

Experiment 5

Automation and Optimization with Amazon S3

Name : PUSHPITA BORAL

Reg No : RA2011028010066

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. On the left, the navigation menu includes 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', and 'Instances'. The 'Instances' section is expanded, showing a list of instances. One instance, 'pushpita04', is highlighted with a blue row. Below the list, the details for instance 'i-0a1367cb3c8234820' are shown. The instance is in a 'Running' state, using a 't2.micro' instance type. The public IPv4 address is '15.207.99.174' and the private IPv4 address is '172.31.8.196'. The instance state is confirmed as 'Running'.

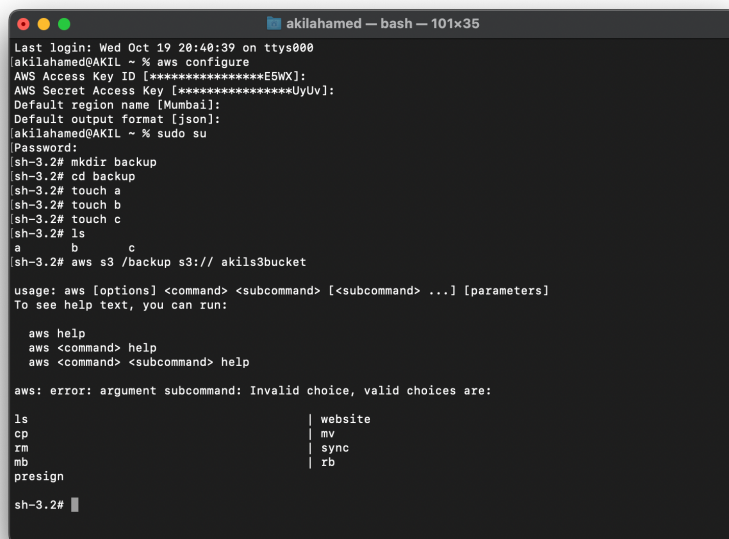
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
pushpita04	i-0a1367cb3c8234820	Running	t2.micro	-	No alarms	ap-south-1

Instance: i-0a1367cb3c8234820		
Details	Security	Networking
Instance summary Instance ID: i-0a1367cb3c8234820 IPv6 address: -	Public IPv4 address: 15.207.99.174 open address Instance state: Running	Private IPv4 addresses: 172.31.8.196 Public IPv4 DNS: ec2-15-207-99-174.ap-south-

(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

A terminal window titled 'akilahamed -- bash -- 101x35'. The terminal shows the following commands and output:

```
Last login: Wed Oct 19 20:40:39 on ttys000
[akilahamed@AKIL ~ % aws configure
AWS Access Key ID [*****E5WX]:
AWS Secret Access Key [*****UyUv]:
Default region name [Mumbai]:
Default output format [json]:
[akilahamed@AKIL ~ % sudo su
Password:
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

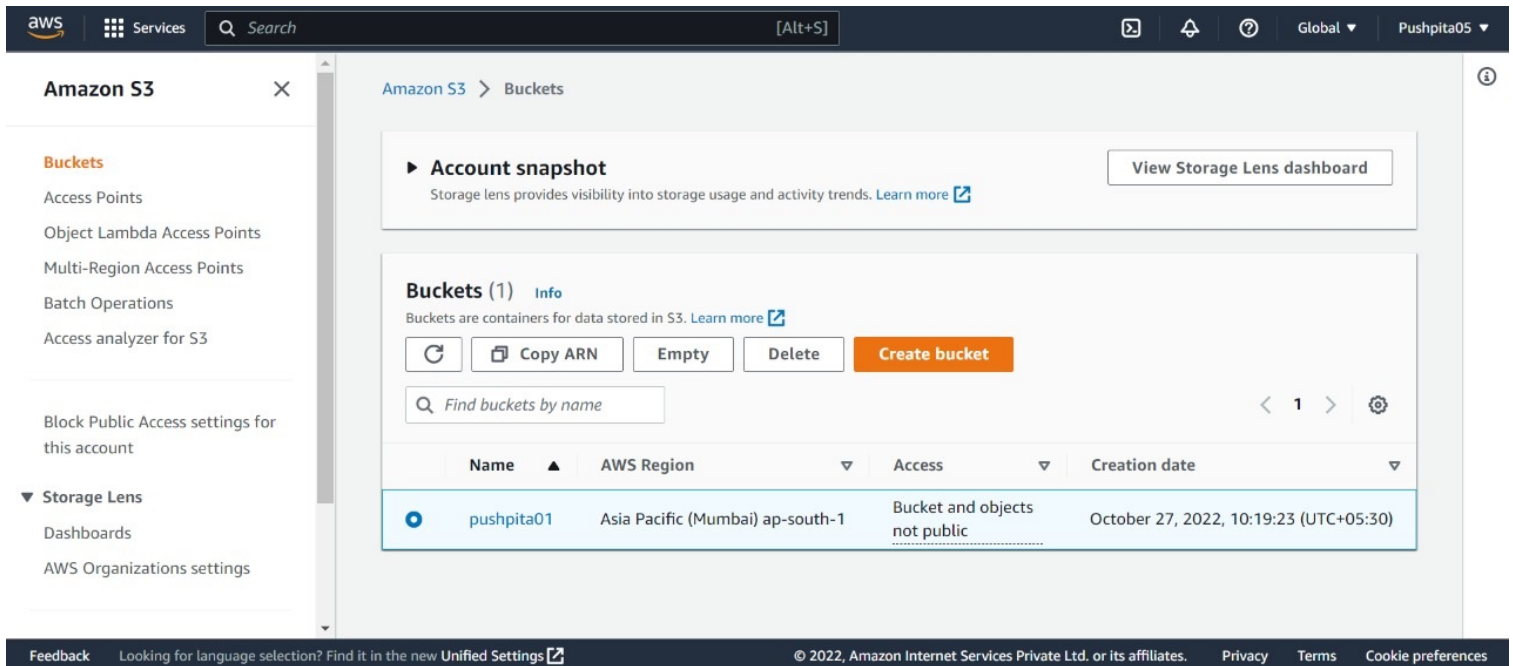
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

    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
sh-3.2#
```

9. List Them By Cmd-ls



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/>

```

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.

aws Services Search for services, features, blogs, docs, and more [Option+S] Global AWSMangloo @ 8358-8652-2139

Resource Groups & Tag Editor

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

automate-upload

Objects Properties Permissions Metrics Management Access Points

Objects (6)

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

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	3	-	October 12, 2022, 10:52:03 (UTC+05:30)	0 B	Standard
<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, 10:52:03 (UTC+05:30)	0 B	Standard
<input type="checkbox"/>	d	-	October 12, 2022, 10:52:03 (UTC+05:30)	0 B	Standard
<input type="checkbox"/>	e	-	October 12, 2022, 10:52:03 (UTC+05:30)	0 B	Standard

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

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