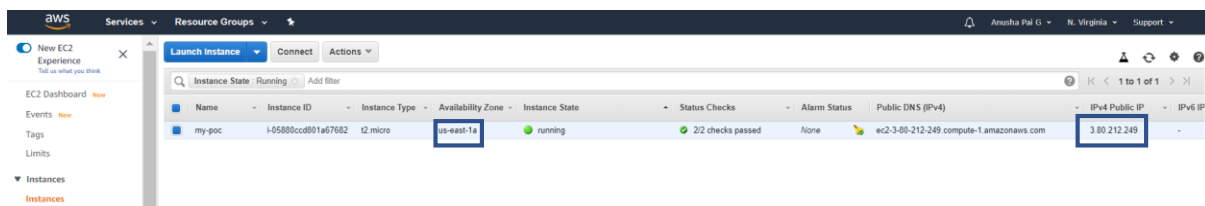
Add volumes to EC2 instance, migrate data from one volume to the other, write a CLI to upload documents to S3 from local machine, cross region replication.

### FINAL OUTCOME



How to do it?

1. Create an EC2 instance using the 7- step workflow
  - a) Use Amazon Linux AMI in AZ1



- b) Download the PEM file and SSH to the instance

```
PS C:\Users\Anusha> ssh -i .\lab.pem ec2-user@3.80.212.249
Last login: Sun Aug 30 17:17:19 2020 from 117.201.204.251
```

```

  __|  __|_ )
 _| (    /   Amazon Linux 2 AMI
---| \___|___|

```

```

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-84-210 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1     202:1    0   8G  0 part /
[ec2-user@ip-172-31-84-210 ~]$
```

2. Create a new volume and attach it to EC2
  - a) Grab a 10GB magnetic disk in the same AZ1

Volume Type: Magnetic (standard)

Size (GiB): 10 (Min: 1 GiB, Max: 1024 GiB)

IOPS: Not applicable

Availability Zone: us-east-1a

Throughput (MB/s): Not applicable

Snapshot ID: Select a snapshot

Encryption: ☐ Encrypt this volume

Key (128 characters maximum): Value (256 characters maximum)

Name: my-vol

Add Tag: 49 remaining (Up to 50 tags maximum)

\* Required

Cancel Create Volume

- b) Attach the volume to the instance

Attach Volume

Volume: vol-0ab4ab6d3b4093526 (my-vol) in us-east-1a

Instance: i-0580cc801a67682 in us-east-1a

Device: /dev/sdf

Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel Attach

- c) Format the volume and mount it

```

[root@ip-172-31-84-210 ~]# 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  10G  0  disk
[root@ip-172-31-84-210 ~]# file -s /dev/xvdf
/dev/xvdf: data
[root@ip-172-31-84-210 ~]# mkfs.ext4 /dev/xvdf
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stripe=0 blocks, Stripe width=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-84-210 ~]# mkdir /appdata
[root@ip-172-31-84-210 ~]# mount /dev/xvdf /appdata
[root@ip-172-31-84-210 ~]# df -hT /appdata/
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/xvdf       ext4   9.8G   37M   9.2G   1% /appdata
[root@ip-172-31-84-210 ~]# |

```

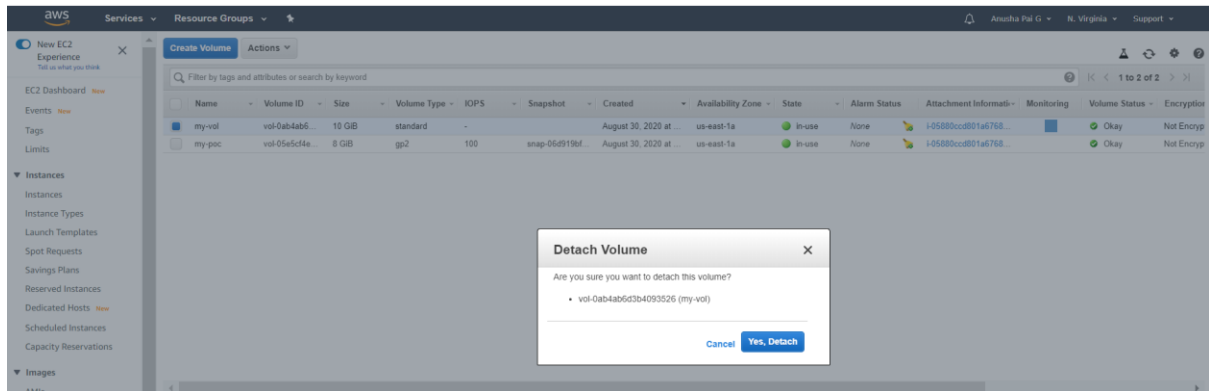
d) Create a sample text file in the volume to simulate data creation

```

[root@ip-172-31-84-210 ~]# cd /appdata/
[root@ip-172-31-84-210 appdata]# ls
lost+found
[root@ip-172-31-84-210 appdata]# echo "This is a sample test file" > sample.txt
[root@ip-172-31-84-210 appdata]# ls
lost+found sample.txt
[root@ip-172-31-84-210 appdata]# cat sample.txt
This is a sample test file
[root@ip-172-31-84-210 appdata]# cd
[root@ip-172-31-84-210 ~]# umount /appdata/
[root@ip-172-31-84-210 ~]# |

