

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

LAB RECORD

Academic Year: 2022-2023

ODD SEMESTER

Programme (UG/PG): UG

Semester: V

Course Code: 18CSC310J

COURSE TITLE: DATA CENTRIC NETWORKING AND SYSTEM DESIGN

STUDENT NAME: Marla Sai Ruthwik

REGISTER NUMBER: RA2011028010124

BRANCH WITH SPECIALIZATION: CSE- CLOUD COMPUTING

SECTION: L1



SCHOOL OF COMPUTING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur- 603203

Chengalpattu District

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

ABOUT THE DEPARTMENT

Department of Networking and Communications under School of Computing intend to meet the expectations of the aspiring students and to add more value to the degrees offered. The Department ensures to provide quality and value-laden education for students in the traditional and contemporary areas of Cloud Computing, Computer Networks, Cyber Security, Information Technology and Internet of Things. The programs are introduced in partnership with reputed IT companies like Amazon Web services, K7 Security, Virtusa etc. The department consists of a medley of faculty members with industrial and academic experience. The Department's keen focus is towards "networks" domain specific and specialization-based placement drives for its students. The department inculcates entrepreneurial skills in budding aspirants to pitch their innovative ideas through SRM Innovation and Incubation Center. Our International and alumni connect intrigue in bridging the gap between the trio: Academics-Industry -Research

Vision

To Nurture as globally recognisable department in imparting the student's high quality education and providing high confidence, unique knowledge and research experience in the field of networking, cyber security, forensics, information technology, cognitive computing and internet of things.

Mission

- To provide world class IT professionals with appropriate industry and research-based curriculum
- To train the students in such a way that leads to entrepreneurship and develop societal need-based industries
- To nourish the students as a socially responsible professionals by providing them training in personality development, ethics and leadership program

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

**SRM Nagar, Kattankulathur - 603203
Chengalpattu District**



Name- Marla Sai Ruthwik

Year- 3rd

Semester-5th

Branch- CSE-CLOUD COMPUTING

University Register No.: RA2011028010124

Certified that this is a Bonafide Record work done by the above student in the year 2021 - 2022

Signature of Lab. Incharge.

Signature of Head of the Dep

DATE:_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

CONTENTS

Expt. no	Date	Name of the Experiment	Page. no	Marks	Signature
1	25/8/22	Install and Practice Using the AWS CLI.	5		
2	7/9/22	Creating Amazon EC2 Instances - Creating a LAMP Instance in the AWS CLI.	9		
3	14/9/22	Configuring Virtual Private Cloud VPC & Troubleshoot a VPC	13		
4	20/9/22	Using AWS Systems Manager - Create a Static Website on Amazon S3	17		
5	27/9/22	Automation and Optimization with Amazon S3	23		
6	13/10/22	Querying Data in S3 with Amazon Athena	27		
7	19/10/22	Create an AWS Lambda Function to Email Daily Reports	34		
8	2/11/22	Migrate to Amazon RDS	37		
9	9/11/22	Configure Failover Routing with Amazon Route 53	44		
10	10/11/22	Optimize AWS Resource Utilization	49		

Experiment-1

Install and learn using AWS CLI

1] AWS Help

The built-in AWS CLI help command. You can get help with any command when using the AWS Command Line Interface (AWS CLI). To do so, simply type help at the end of a command name. For example, the following command displays help for the general AWS CLI options and the available top-level commands.

```
C:\Users\Admin>aws help

aws
^^^

Description
*****
The AWS Command Line Interface is a unified tool to manage your AWS services.

Synopsis
*****
aws [options] <command> <subcommand> [parameters]

Use *aws command help* for information on a specific command. Use *aws help topics* to view a list of available help topics. The synopsis for each command shows its parameters and their usage. Optional parameters are shown in square brackets.

Options
*****
"--debug" (boolean)

Turn on debug logging.

"--endpoint-url" (string)

Override command's default URL with the given URL.

"--no-verify-ssl" (boolean)

By default, the AWS CLI uses SSL when communicating with AWS services.
For each SSL connection, the AWS CLI will verify SSL certificates
```

