

Experiment 5

Automation and Optimization with Amazon S3

Name: Aditya Singh

Reg

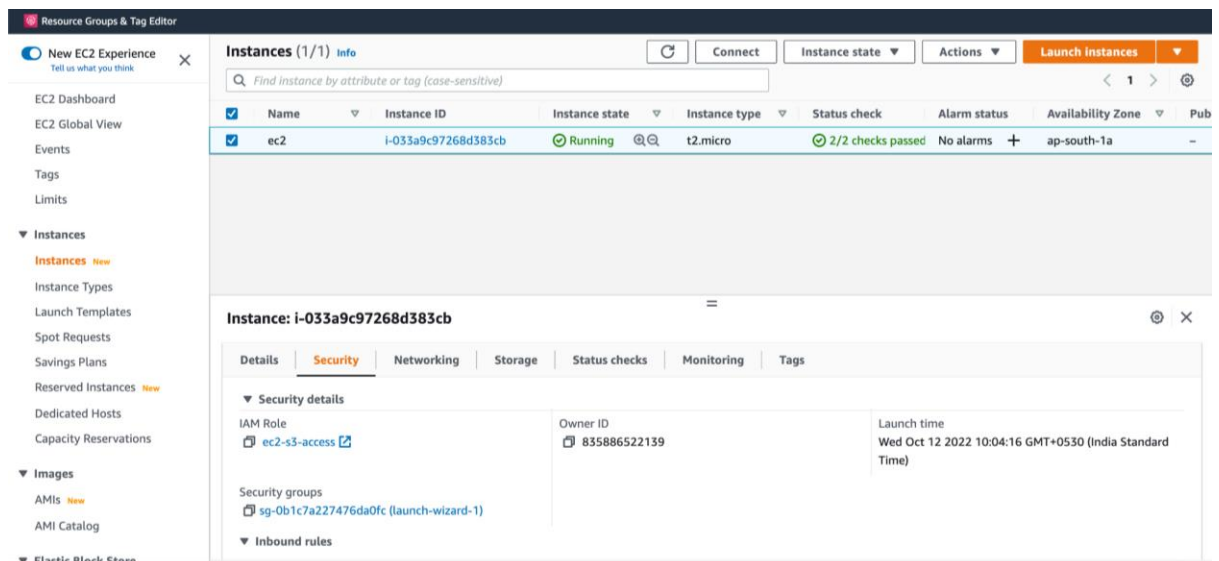
No:RA2011028010089 **Aim:** Automate Files backup to AWS

S3 bucket on Linux machine.

Procedure: -

Steps:

1. Create a S3 bucket.
2. Create a EC2 instance.
3. Give EC2 instance Role to access S3.



(Or you may also grant access to your local Linux machine using aws configure cmd and entering your IAM user credentials over there)

1. Connect to your EC2 instance CLI.
2. Type “sudo su” to give access root directory.
3. Create a directory “backup”. Type: mkdir backup
4. Go inside the “backup” directory.
5. Make some test files. Type: touch a

```
The user-provided path /root/backup does not exist.
[root@ip-172-31-0-253 backup]# aws s3 sync /backup s3://automate-uploadd

The user-provided path /backup does not exist.
[root@ip-172-31-0-253 backup]# aws s3 /backup s3://automate-uploadd
Note: AWS CLI version 2, the latest major version of the AWS CLI, is now stable and recommended for general use. For more information, see the AWS CLI version
2 installation instructions at: https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html

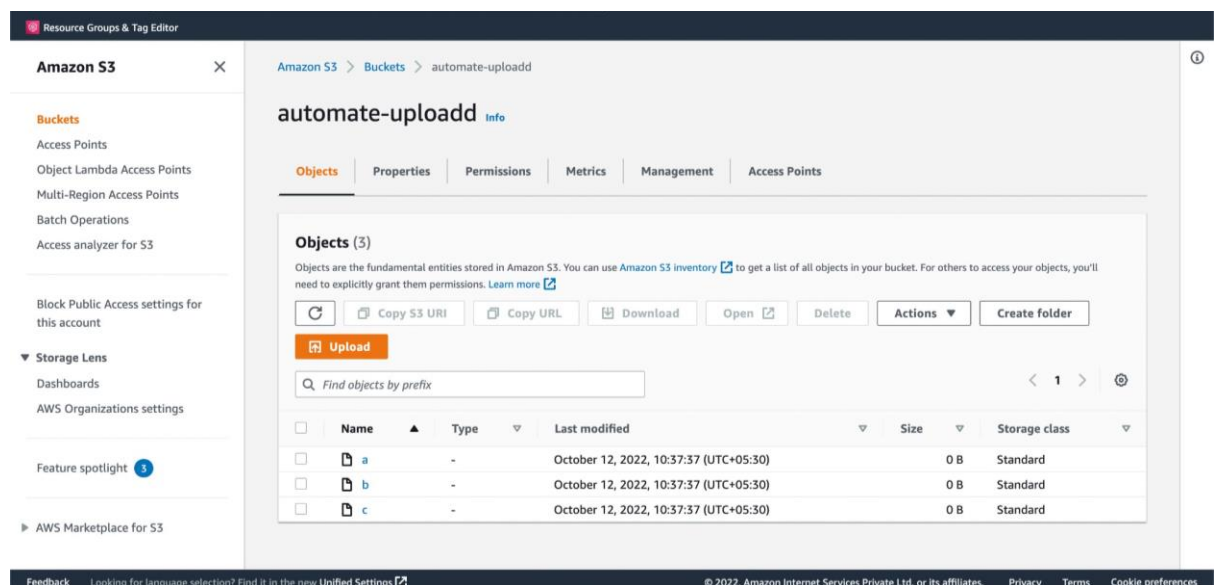
usage: aws [options] <command> [<subcommand>] [<subcommand> ...] [parameters]
To see help text, you can run:

    aws help
    aws <command> help
    aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                                | website
cp                                | mv
rm                                | sync
mb                                | rb
presign

[root@ip-172-31-0-253 backup]# pwd
/home/ec2-user/backup
[root@ip-172-31-0-253 backup]# aws s3 sync /home/ec2-user/backup s3://automate-uploadd/a
upload: ./a to s3://automate-uploadd/a
upload: ./c to s3://automate-uploadd/c
upload: ./b to s3://automate-uploadd/b
[root@ip-172-31-0-253 backup]#
```