```

e) Unmount, detach the volume



3. Create a snapshot
  - a) Create a snapshot of the detached volume

**Create Snapshot**

Volume: vol-0ab4ab6d3b4093526

Description: This is my sample data

Encrypted: ☒ Not Encrypted

Key (128 characters maximum): Value (256 characters maximum)

Name: my-data

Add Tag: 49 remaining (Up to 50 tags maximum)

\* Required Cancel Create Snapshot

- b) Create a new SSD volume of size 15GB and attach the snapshot to it

**Create Volume**

Snapshot ID: snap-09f7b3770eb71f6 (my-data)

Volume Type: General Purpose SSD (gp2)

Size (GiB): 15 (Min: 1 GiB, Max: 16384 GiB)

IOPS: 100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)

Availability Zone: us-east-1a

Fast Snapshot Restore: Not enabled

Throughput (MB/s): Not applicable

Encryption: ☐ Encrypt this volume

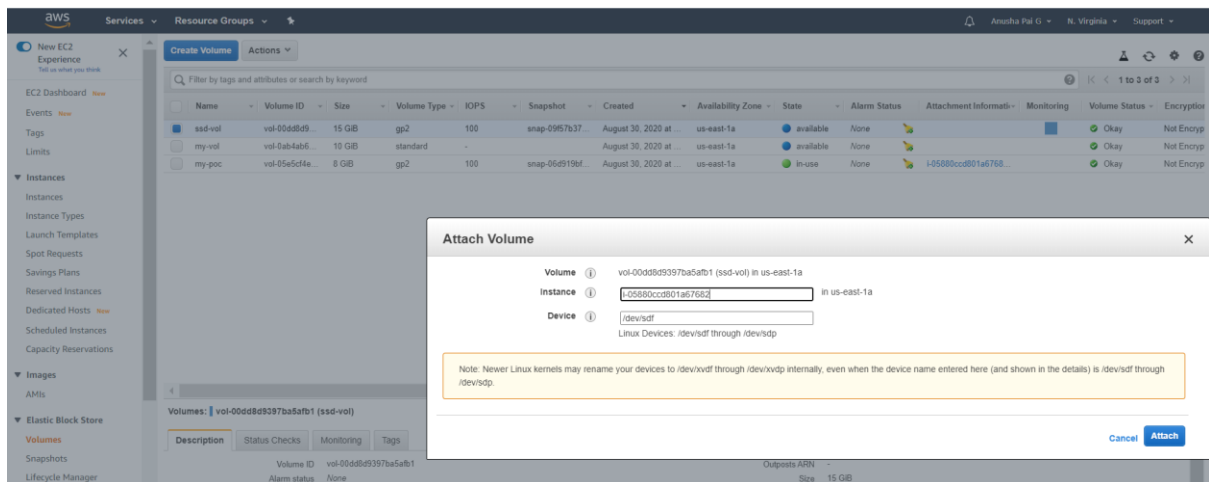
Key (128 characters maximum): Value (256 characters maximum)

Name: ssd-vol

Add Tag: 49 remaining (Up to 50 tags maximum)

\* Required Cancel Create Volume

- c) Attach the new disk to the EC2 instance, mount and check for the data availability