2] AWS – version

The AWS CLI version 2 is the most recent major version of the AWS CLI and supports all of the latest features

```
Microsoft Windows [Version 10.0.22622.575]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>aws --version
aws-cli/2.7.25 Python/3.9.11 Windows/10 exe/AMD64 prompt/off

C:\Users\Admin>
```

3] AWS configure

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations.

```
C:\Users\Admin>aws configure
AWS Access Key ID [None]: AKIA2TKXESQCEU2VFL4K
AWS Secret Access Key [None]: 6JXWEoKLXf/SamzNOZQzCEbZKyE2MG4o/2R8HQE0
Default region name [None]:
Default output format [None]:
```

sts get-caller-identity

To get your account id using AWS CLI, run the sts get-caller-identity command, setting the --query parameter to Account to filter the output. Copied! The get-caller-identity command returns the User Id, Account Id, and the ARN of the caller

```
C:\Users\Admin>aws sts get-caller-identity
{
    "UserId": "728716579844",
    "Account": "728716579844",
    "Arn": "arn:aws:iam::728716579844:root"
}
```

5] aws s3 ls

To list your buckets, folders, or objects, use the s3 ls command. Using the command without a target or options lists all buckets.

```
C:\Users\Admin>aws s3 ls
2022-08-22 21:53:20 my-new-test-bucket102
2022-08-22 22:00:23 my-new-test-bucket103
```

6] aws s3 ls bucketName

The following ls command lists objects and common prefixes under a specified bucket and prefix. In this example, the user owns the bucket mybucket with the objects test.txt and somePrefix/test.txt. The LastWriteTime and Length are arbitrary. Note that since the ls command has no interaction with the local filesystem, the s3:// URI scheme is not required to resolve ambiguity and may be omitted

```
C:\Users\Admin>aws s3 ls my-new-test-bucket103
2022-08-22 22:10:11      6000446 btech-curricula-reg-2018-vol-III.pdf
```

7] create bucket

To create a bucket, you must register with Amazon S3 and have a valid Amazon Web Services Access Key ID to authenticate requests. Anonymous requests are never allowed to create buckets. By creating the bucket, you become the bucket owner. Not every string is an acceptable bucket name.

```
C:\Users\Admin>aws s3api create-bucket --bucket my-new-test-bucket101 --region us-east-1
{
    "Location": "/my-new-test-bucket101"
}
```

8] copy bucket

To download an entire bucket to your local file system, use the AWS CLI sync command, passing it the s3 bucket as a source and a directory on your file system as a destination, e.g. aws s3 sync s3://YOUR_BUCKET ..

```
C:\Users\Admin>aws s3 cp s3://my-new-test-bucket102/btech-curricula-reg-2018-vol-III.pdf s3://my-new-test-bucket101/
copy: s3://my-new-test-bucket102/btech-curricula-reg-2018-vol-III.pdf to s3://my-new-test-bucket101/btech-curricula-reg-2018-vol-III.pdf

C:\Users\Admin>
```

9] delete bucket

If your bucket does not have versioning enabled, you can use the rb (remove bucket) AWS CLI command with the --force parameter to delete the bucket and all the objects in it. This command deletes all objects first and then deletes the bucket.

```
C:\Users\Admin>aws s3 rb s3://my-new-test-bucket103 --force
delete: s3://my-new-test-bucket103/btech-curricula-reg-2018-vol-III.pdf
remove_bucket: my-new-test-bucket103
```

10] remove file from bucket

To delete objects in a bucket or your local directory, use the s3 rm command. For a few common options to use with this command, and examples, see Frequently used options for s3 commands. For a complete list of options, see s3 rm in the AWS CLI Command Reference. The following example deletes filename

```
C:\Users\Admin>aws s3 rm s3://my-new-test-bucket102 --recursive
delete: s3://my-new-test-bucket102/btech-curricula-reg-2018-vol-III.pdf
```

Experiment 2

Creating Amazon EC2 Instances - Creating a LAMP Instance in the AWS CLI

AIM: To create a LAMP instance in the AWS CLI.

