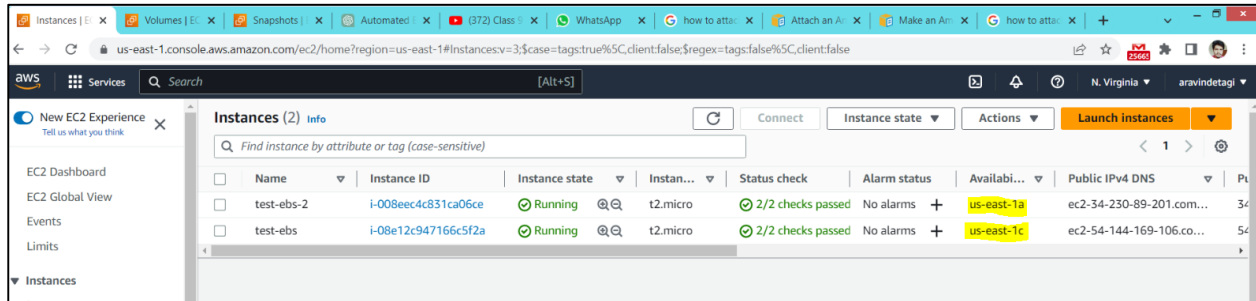


Assignment: 1: Create an instance, attach a new EBS volume, add some files to EBS, Detach the EBS volume and attach it to another instance in different availability zone

Created 2 EC2 instances in two different availability zones

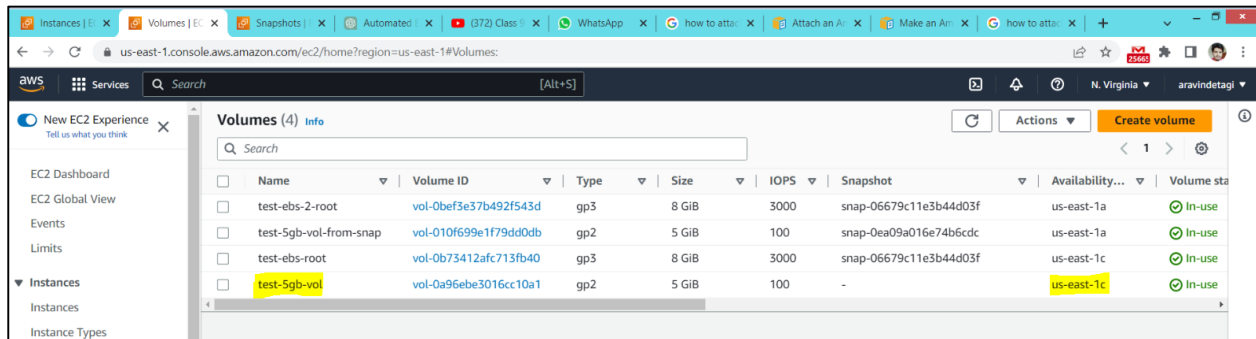
test-ebs → us-east-1c & test-ebs-2 → us-east-1a



The screenshot shows the AWS Management Console 'Instances' page. Two EC2 instances are listed: 'test-ebs-2' and 'test-ebs'. Both are in the 'Running' state, using 't2.micro' instances, and have '2/2 checks passed'. The availability zones are highlighted in yellow: 'us-east-1a' for 'test-ebs-2' and 'us-east-1c' for 'test-ebs'.

Name	Instance ID	Instance state	Instan...	Status check	Alarm status	Availabi...	Public IPv4 DNS
test-ebs-2	i-008eec4c831ca06ce	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-230-89-201.com...
test-ebs	i-08e12c947166c5f2a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-54-144-169-106.co...

Created new EBS volume in us-east-1c AZ and attached it to **test-ebs** instance



The screenshot shows the AWS Management Console 'Volumes' page. Four EBS volumes are listed. The volume 'test-5gb-vol' is highlighted in yellow and is in the 'In-use' state, attached to the 'test-ebs' instance in the 'us-east-1c' availability zone.