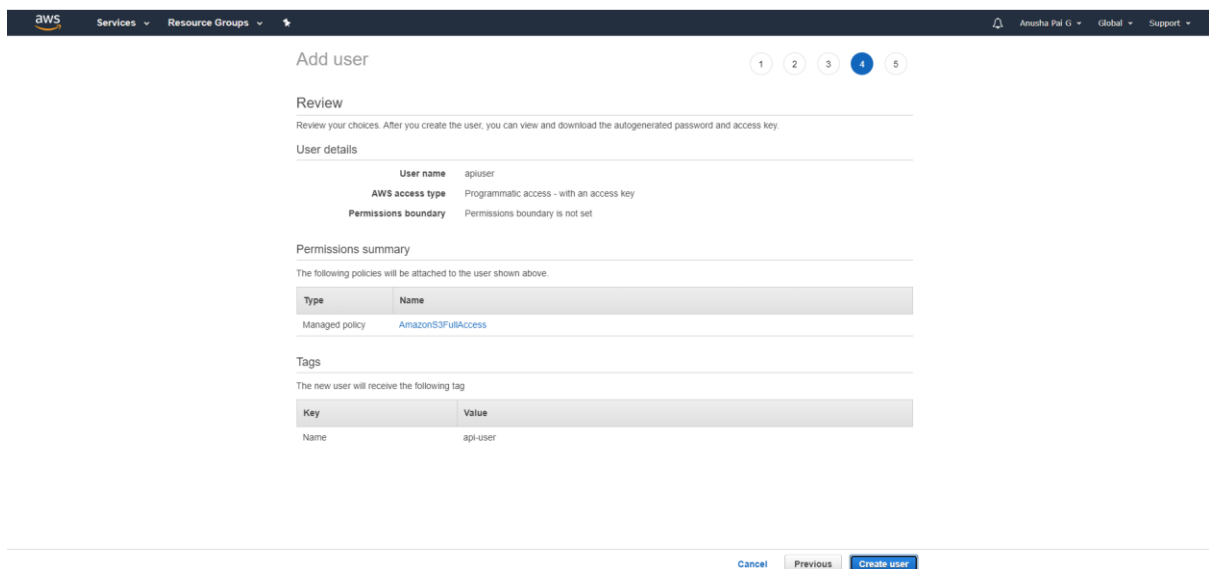


```
[root@ip-172-31-84-210 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk 
└─xvda1     202:1    0   8G  0 part /
xvdf        202:8    0  15G  0 disk 
[root@ip-172-31-84-210 ~]# file -s /dev/xvdf
/dev/xvdf: Linux rev 1.0 ext4 filesystem data, UUID=b26fe0f9-c8b8-4dd3-a449-4574e59ab866 (extents) (64bit) (large files) (huge files)
[root@ip-172-31-84-210 ~]# cd /appdata/
[root@ip-172-31-84-210 appdata]# ls
[root@ip-172-31-84-210 appdata]# cd
[root@ip-172-31-84-210 ~]#
[root@ip-172-31-84-210 ~]# mount /dev/xvdf /appdata/
[root@ip-172-31-84-210 ~]#
[root@ip-172-31-84-210 ~]# cd /appdata/
[root@ip-172-31-84-210 appdata]# ls
lost+found  sample.txt
[root@ip-172-31-84-210 appdata]# cat sample.txt
This is a sample test file
[root@ip-172-31-84-210 appdata]#
```

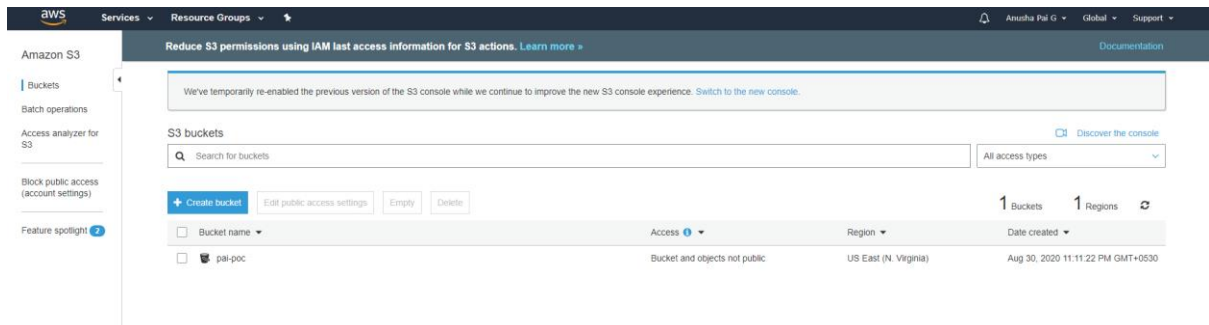
#### 4. AWS CLI setup

##### a) Go to IAM

- Create the user for the CLI access, name it "apiuser"
- Attach the AWS managed policy "S3FullAccess"
- Download the CSV file having the access key + secret key



- b) CLI setup
  - Launch a terminal on your local machine and configure aws using the key pair
- 5. Use the S3 console to create a bucket