PROCEDURE:

1. Firstly, type sudo su to become the root user.
2. To update all the packages in your instance type “yum update -y”.

```
root@ip-172-31-32-239:/home/ec2-user
Using username "ec2-user".
Authenticating with public key "imported-ssh-key"
Last login: Sat Aug 27 04:29:33 2022 from 106.208.16.252
[ec2-user@ip-172-31-32-239 ~]$
[ec2-user@ip-172-31-32-239 ~]$
[ec2-user@ip-172-31-32-239 ~]$
[ec2-user@ip-172-31-32-239 ~]$ sudo su
[root@ip-172-31-32-239 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                         | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package chrony.x86_64 0:4.0-0.3.amzn2.0.2 will be updated
--> Package chrony.x86_64 0:4.2-5.amzn2.0.2 will be an update
--> Package dhclient.x86_64 12:4.2.5-77.amzn2.1.6 will be updated
--> Package dhclient.x86_64 12:4.2.5-79.amzn2.1.1 will be an update
--> Package dhcp-common.x86_64 12:4.2.5-77.amzn2.1.6 will be updated
--> Package dhcp-common.x86_64 12:4.2.5-79.amzn2.1.1 will be an update
--> Package dhcp-libs.x86_64 12:4.2.5-77.amzn2.1.6 will be updated
--> Package dhcp-libs.x86_64 12:4.2.5-79.amzn2.1.1 will be an update
--> Package gnupg2.x86_64 0:2.0.22-5.amzn2.0.4 will be updated
--> Package gnupg2.x86_64 0:2.0.22-5.amzn2.0.5 will be an update
--> Package kernel.x86_64 0:5.10.135-122.509.amzn2 will be installed
--> Package kernel-tools.x86_64 0:5.10.130-118.517.amzn2 will be updated
--> Package kernel-tools.x86_64 0:5.10.135-122.509.amzn2 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package          Arch      Version       Repository      Size
=====
Installing:
kernel          x86_64    5.10.135-122.509.amzn2   amzn2extra-kernel-5.10   32 M
Updating:
chrony          x86_64    4.2-5.amzn2.0.2      amzn2-core           302 k
dhclient        x86_64    12:4.2.5-79.amzn2.1.1  amzn2-core           287 k
dhcp-common     x86_64    12:4.2.5-79.amzn2.1.1  amzn2-core           177 k
dhcp-libs       x86_64    12:4.2.5-79.amzn2.1.1  amzn2-core           132 k
gnupg2          x86_64    2.0.22-5.amzn2.0.5   amzn2-core           1.5 M
kernel-tools    x86_64    5.10.135-122.509.amzn2  amzn2extra-kernel-5.10  176 k
```

3. To install Apache server in linux, type “yum install httpd”.

```
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0() (64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0() (64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.54-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.54-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch   Version        Repository      Size
=====
```

4. To install mysql or mariadb type “yum install mariadb mariadb-server”.

```
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]# yum install mariadb mariadb-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2 will be installed
--> Package mariadb-server.x86_64 1:5.5.68-1.amzn2 will be installed
--> Processing Dependency: perl-DBI for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl-DBD-MySQL for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl(Data::Dumper) for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl(DBI) for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Running transaction check
--> Package perl-DBD-MySQL.x86_64 0:4.023-6.amzn2 will be installed
--> Package perl-DBI.x86_64 0:1.627-4.amzn2.0.2 will be installed
--> Processing Dependency: perl(RPC::PlServer) >= 0.2001 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Processing Dependency: perl(RPC::PlClient) >= 0.2000 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Package perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-PlRPC.noarch 0:0.2020-14.amzn2 will be installed
--> Processing Dependency: perl(Net::Daemon) >= 0.13 for package: perl-PlRPC-0.2020-14.amzn2.noarch
```

5. To install php, type “yum install php php-mysql”.

```
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]# yum install php php-mysql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package php-mysql is obsoleted by php-mysqld, trying to install php-mysqld-5.4.16-46.amzn2.
0.2.x86_64 instead
Resolving Dependencies
--> Running transaction check
--> Package php.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: php-cli(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-46.amzn2.0.2.x86_64
--> Processing Dependency: php-common(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-46.amzn2.0.2.x86_64
--> Package php-mysqld.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: php-pdo(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-mysqld-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
--> Package php-cli.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Package php-common.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: libzip.so.2()(64bit) for package: php-common-5.4.16-46.amzn2.0.2.x86_64
```

