AARHEE PHUKAN(RA2011028010092)

Experiment : 5

Title : Automation and Optimization with Amazon S3

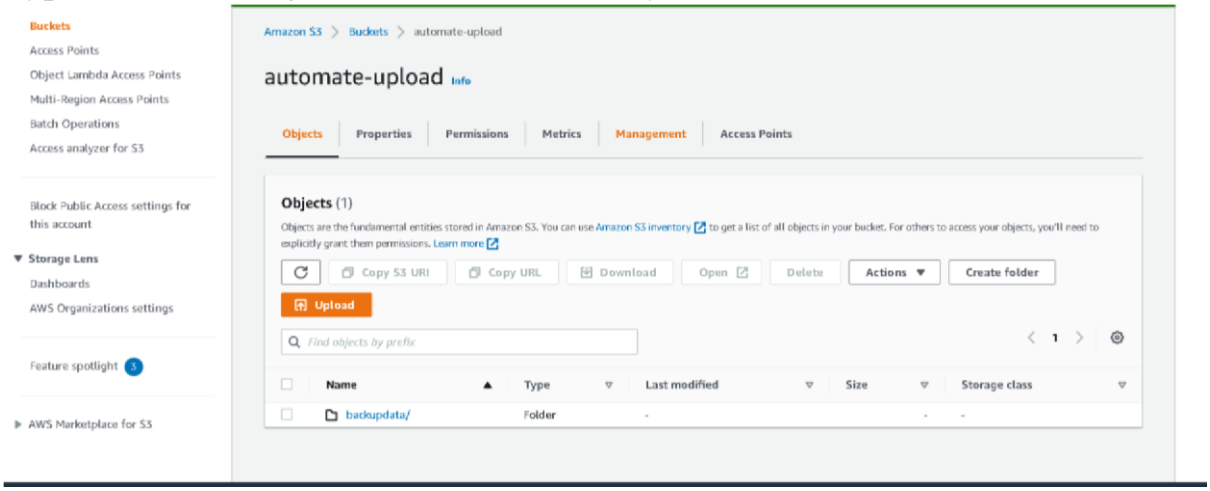
Aim : Automate Files backup to aws S3 bucket on Linux machine.

Pre-requisites : AWS Console, Amazon S3, crontab, aws cli

Procedure :

Steps:

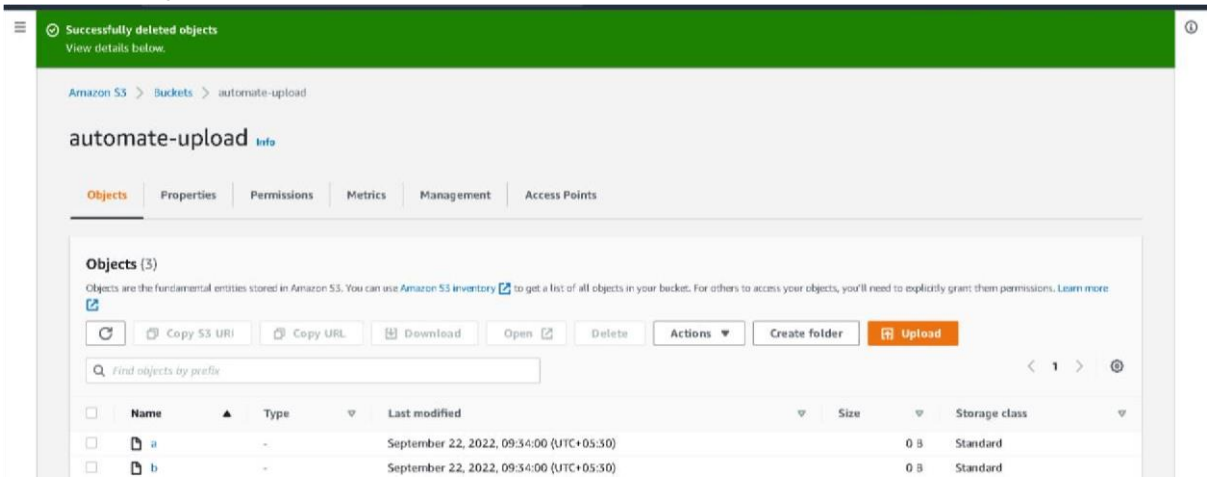
1. Create a S3 bucket.
2. Create a EC2 instance.
3. Give EC2 instance Role to access S3.
4. Connect to your EC2 instance CLI.
5. Type “sudo su” to give access root directory.