Name	Volume ID	Type	Size	IOPS	Snapshot	Availability...	Volume sta
test-ebs-2-root	vol-0bef3e37b492f543d	gp3	8 GiB	3000	snap-06679c11e3b44d03f	us-east-1a	In-use
test-5gb-vol-from-snap	vol-010f699e1f79dd0db	gp2	5 GiB	100	snap-0ea09a016e74b6cdc	us-east-1a	In-use
test-ebs-root	vol-0b73412afc713fb40	gp3	8 GiB	3000	snap-06679c11e3b44d03f	us-east-1c	In-use
test-5gb-vol	vol-0a96ebe3016cc10a1	gp2	5 GiB	100	-	us-east-1c	In-use

Created partition and mounted the partition

```
ec2-user@ip-172-31-6-111 ~]$ 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   5G  0 disk
(ec2-user@ip-172-31-6-111 ~)$ fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

fdisk: cannot open /dev/xvdf: Permission denied
(ec2-user@ip-172-31-6-111 ~)$ sudo su
[root@ip-172-31-6-111 ec2-user]# fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x2a5312ba.

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 (2048-10485759, default 2048):
Last sector, +/-sectors or +/-size[K,M,G,T,P] (2048-10485759, default 10485759):

Created a new partition 1 of type 'Linux' and of size 5 GiB.

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

[root@ip-172-31-6-111 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   5G  0 disk
|-xvdf1     202:81   0   5G  0 part
[root@ip-172-31-6-111 ec2-user]# mkfs.ext4 /dev/xvdf1
mke2fs 1.46.5 (30-Dec-2021)
```

1. Checking all the attached volumes

2. Creating one partition

3. Making sure that the partition is created

```
Calling ioctl() to re-read partition table.
Syncing disks.

[root@ip-172-31-6-111 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   5G  0 disk
|-xvdf1     202:81   0   5G  0 part
[root@ip-172-31-6-111 ec2-user]# mkfs.ext4 /dev/xvdf1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1310464 4k blocks and 327680 inodes
Filesystem UUID: d857b90c-2bcb-4645-809a-c1c97a67e6b4
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-6-111 ec2-user]# cd
[root@ip-172-31-6-111 ~]# mkdir disk
[root@ip-172-31-6-111 ~]# pwd
/root
[root@ip-172-31-6-111 ~]# mount /dev/xvdf1 /disk
mount: /disk: mount point does not exist.
[root@ip-172-31-6-111 ~]# mount /dev/xvdf1 disk
[root@ip-172-31-6-111 ~]# 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  19% /
tmpfs           475M   0  475M   0% /tmp
tmpfs           95M   0   95M   0% /run/user/1000
/dev/xvdf1      4.9G  24K  4.9G   1% /root/disk
[root@ip-172-31-6-111 ~]# cd disk
[root@ip-172-31-6-111 disk]# touch f{1..20}.txt
[root@ip-172-31-6-111 disk]# ls
f1.txt f10.txt f11.txt f12.txt f13.txt f14.txt f15.txt f16.txt f17.txt f18.txt f19.txt f2.txt f20.txt f3.txt f4.txt f5.txt f6.txt f7.txt f8.txt f9.txt lost+found
[root@ip-172-31-6-111 disk]# cd
[root@ip-172-31-6-111 ~]#
Broadcast message from root@ip-172-31-6-111.ec2.internal (Fri 2023-06-16 01:32:19 UTC):

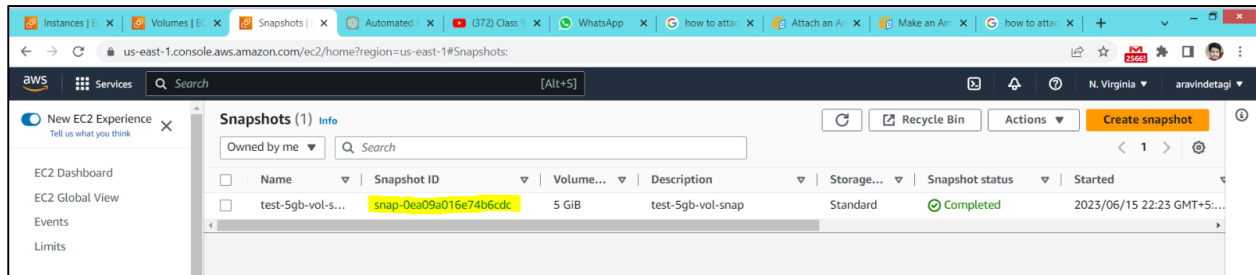
The system will power off now!
```