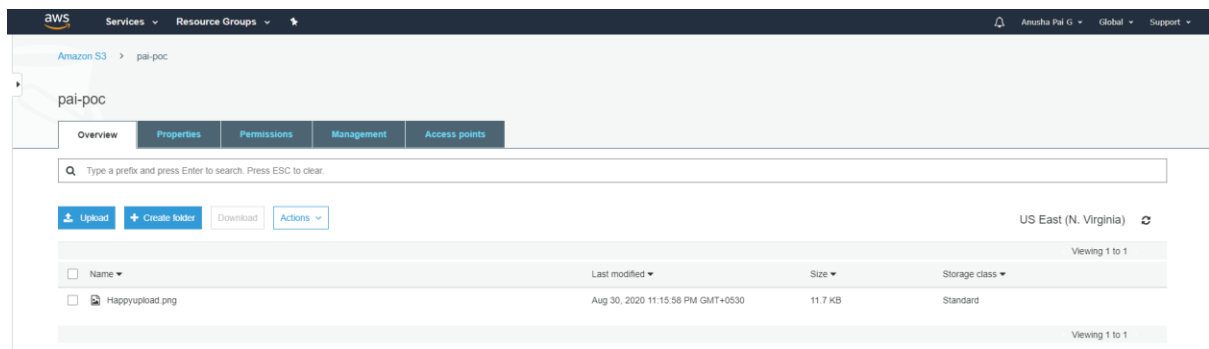
- 6. Use the CLI
  - a) To upload a file from your local machine to the S3 bucket

```
PS C:\Users\Anusha> aws s3 cp '.\Pictures\Saved Pictures\Happyface.png' s3://pai-poc/Happyupload.png
upload: Pictures\Saved Pictures\Happyface.png to s3://pai-poc/Happyupload.png
```

- b) List the buckets

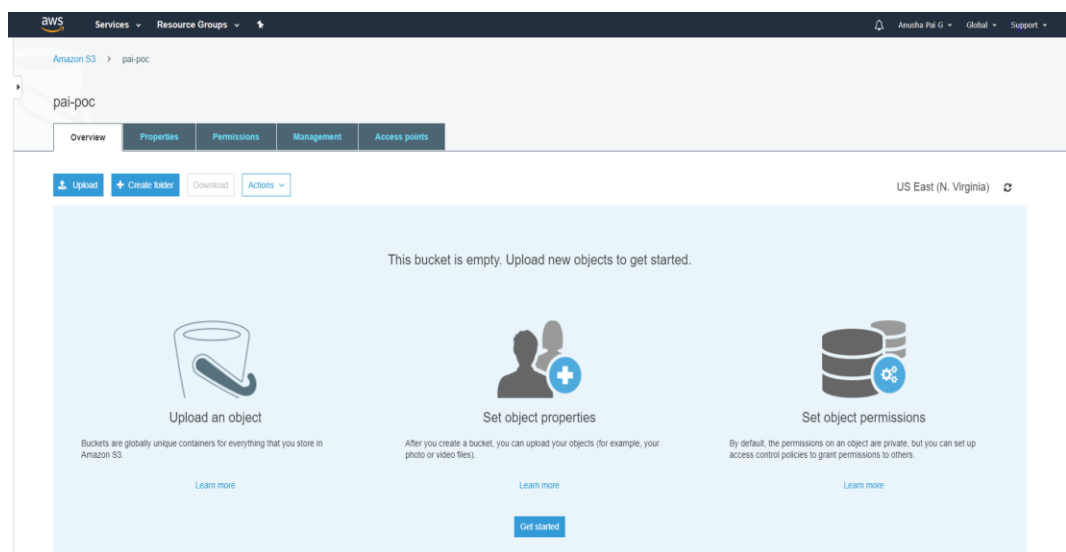
```
PS C:\Users\Anusha> aws s3 ls
2020-08-30 23:11:22 pai-poc
```