6. Create a directory “backup”. Type: mkdir backup
7. Go inside the “backup” directory.
8. Make some test files. Type : touch a

```
2022-09-19 08:27:17 paint-ad1543
[root@ip-172-31-32-239 ec2-user]# aws s3 ls automate-upload
                           PRE backupdata/
[root@ip-172-31-32-239 ec2-user]# mkdir backup
[root@ip-172-31-32-239 ec2-user]# cd backup
[root@ip-172-31-32-239 backup]# touch a
[root@ip-172-31-32-239 backup]# touch b
[root@ip-172-31-32-239 backup]# touch c
[root@ip-172-31-32-239 backup]# ls
a  b  c
[root@ip-172-31-32-239 backup]# aws s3 sync /root/backup s3://automate-upload

The user-provided path /root/backup does not exist.
[root@ip-172-31-32-239 backup]# aws s3 /backup s3://automate-upload
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-172-31-32-239 backup]# pwd
/home/ec2-user/backup
[root@ip-172-31-32-239 backup]# aws s3 sync /home/ec2-user/backup s3://automate-upload
upload: ./c to s3://automate-upload/c
upload: ./b to s3://automate-upload/b
upload: ./a to s3://automate-upload/a
[root@ip-172-31-32-239 backup]#
```

9. List them by cmd – ls



10. Now to sync these files of backup directory on the S3 bucket. Cmd : aws s3 sync localfilepath 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/

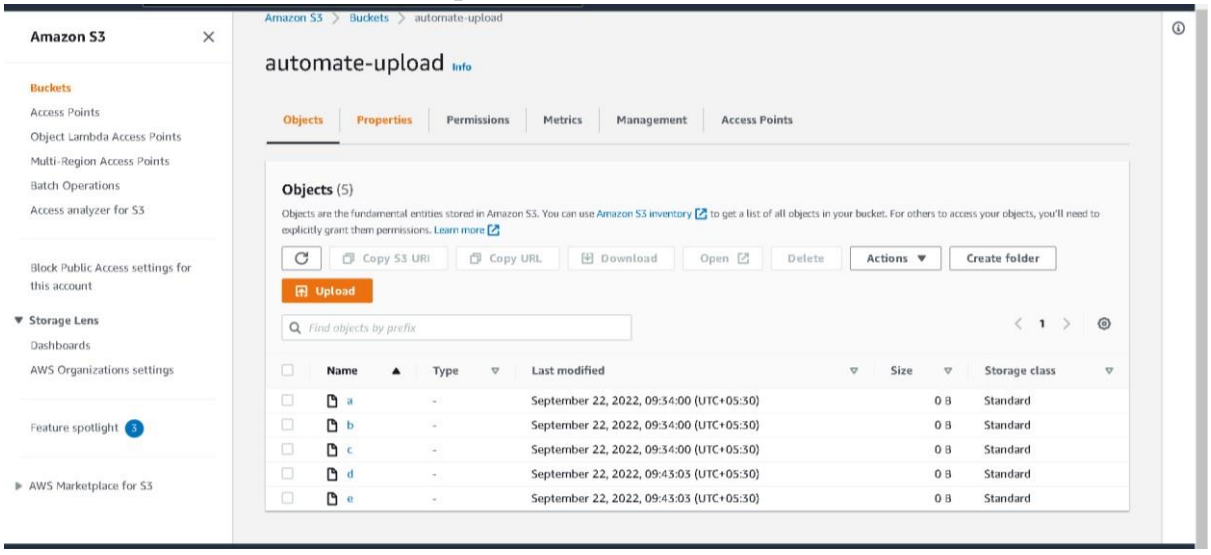
```
* * * * * aws s3 sync /home/ec2-user/backup s3://automate-upload
```

```
[root@ip-172-31-32-239 backup]# touch a
[root@ip-172-31-32-239 backup]# touch b
[root@ip-172-31-32-239 backup]# touch c
[root@ip-172-31-32-239 backup]# ls
a  b  c
[root@ip-172-31-32-239 backup]# aws s3 sync /root/backup s3://automate-upload

The user-provided path /root/backup does not exist.
[root@ip-172-31-32-239 backup]# aws s3 /backup s3://automate-upload
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-172-31-32-239 backup]# pwd
/home/ec2-user/backup
[root@ip-172-31-32-239 backup]# aws s3 sync /home/ec2-user/backup s3://automate-upload
upload: ./c to s3://automate-upload/c
upload: ./b to s3://automate-upload/b
upload: ./a to s3://automate-upload/a
[root@ip-172-31-32-239 backup]#
[root@ip-172-31-32-239 backup]#
[root@ip-172-31-32-239 backup]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
[root@ip-172-31-32-239 backup]#
```

12. Restart the Crond service

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

13. Now, we are going to create some test files to check if they are uploaded every minute or not.
14. File d and file e have been updated.



Result:

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