4. Formatting the partition before using

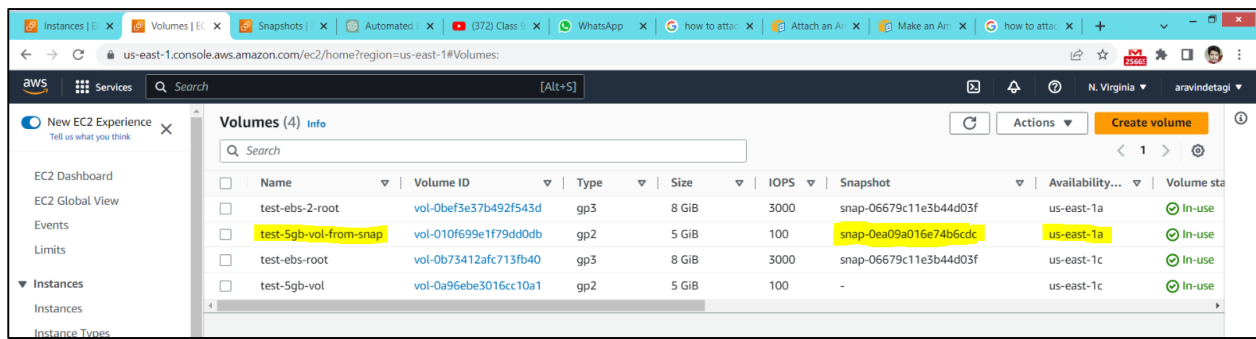
5. Mounting the partition to a directory

6. Creating some files inside the mounted volume

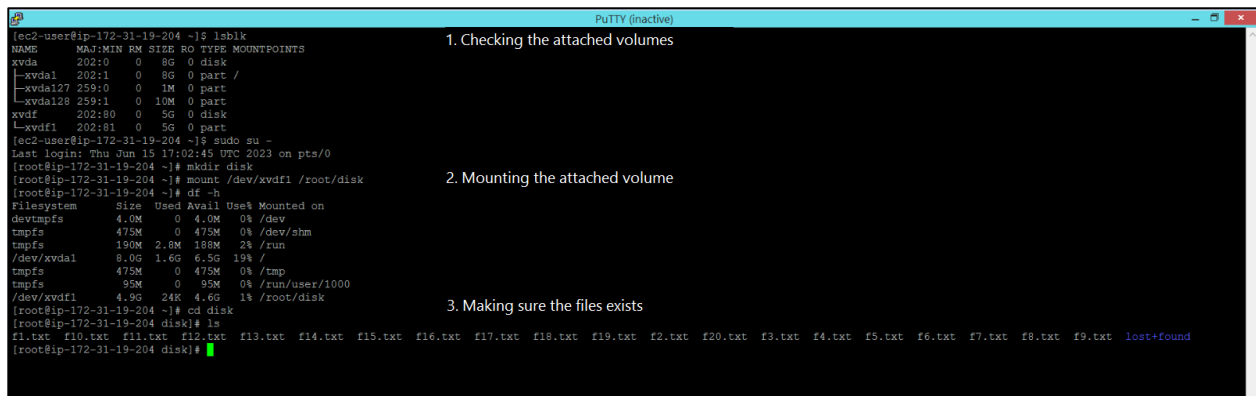
Created a snapshot from *test-5gb-vol* volume



Created a volume from snapshot to us-east-1a AZ and attached it to an instance in new AZ (us-east-1a)