6. Type “yum search php” to see all the packages installed in the server.

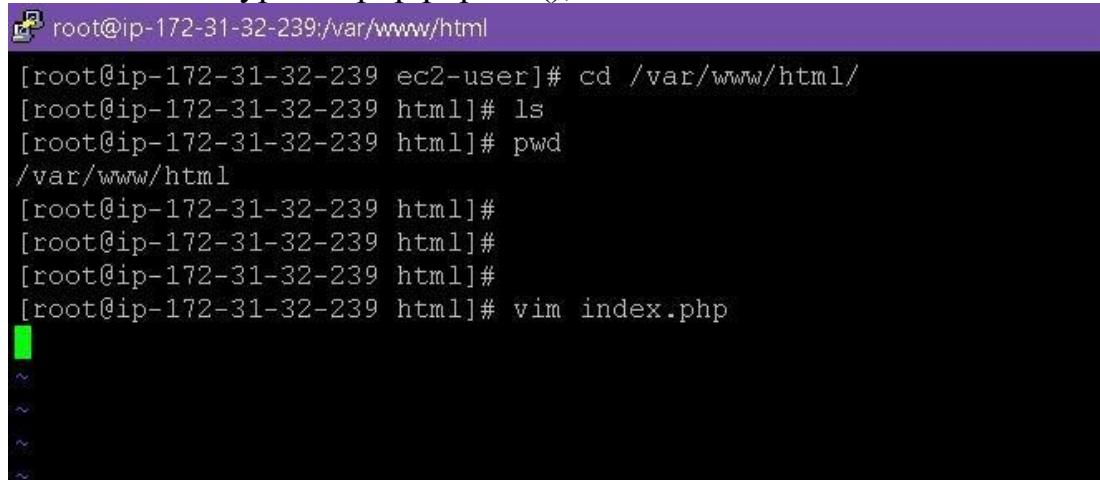
```
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]# yum search php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
=====
==== N/S matched: php =====
graphviz-php.x86_64 : PHP extension for graphviz
php.x86_64 : PHP scripting language for creating dynamic web sites
php-bcmath.x86_64 : A module for PHP applications for using the bcmath library
php-cli.x86_64 : Command-line interface for PHP
php-common.x86_64 : Common files for PHP
php-dba.x86_64 : A database abstraction layer module for PHP applications
php-devel.x86_64 : Files needed for building PHP extensions
php-embedded.x86_64 : PHP library for embedding in applications
php-enchant.x86_64 : Enchant spelling extension for PHP applications
php-fpm.x86_64 : PHP FastCGI Process Manager
php-gd.x86_64 : A module for PHP applications for using the gd graphics library
php-intl.x86_64 : Internationalization extension for PHP applications
php-ldap.x86_64 : A module for PHP applications that use LDAP
php-mbstring.x86_64 : A module for PHP applications which need multi-byte string handling
php-mysql.x86_64 : A module for PHP applications that use MySQL databases
php-mysqld.x86_64 : A module for PHP applications that use MySQL databases
php-odbc.x86_64 : A module for PHP applications that use ODBC databases
php-pdo.x86_64 : A database access abstraction module for PHP applications
php-pear.noarch : PHP Extension and Application Repository framework
php-pgsql.x86_64 : A PostgreSQL database module for PHP
php-process.x86_64 : Modules for PHP script using system process interfaces
php-pspell.x86_64 : A module for PHP applications for using pspell interfaces
php-recode.x86_64 : A module for PHP applications for using the recode library
php-snmp.x86_64 : A module for PHP applications that query SNMP-managed devices
php-soap.x86_64 : A module for PHP applications that use the SOAP protocol
php-xml.x86_64 : A module for PHP applications which use XML
php-xmlrpc.x86_64 : A module for PHP applications which use the XML-RPC protocol
rrdtool-php.x86_64 : PHP RRDtool bindings
uuid-php.x86_64 : PHP support for Universally Unique Identifier library
php-pecl-memcache.x86_64 : Extension to work with the Memcached caching daemon
```

7. Enabling the mariadb server.

```
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]# systemctl start mariadb
[root@ip-172-31-32-239 ec2-user]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
[root@ip-172-31-32-239 ec2-user]#
```