- c) List the contents of the bucket

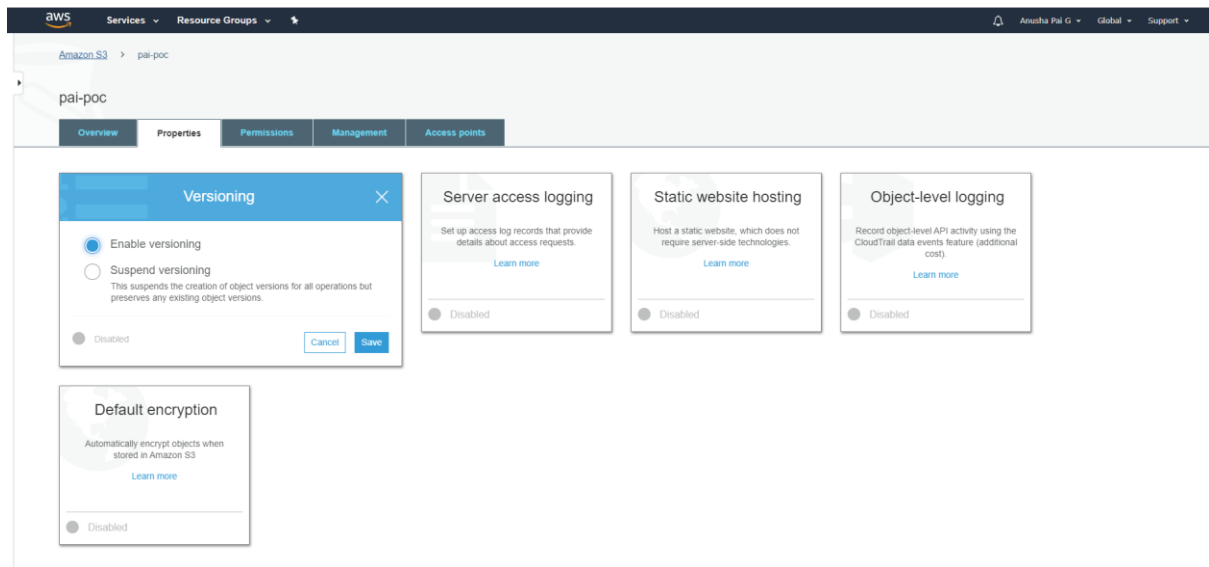


- d) Remove the object in the bucket

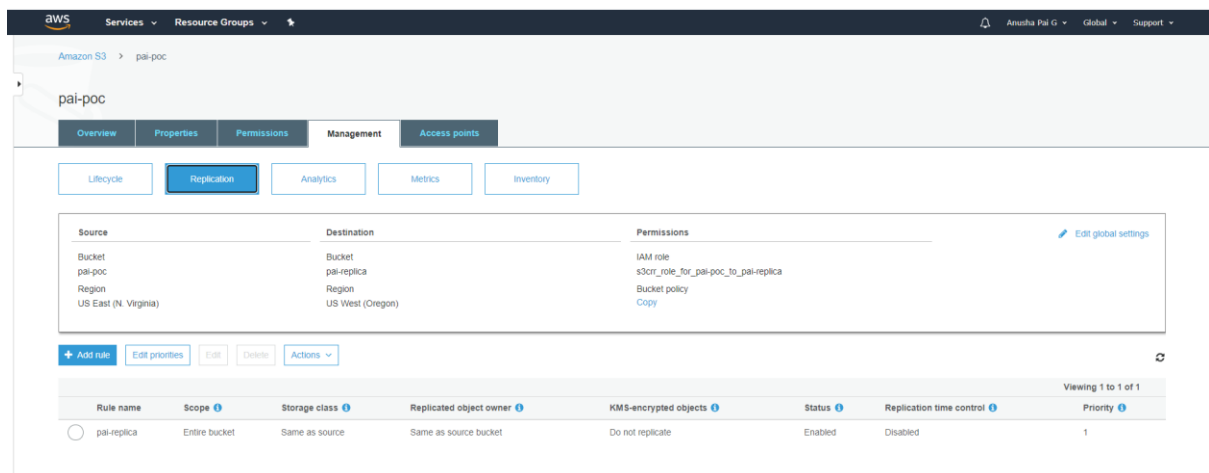
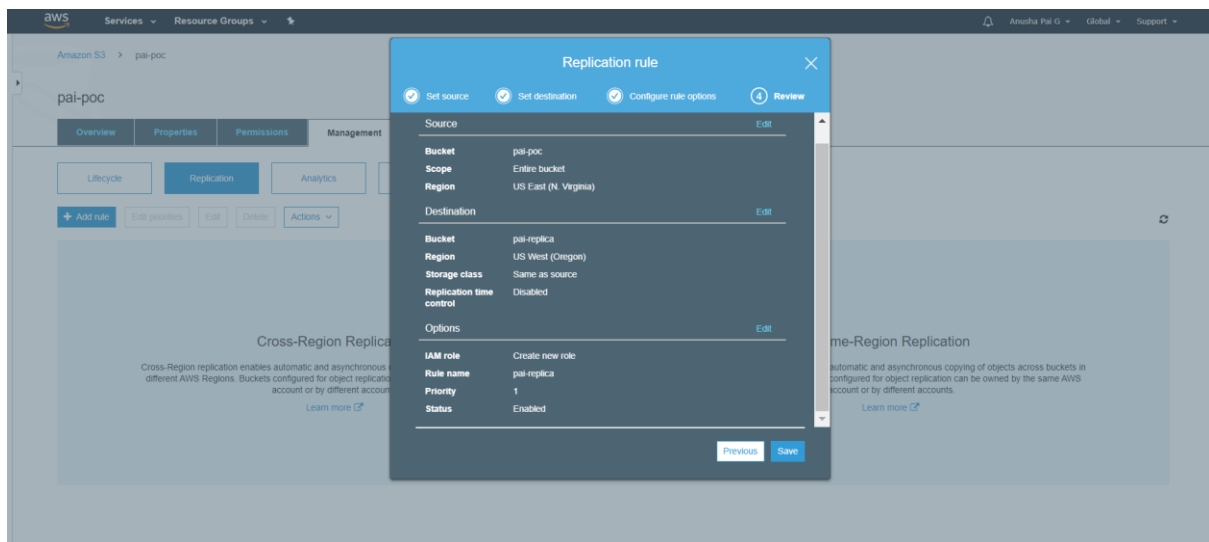
```
PS C:\Users\Anusha> aws s3 rm s3://pai-poc/Happyupload.png
delete: s3://pai-poc/Happyupload.png
```