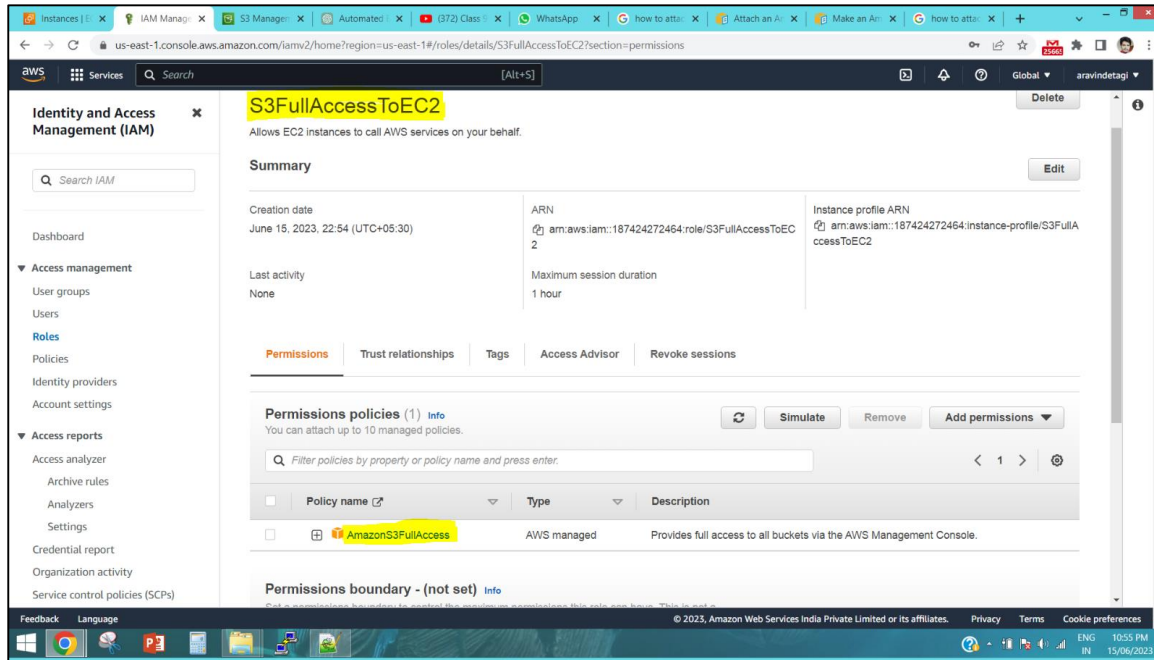


Mounted the volume and made sure that files exists

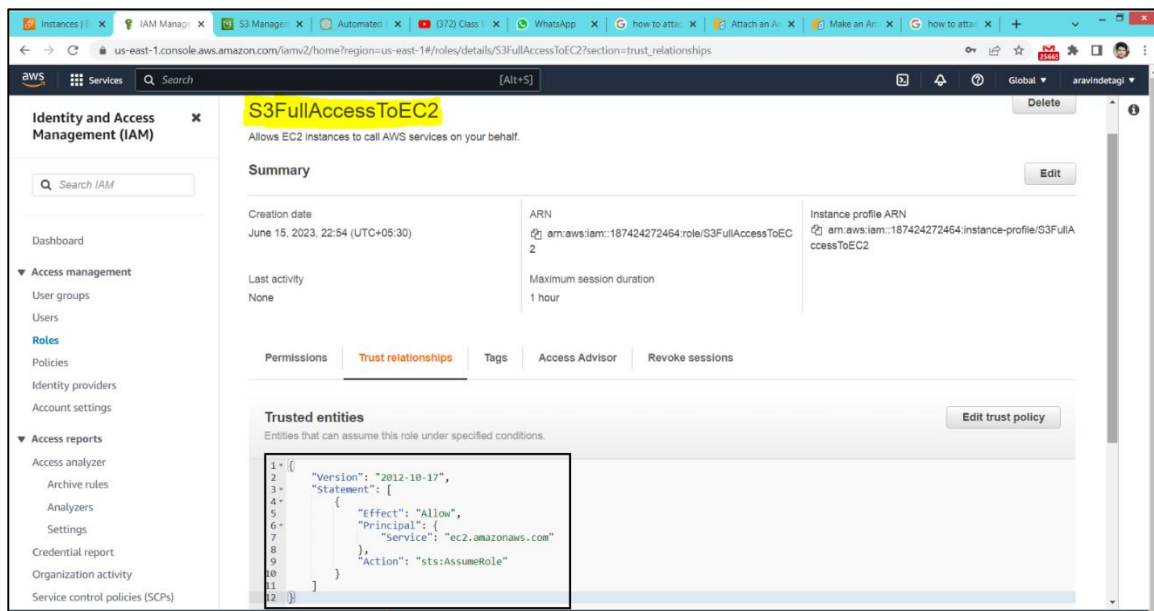


Assignment: 2: Create S3 bucket from aws cli, copy few files from ec2 to s3 bucket through aws cli command. Configure lifecycle management to move files to glacier after one day of upload.

