

Experiment 5

Automation and Optimization with Amazon S3

Name : Pranav Yadav

Reg No : RA2011028010079

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.

The screenshot displays the AWS Management Console interface for the 'Resource Groups & Tag Editor'. The left sidebar shows the navigation menu with options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', and 'Instances'. The main content area shows a table of EC2 instances. One instance, 'atulec2', is listed with the following details:

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Public DNS (IPv4) |
|---------|---------------------|---------------|-------------------|----------------|---------------|--------------|---|
| atulec2 | i-0d832705aeb39898f | t2.micro | us-east-1d | running | Initializing | None | ec2-34-228-69-177.compute-1.amazonaws.com |

Below the table, the details for the selected instance 'atulec2' are shown, including its Public DNS: 'ec2-34-228-69-177.compute-1.amazonaws.com'. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 11:40 PM on 11/10/2022.

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

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

```
sh-3.2# mkdir backup
sh-3.2# cd backup
sh-3.2# touch a
sh-3.2# touch b
sh-3.2# touch c
sh-3.2# ls
a      b      c
sh-3.2# aws s3 /backup s3:// akils3bucket
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
```

9. List Them By Cmd–ls

The screenshot shows the Amazon S3 console interface. At the top, the breadcrumb navigation reads 'Amazon S3 > Buckets > atulbuckets3'. The main heading is 'atulbuckets3' with an 'Info' link. Below this are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected, showing 'Objects (0)'. A descriptive text states: '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](#)'. Below the text is a row of action buttons: 'Refresh', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'. An 'Upload' button is also present. A search bar with the placeholder 'Find objects by prefix' is located below the buttons. At the bottom, a table header is visible with columns: 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The footer of the console shows '© 2022, Amazon Internet Services Private Ltd. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'. The Windows taskbar at the very bottom shows the date as 11/10/2022 and the time as 11:43 PM.

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.gu>

```
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
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]# crontab -e
no crontab for root - using an empty one

[1]+  Stopped                  crontab -e
[root@ip-172-31-0-253 backup]# cron code aws s3 sync /home/ec2-user/backup s3://automate-uploadd
bash: cron: command not found
[root@ip-172-31-0-253 backup]# cron code aws s3 sync /backup s3://automate-uploadd
bash: cron: 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.

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.

The screenshot shows the AWS Management Console interface for an Amazon S3 bucket named 'automate-upload'. The left sidebar displays the 'Amazon S3' service with various options like Buckets, Access Points, and Storage Lens. The main content area shows the 'Objects (6)' tab, which lists six objects: '3', 'a', 'b', 'c', 'd', and 'e'. Each object is 0 B in size and uses the 'Standard' storage class. The 'Last modified' column shows the date and time for each object. The interface includes a search bar, a list of actions (Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder), and an 'Upload' button.

| Name | Type | Last modified | Size | Storage class |
|------|------|--|------|---------------|
| 3 | - | October 12, 2022, 10:52:03 (UTC+05:30) | 0 B | Standard |
| a | - | October 12, 2022, 10:37:37 (UTC+05:30) | 0 B | Standard |
| b | - | October 12, 2022, 10:37:37 (UTC+05:30) | 0 B | Standard |
| c | - | October 12, 2022, 10:52:03 (UTC+05:30) | 0 B | Standard |
| d | - | October 12, 2022, 10:52:03 (UTC+05:30) | 0 B | Standard |
| e | - | October 12, 2022, 10:52:03 (UTC+05:30) | 0 B | Standard |

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