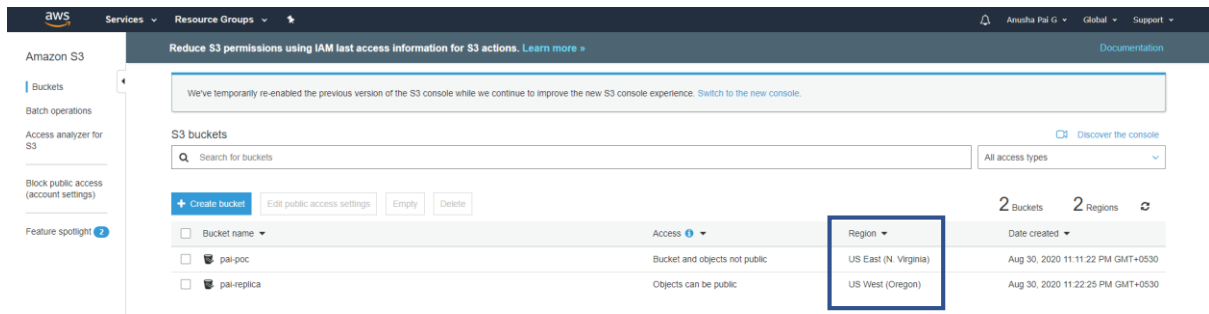


## 7. Enable versioning for the bucket



## 8. Enable cross region replication

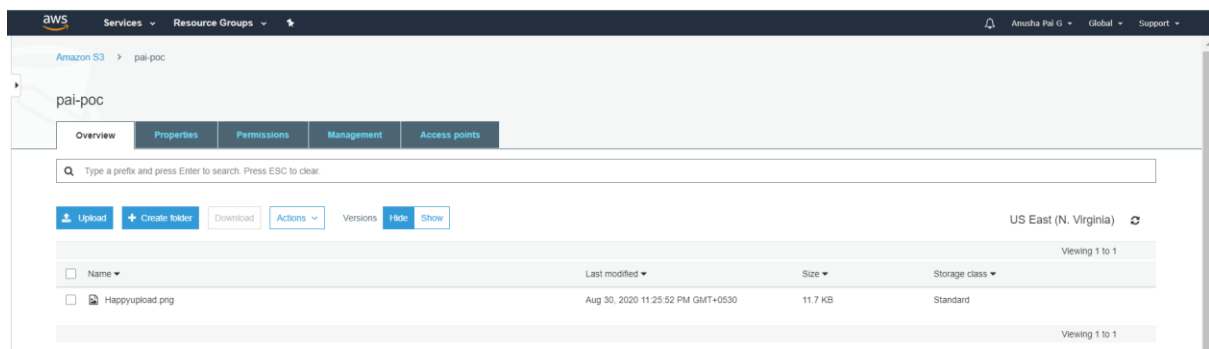




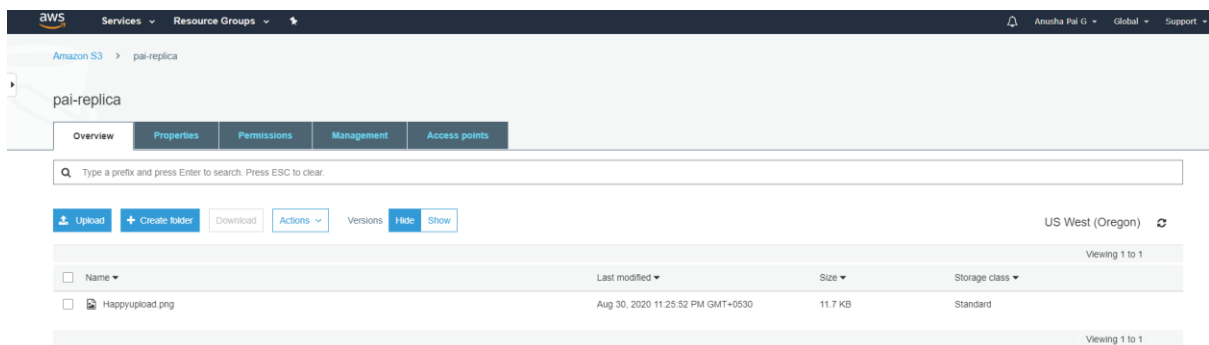
## TEST CASES:

- 1) Upload a file from the CLI to the original bucket and see if the content is replicated in the destination bucket

## SOURCE BUCKET



## DESTINATION BUCKET



- 2) Upload the same file again and check if versioning is reflecting in both the buckets

## SOURCE BUCKET



aws Services Resource Groups

Amazon S3 > pai-poc

pai-poc

Overview Properties Permissions Management Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions Versions Hide Show US East (N. Virginia)

Name	Version ID	Last modified	Size	Storage class
Happyupload.png				
Aug 30, 2020 11:27:16 PM (Latest version)	PQG.n7OF2vGnd96RnQrg_M6mntlQtb	Aug 30, 2020 11:27:16 PM	11.7 KB	Standard
Aug 30, 2020 11:25:52 PM	RTM9HnsdPW5.7zwAOAy3Gz7NECuNAvIP		11.7 KB	Standard

Viewing 1 to 2

## DESTINATION BUCKET

aws Services Resource Groups

Amazon S3 > pai-replica

pai-replica

Overview Properties Permissions Management Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions Versions Hide Show US West (Oregon)

Name	Version ID	Last modified	Size	Storage class
Happyupload.png				
Aug 30, 2020 11:27:16 PM (Latest version)	PQG.n7OF2vGnd96RnQrg_M6mntlQtb	Aug 30, 2020 11:27:16 PM	11.7 KB	Standard
Aug 30, 2020 11:25:52 PM	RTM9HnsdPW5.7zwAOAy3Gz7NECuNAvIP		11.7 KB	Standard

Viewing 1 to 2

3) Delete a file from the source bucket and see the effect on the destination bucket

## SOURCE BUCKET

aws Services Resource Groups

Amazon S3 > pai-poc

pai-poc

Overview Properties Permissions Management Access points

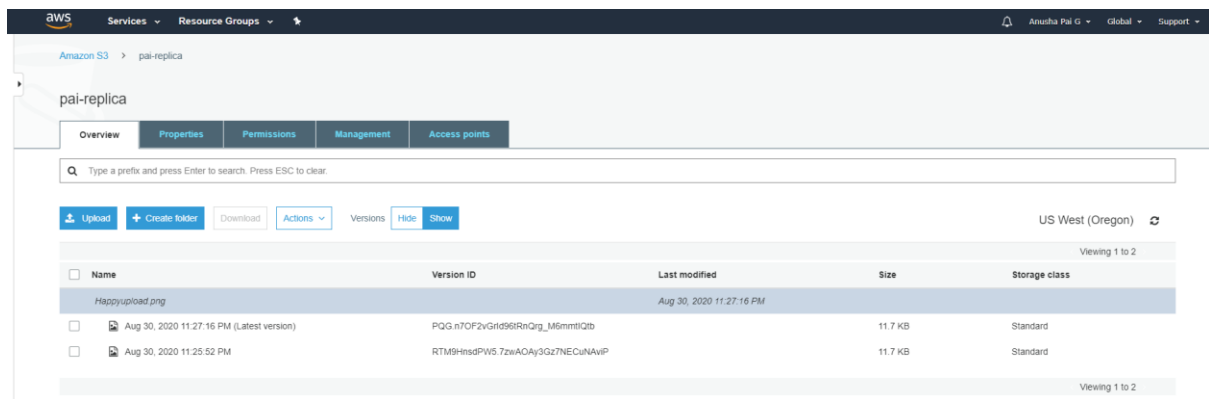
Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions Versions Hide Show US East (N. Virginia)

Name	Version ID	Last modified	Size	Storage class
Happyupload.png				
Aug 30, 2020 11:25:52 PM (Latest version)	RTM9HnsdPW5.7zwAOAy3Gz7NECuNAvIP	Aug 30, 2020 11:25:52 PM	11.7 KB	Standard