Created an IAM role **S3FullAccessToEC2** and added **AmazonS3FullAccess** policy and made ec2.amazonaws.com service as principal



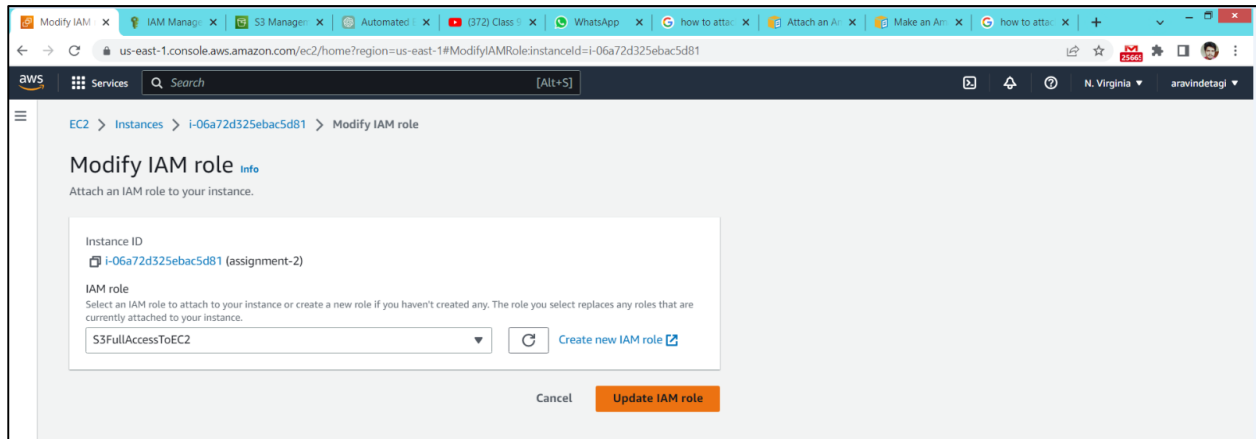
The screenshot shows the AWS IAM console for the role **S3FullAccessToEC2**. The role is created on June 15, 2023, at 22:54 UTC+05:30. It has an ARN of `arn:aws:iam::187424272464:role/S3FullAccessToEC2` and an instance profile ARN of `arn:aws:iam::187424272464:instance-profile/S3FullAccessToEC2`. The role is currently inactive. The **Permissions** tab is selected, showing one attached policy: **AmazonS3FullAccess**, which is an AWS managed policy providing full access to all buckets via the AWS Management Console.



The screenshot shows the AWS IAM console for the role **S3FullAccessToEC2**, with the **Trust relationships** tab selected. The role is created on June 15, 2023, at 22:54 UTC+05:30. It has an ARN of `arn:aws:iam::187424272464:role/S3FullAccessToEC2` and an instance profile ARN of `arn:aws:iam::187424272464:instance-profile/S3FullAccessToEC2`. The role is currently inactive. The **Trust relationships** tab is selected, showing the trust policy for the role. The trust policy is a JSON document that allows the role to assume the `ec2.amazonaws.com` service as a principal.

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Principal": {
7-         "Service": "ec2.amazonaws.com"
8-       },
9-       "Action": "sts:AssumeRole"
10-    }
11-  ]
12- }
```

Modify the IAM role of the instance to our created **S3FullAccessToEC2** role



Created an S3 bucket and added files to it

