Experiment 5

Automation and Optimization with Amazon S3

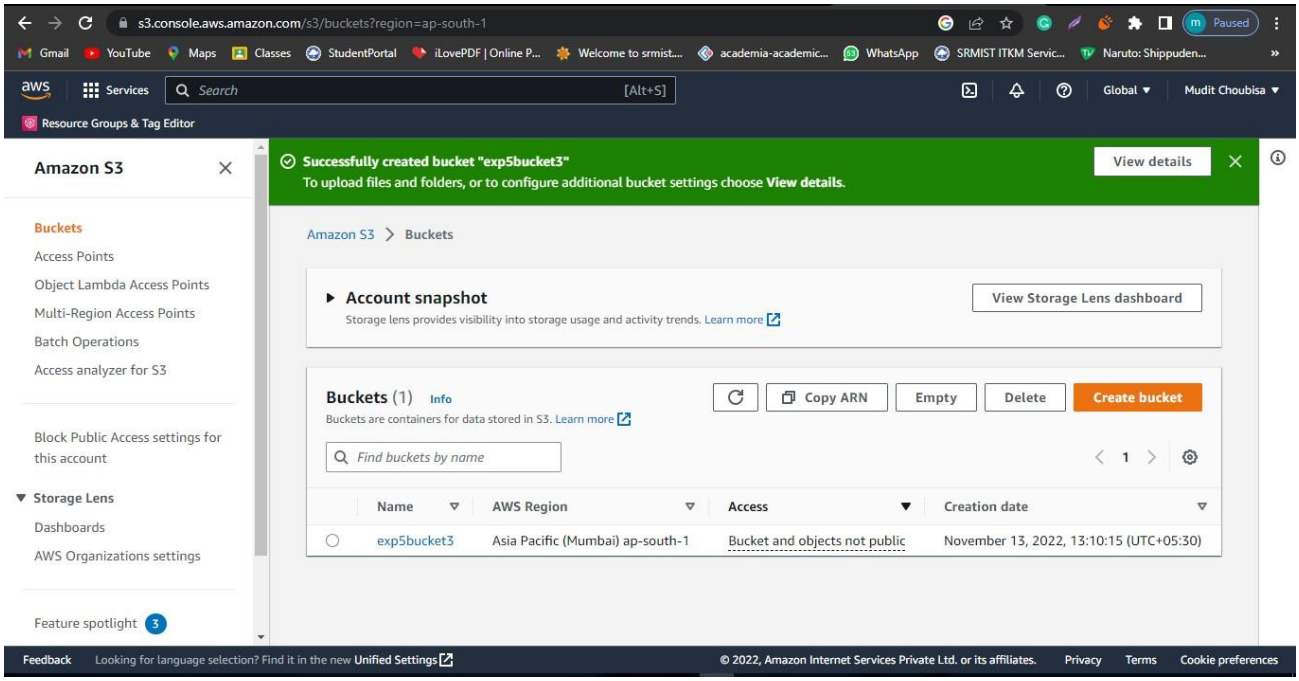
Name :Piyush kumar Reg No : **RA2011028010084**

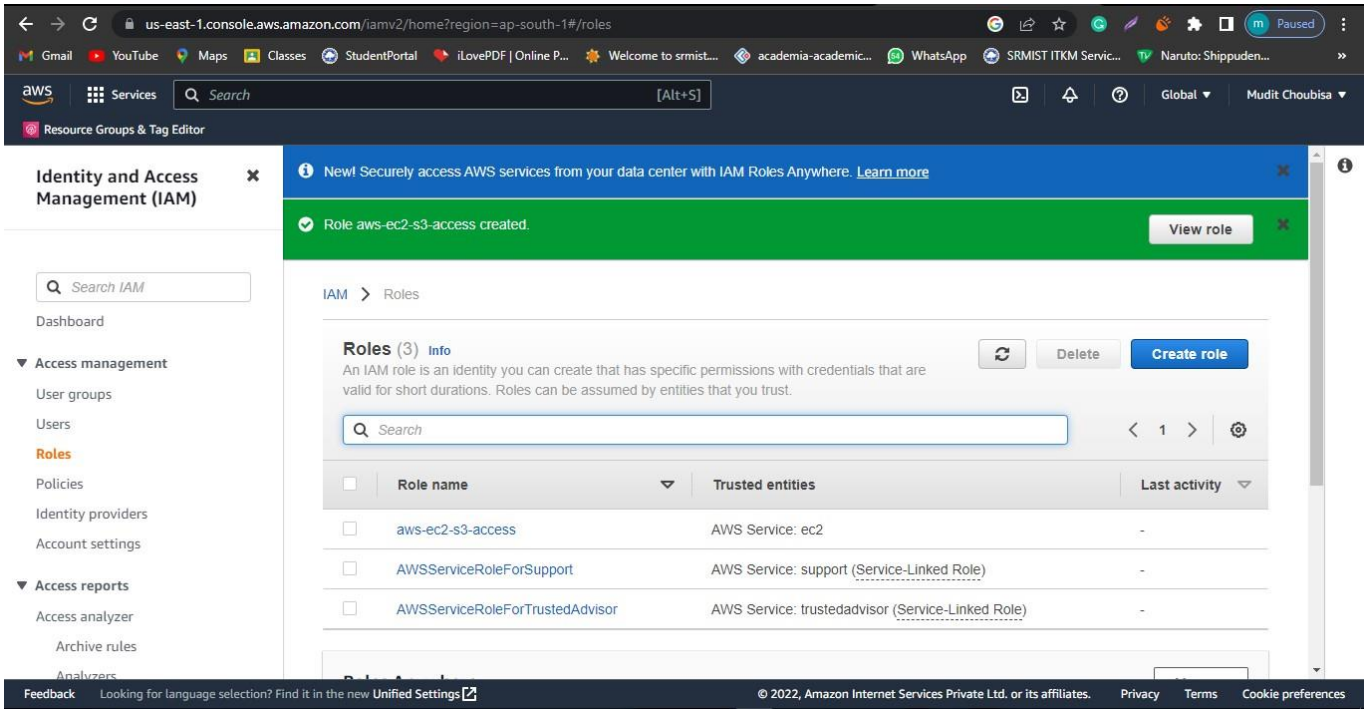
Aim : Automate Files backup to aws S3 bucket on Linux machine.