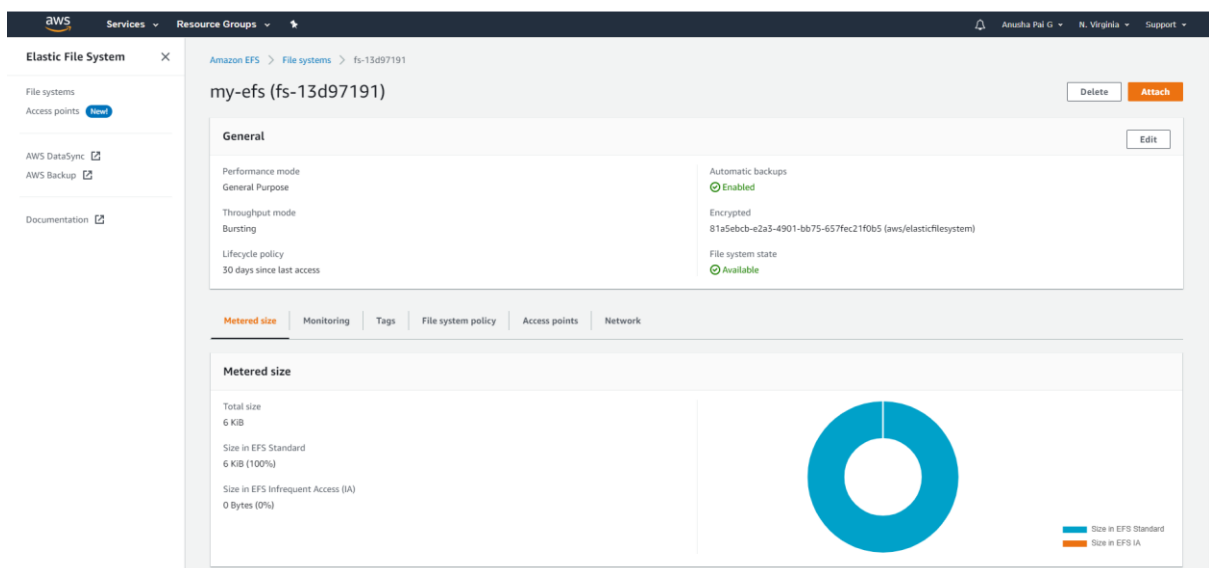
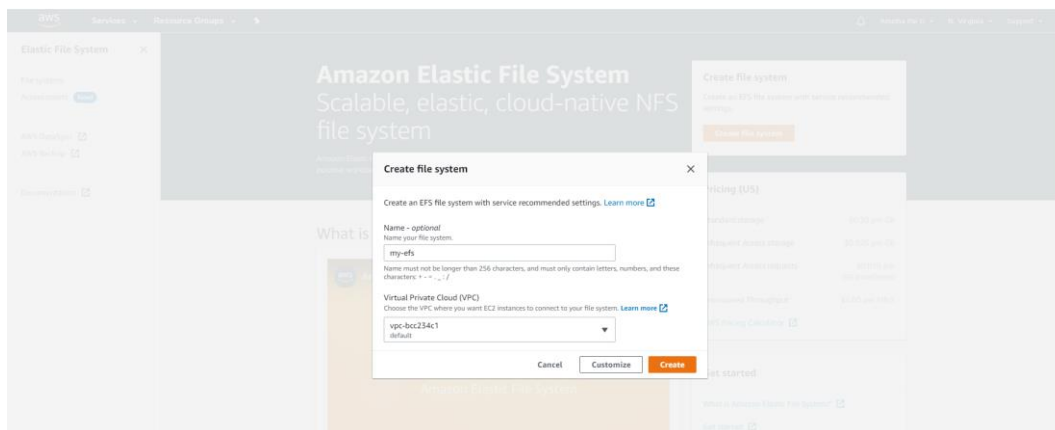
Viewing 1 to 1

## DESTINATION BUCKET



## 9. EFS

- Launch another EC2 instance and SSH to it
- Create an EFS



- Mount the EFS on both the EC2 instances
- Create a file in one of the instances and see if it replicates in the other instance
- Unmount the EFS

```
[root@ip-172-31-84-210 ~]# mkdir efs
[root@ip-172-31-84-210 ~]# mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-13d97191.efs.us-east-1.amazonaws.com:/ efs
[root@ip-172-31-84-210 ~]# cd efs
[root@ip-172-31-84-210 efs]# ls
[root@ip-172-31-84-210 efs]# touch myefsfile
[root@ip-172-31-84-210 efs]# echo "This is a sample text created for data replication" > myefsfile
[root@ip-172-31-84-210 efs]# ls
myefsfile
[root@ip-172-31-84-210 efs]# cat myefsfile
This is a sample text created for data replication
[root@ip-172-31-84-210 efs]# |
```

```
[root@ip-172-31-20-216 ~]# mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-13d97191.efs.us-east-1.amazonaws.com:/ efs
[root@ip-172-31-20-216 ~]# cd efs
[root@ip-172-31-20-216 efs]# ls
[root@ip-172-31-20-216 efs]# ls
myefsfile
[root@ip-172-31-20-216 efs]# cat myefsfile
This is a sample text created for data replication
[root@ip-172-31-20-216 efs]# |
```