```
ec2-user@ip-172-31-15-190 ~$ aws --version
aws-cli/2.9.19 Python/3.9.16 Linux/x86_64/amzn2023.x86_64 source/x86_64/amzn.2023 prompt/off
[ec2-user@ip-172-31-15-190 ~]$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [us-east-1]:
Default output format [None]:
[ec2-user@ip-172-31-15-190 ~]$ aws s3 ls
2023-06-05 12:13:13 grandcoffee1234
[ec2-user@ip-172-31-15-190 ~]$ aws s3 mb s3://aravindetagi
make_bucket: aravindetagi
[ec2-user@ip-172-31-15-190 ~]$ aws s3 ls
2023-06-15 17:35:00 aravindetagi
2023-06-05 12:13:13 grandcoffee1234
[ec2-user@ip-172-31-15-190 ~]$ touch file{1..5}.txt
[ec2-user@ip-172-31-15-190 ~]$ ls -l >> data.txt
[ec2-user@ip-172-31-15-190 ~]$ aws s3 cp /home/ec2-user/* s3://aravindetagi
Unknown options: /home/ec2-user/file2.txt,/home/ec2-user/file3.txt,/home/ec2-user/file4.txt,/home/ec2-user/file5.txt,s3://aravindetagi
[ec2-user@ip-172-31-15-190 ~]$ aws s3 cp /home/ec2-user/* s3://aravindetagi
Unknown options: /home/ec2-user/file2.txt,/home/ec2-user/file3.txt,/home/ec2-user/file4.txt,/home/ec2-user/file5.txt,s3://aravindetagi
[ec2-user@ip-172-31-15-190 ~]$ aws s3 cp /home/ec2-user s3://aravindetagi --recursive
upload: aws/config to s3://aravindetagi/aws/config
upload: ./file2.txt to s3://aravindetagi/file2.txt
upload: ./file1.txt to s3://aravindetagi/file1.txt
upload: ./file3.txt to s3://aravindetagi/file3.txt
upload: ./file4.txt to s3://aravindetagi/file4.txt
upload: ./bash_profile to s3://aravindetagi/.bash_profile
upload: ./file5.txt to s3://aravindetagi/file5.txt
upload: ./bash_logout to s3://aravindetagi/.bash_logout
upload: .ssh/authorized_keys to s3://aravindetagi/.ssh/authorized_keys
upload: ./data.txt to s3://aravindetagi/data.txt
upload: ./bashrc to s3://aravindetagi/.bashrc
[ec2-user@ip-172-31-15-190 ~]$
[ec2-user@ip-172-31-15-190 ~]$
[ec2-user@ip-172-31-15-190 ~]$ aws s3 ls s3://aravindetagi
PRE .aws/
PRE .ssh/
2023-06-15 17:41:00 18 .bash_logout
2023-06-15 17:41:00 141 .bash_profile
2023-06-15 17:41:00 492 .bashrc
2023-06-15 17:41:00 349 data.txt
2023-06-15 17:41:00 0 file1.txt
2023-06-15 17:41:00 0 file2.txt
2023-06-15 17:41:00 0 file3.txt
2023-06-15 17:41:00 0 file4.txt
2023-06-15 17:41:00 0 file5.txt
[ec2-user@ip-172-31-15-190 ~]$
```

Instances | Billing | IAM | aravindetagi | Creating of | Automated | (388) Class | WhatsApp | (388) AWS | Channel con | +

s3.console.aws.amazon.com/s3/buckets/aravindetagi?region=us-east-1&tab=objects

Amazon S3 > Buckets > aravindetagi

aravindetagi Info

Objects Properties Permissions Metrics Management Access Points

Objects (11)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix Show versions

	Name	Type	Last modified	Size	Storage class
	.aws/	Folder	-	-	-
	.bash_logout	bash_logout	June 15, 2023, 23:11:00 (UTC+05:30)	18.0 B	Standard
	.bash_profile	bash_profile	June 15, 2023, 23:11:00 (UTC+05:30)	141.0 B	Standard
	.bashrc	bashrc	June 15, 2023, 23:11:00 (UTC+05:30)	492.0 B	Standard
	.ssh/	Folder	-	-	-
	data.txt	txt	June 15, 2023, 23:11:00 (UTC+05:30)	349.0 B	Standard
	file1.txt	txt	June 15, 2023, 23:11:00 (UTC+05:30)	0 B	Standard
	file2.txt	txt	June 15, 2023, 23:11:00 (UTC+05:30)	0 B	Standard
	file3.txt	txt	June 15, 2023, 23:11:00 (UTC+05:30)	0 B	Standard

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Configured the lifecycle through aws cli as below