Procedure :

Steps:

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


s3.console.aws.amazon.com/s3/buckets?region=ap-south-1
Amazon S3
Successfully created bucket "exp5bucket3"
To upload files and folders, or to configure additional bucket settings choose View details.
View details
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (1) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
exp5bucket3
Asia Pacific (Mumbai) ap-south-1
Bucket and objects not public
November 13, 2022, 13:10:15 (UTC+05:30)
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
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


us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/roles
Identity and Access Management (IAM)
New! Securely access AWS services from your data center with IAM Roles Anywhere. Learn more
Role aws-ec2-s3-access created.
View role
IAM > Roles
Roles (3) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Search
Role name
Trusted entities
Last activity
aws-ec2-s3-access
AWS Service: ec2
AWSServiceRoleForSupport
AWS Service: support (Service-Linked Role)
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service-Linked Role)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
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

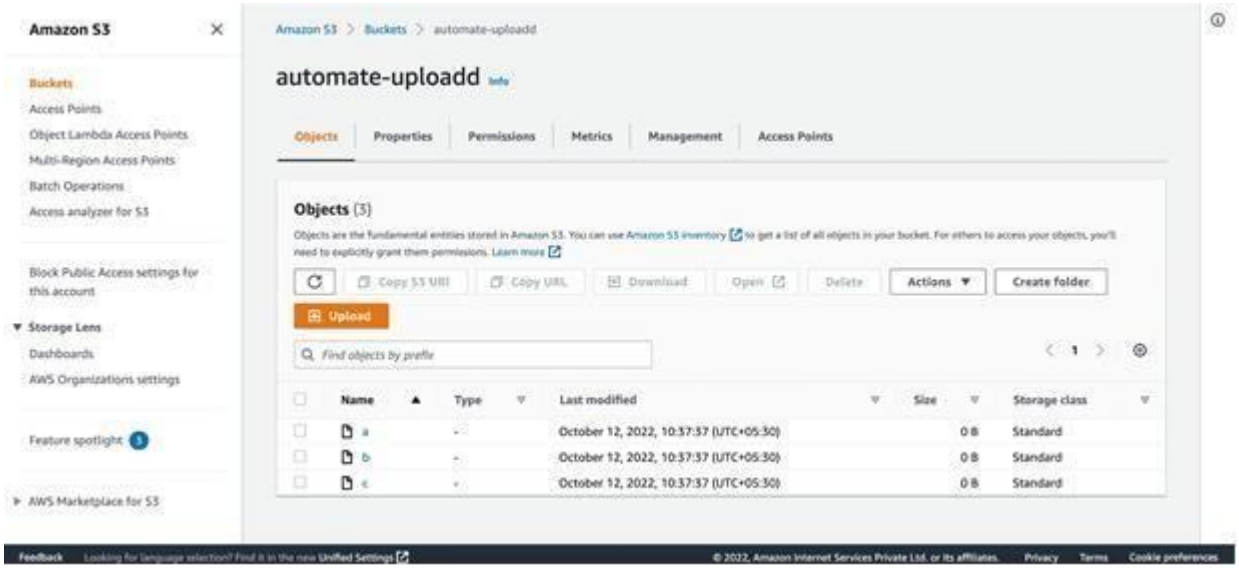
(or you may also grant access to your local linux machine using aws configure cmd and entering your IAM user credentials over there)