9. List Them by Cmd-ls



The screenshot shows the Amazon S3 console interface. On the left is a sidebar with navigation options like Buckets, Access Points, and Storage Lens. The main area displays the 'automate-uploadd' bucket. Under the 'Objects' tab, there is a table listing three objects: 'a', 'b', and 'c'. Each object is 0 B in size and stored in the Standard storage class. The last modified date for all objects is October 12, 2022, at 10:37:37 (UTC+05:30).

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	a	-	October 12, 2022, 10:37:37 (UTC+05:30)	0 B	Standard
<input type="checkbox"/>	b	-	October 12, 2022, 10:37:37 (UTC+05:30)	0 B	Standard
<input type="checkbox"/>	c	-	October 12, 2022, 10:37:37 (UTC+05:30)	0 B	Standard

Now to sync these files of backup directory on the S3 bucket.

Cmd : `aws s3 sync local file path s3://bucketname`

11. Now, we are going to create a cron job in order to automate this process. Cmd: `crontab -e`

Enter the cmd: cron code `aws s3 sync /directory s3://bucketname`

For e.g.: cron code for 1 min is `* * * * *`

(you may use crontab.guru to create your own job expression)

URL :

<https://crontab.guru/>

```
usage: aws [options] <command> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                               | website
cp                               | mv
rm                               | sync
mb                               | rb
presign

[root@ip-172-31-0-253 backup]# pwd
/home/ec2-user/backup
[root@ip-172-31-0-253 backup]# aws s3 sync /home/ec2-user/backup s3://automate-upload
upload: ./a to s3://automate-upload/a
upload: ./c to s3://automate-upload/c
upload: ./b to s3://automate-upload/b
[root@ip-172-31-0-253 backup]# crontab -e
no crontab for root - using an empty one

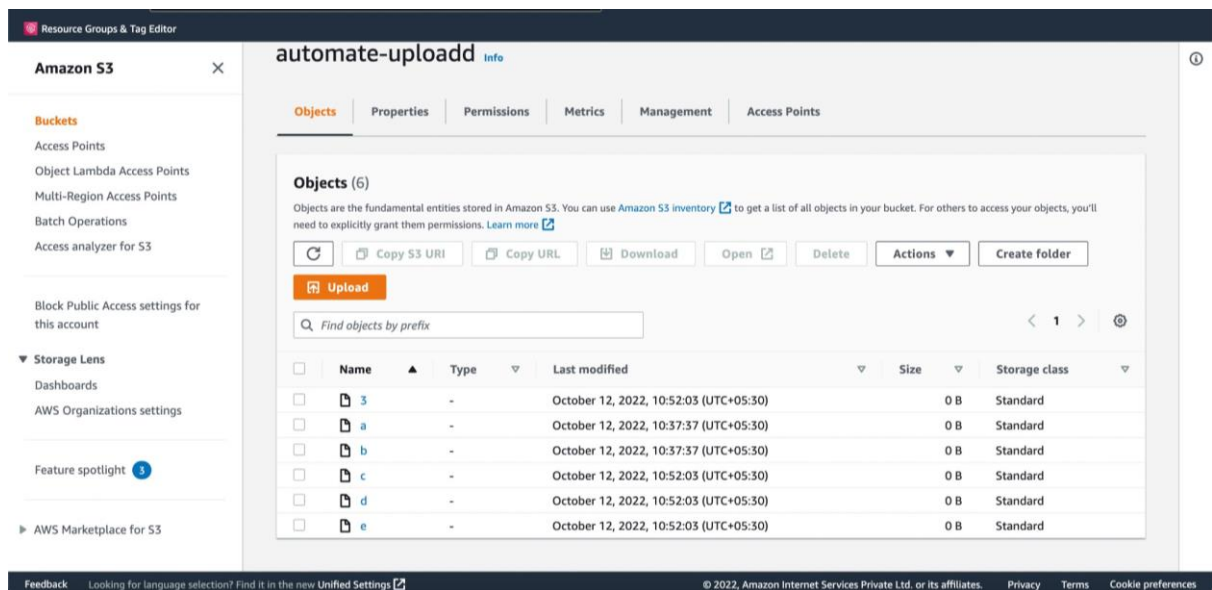
[]+ Stopped crontab -e
[root@ip-172-31-0-253 backup]# crontab -e
bash: crontab: command not found
[root@ip-172-31-0-253 backup]# crontab -e
bash: crontab: command not found
[root@ip-172-31-0-253 backup]#
```

Restart the Crond service

Run “`systemctl restart/stop/start cornd.service`” to restart/stop/start your cron jobs respectively.

12. Now, we are going to create some test files to check if they are uploaded every minute or not.

13. File d and file e have been updated.



Result:

We have successfully automated our local files/directory backup on Amazon S3 buckets using crontab.