```
ec2-user@ip-172-31-15-190 ~]$ aws --version
aws-cli/2.9.19 Python/3.9.16 Linux/6.1.29-50.88.amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off
ec2-user@ip-172-31-15-190 ~]$ aws s3 ls
2023-06-15 17:35:00 aravindetagi
2023-06-05 12:13:13 grandcoffee1234
ec2-user@ip-172-31-15-190 ~]$ vi lifecycle.json
ec2-user@ip-172-31-15-190 ~]$ cat lifecycle.json
{
  "Rules": [
    {
      "ID": "MoveToGlacierAfterOneDay",
      "Status": "Enabled",
      "Filter": {
        "Prefix": ""
      },
      "Transitions": [
        {
          "Days": 1,
          "StorageClass": "GLACIER"
        }
      ]
    }
  ]
}
ec2-user@ip-172-31-15-190 ~]$ ls
data.txt  file1.txt  file2.txt  file3.txt  file4.txt  file5.txt  lifecycle.json
ec2-user@ip-172-31-15-190 ~]$ aws s3api put-bucket-lifecycle-configuration --bucket s3://aravindetagi --lifecycle-configuration lifecycle.json
Error parsing parameter '--lifecycle-configuration': Expected: '=', received: 'EOF' for input:
lifecycle.json
ec2-user@ip-172-31-15-190 ~]$ aws s3api put-bucket-lifecycle-configuration --bucket s3://aravindetagi --lifecycle-configuration=lifecycle.json
Error parsing parameter '--lifecycle-configuration': Expected: '=', received: 'EOF' for input:
lifecycle.json
ec2-user@ip-172-31-15-190 ~]$ aws s3api put-bucket-lifecycle-configuration --bucket s3://aravindetagi --lifecycle-configuration file://lifecycle.json
Parameter validation failed:
Invalid bucket name "s3://aravindetagi": Bucket name must match the regex "^[a-zA-Z0-9-._]{1,255}$" or be an ARN matching the regex "^arn:(aws).*:(s3|s3-object-lambda):[a-z-0-9]{0-9}[12]
j:accesspoint[/]?[a-zA-Z0-9-._]{1,63}$|^arn:(aws).*:s3-outposts:[a-z-0-9]{0-9}[12]:outpost[/]?[a-zA-Z0-9-._]{1,63}$|^arn:(aws).*:s3-accesspoint:[a-z-0-9]{0-9}[12]:accesspoint[/]?[a-zA-Z0-9-._]{1,63}$"
ec2-user@ip-172-31-15-190 ~]$ aws s3api put-bucket-lifecycle-configuration --bucket aravindetagi --lifecycle-configuration file://lifecycle.json
ec2-user@ip-172-31-15-190 ~]$
```

1. Created a json to configure lifecycle of S3 bucket

2. Added the lifecycle to bucket as stated in the json file

Amazon S3 > Buckets > aravindetagi > Lifecycle configuration

Lifecycle configuration [Info](#)

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

Lifecycle rules (1)

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. [Learn more](#)

[Refresh](#) [View details](#) [Edit](#) [Delete](#) [Actions](#) [Create lifecycle rule](#)

Lifecycle rule name	Status	Scope	Current version actions	Noncurrent versions actions	Expired object delete markers	Incomplete multipart uploads
<input type="radio"/> MoveToGlacierAfterOneDay	Enabled	Entire bucket	Transition to Glacier Flexible Retrieval (formerly Glacier)	-	-	-

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Instances | EC2 Manag...

S3 Management Cons...

Creating object key na...

Automated EC2 Instanc...

(388) Class 9 -AWS EB...

(388) AWS CLI on Ami...

Channel content - You...

s3.console.aws.amazon.com/s3/management/aravindetagi/lifecycle/view?region=us-east-1&id=MoveToGlacierAfterOneDay

awsServicesSearch[Alt+S]

Globalaravindetagi

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

MoveToGlacierAfterOneDayinfo

EditDeleteActions

Lifecycle rule configuration

Lifecycle rule name	Prefix	Minimum object size
MoveToGlacierAfterOneDay	-	-
Status	Object tags	Maximum object size
Enabled	-	-
Scope		
Entire bucket		

Review transition and expiration actions

Current version actions

Day 0

- Objects uploaded

↓

Day 1

- Objects move to Glacier Flexible Retrieval (formerly Glacier)

Noncurrent versions actions

Day 0

No actions defined.

CloudShellFeedbackLanguage

© 2023, Amazon Web Services India Private Limited or its affiliates. PrivacyTermsCookie preferences

ENG12:50 PMUS16/06/2023