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

```
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
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



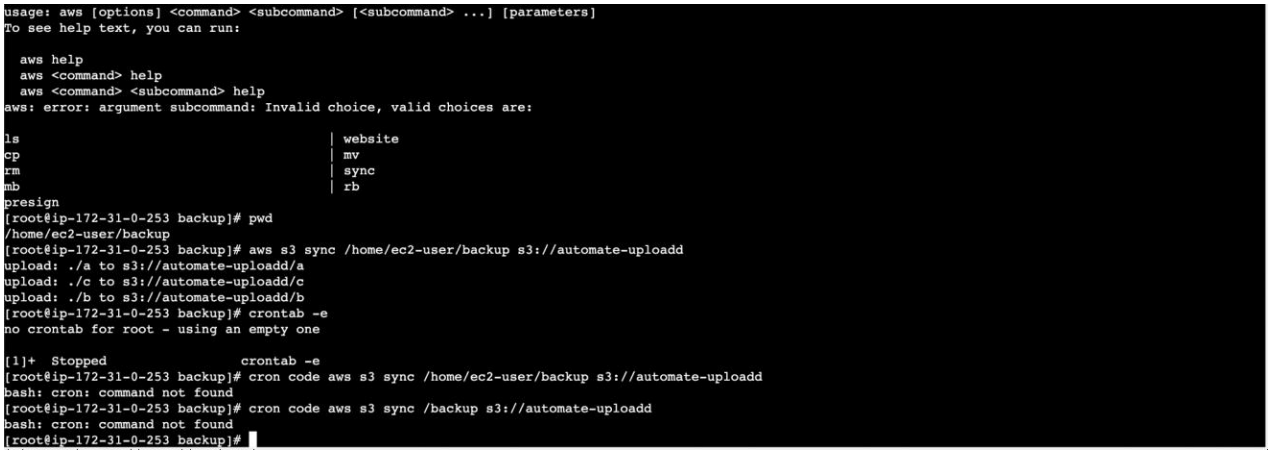
Now to sync these files of backup directory on the S3 bucket. Cmd : aws s3 sync localfilepath s3://bucketname

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/

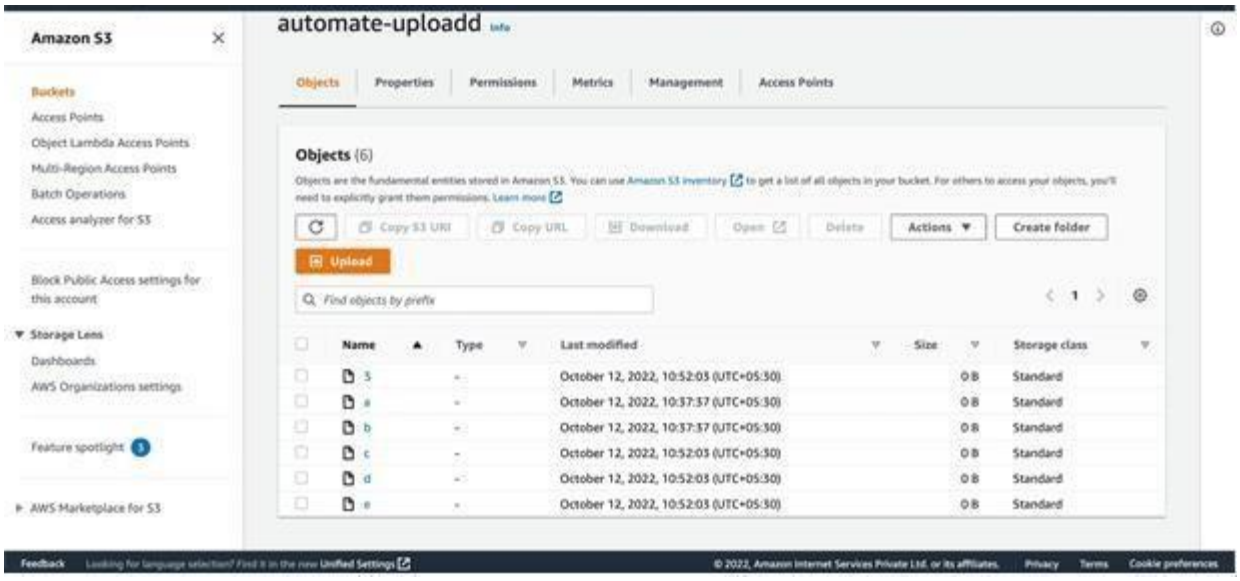


Restart the Crond service

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.