8. After enabling httpd (apache server) , go to the directory where cd /var/www/html/

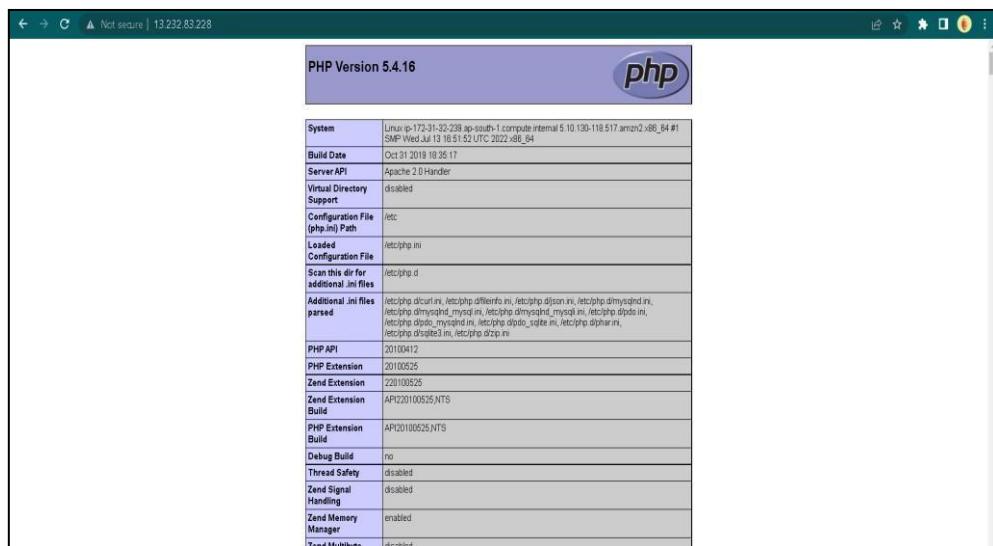
9. Go to vim and type “<?php phpinfo(); ?>”.



```
[root@ip-172-31-32-239 ec2-user]# cd /var/www/html/
[root@ip-172-31-32-239 html]# ls
[root@ip-172-31-32-239 html]# pwd
/var/www/html
[root@ip-172-31-32-239 html]#
[root@ip-172-31-32-239 html]#
[root@ip-172-31-32-239 html]#
[root@ip-172-31-32-239 html]# vim index.php
```

Copy the public ip address or public domain name from the console and paste in the web browser.

We get the following output.



RESULT:

LAMP instance was successfully created and executed in AWS CLI.

Experiment 3

Configuring Virtual Private Cloud VPC & Troubleshoot a VPC

AIM: To configure a Virtual Private Cloud VPC & Troubleshoot a VPC

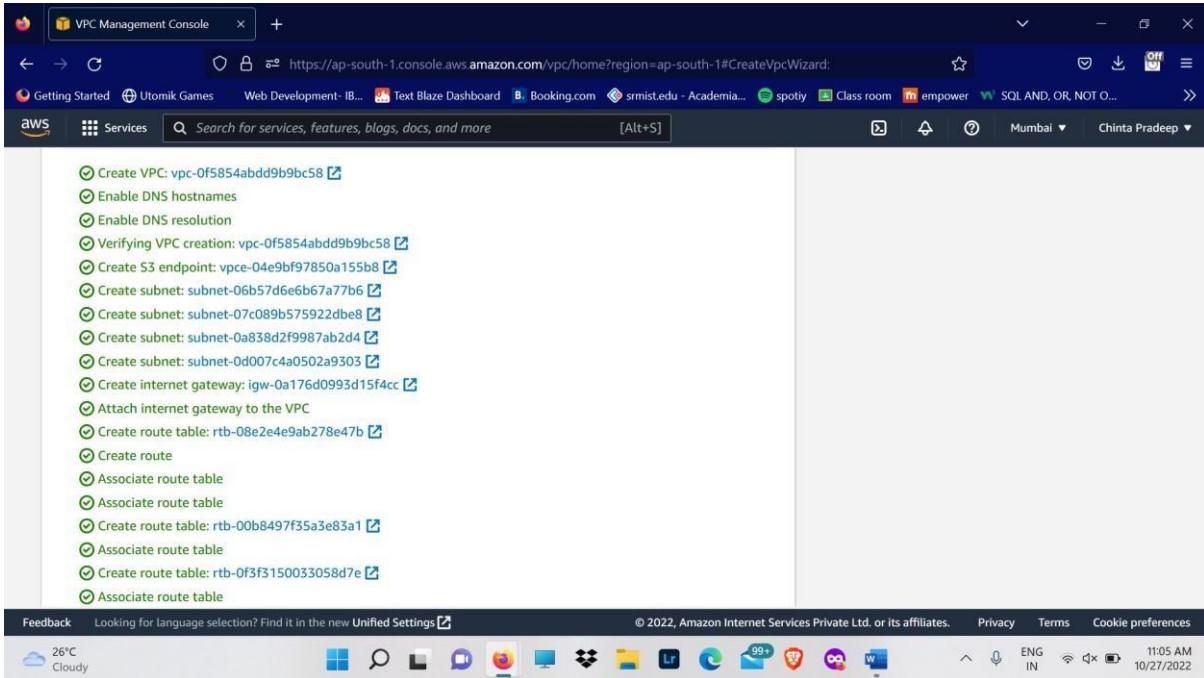
PROCEDURE:

1. Firstly, open the AWS console homepage on browser (<https://aws.amazon.com/console/>)

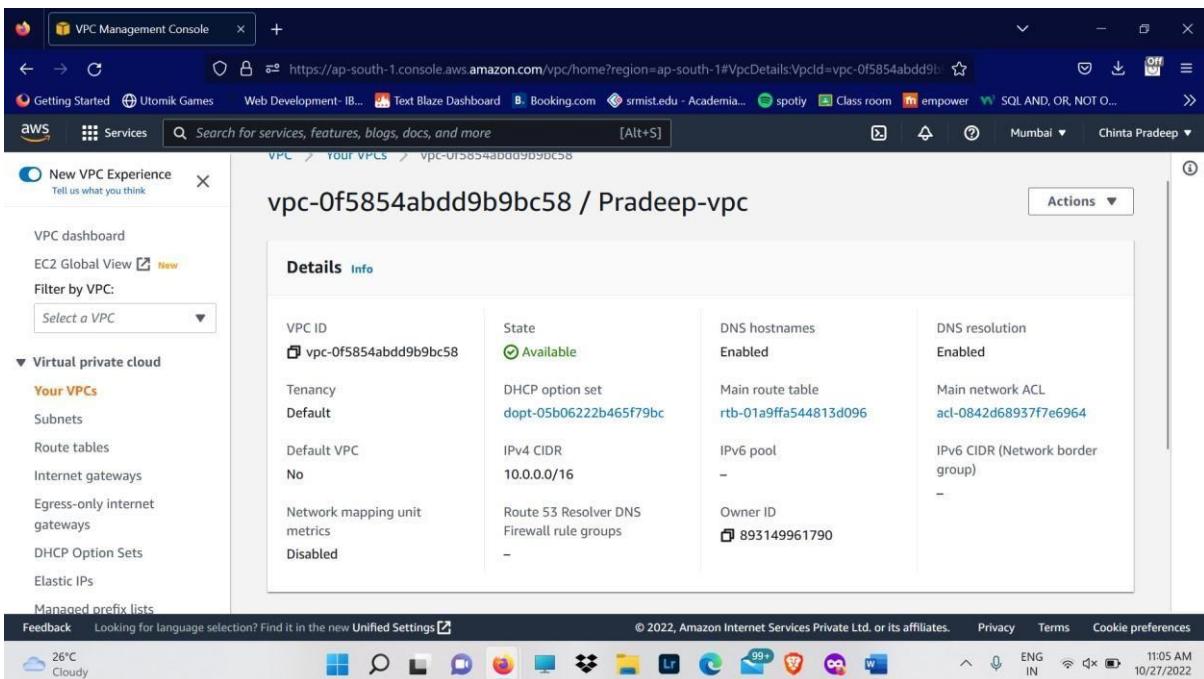
The screenshot shows the AWS VPC settings page. On the left, under 'Resources to create', 'VPC and more' is selected. Below it, 'Name tag auto-generation' is checked and set to 'Pradeep'. Under 'IPv4 CIDR block', the value '10.0.0.0/16' is entered, which is noted to provide 65,536 IPs. On the right, the 'Preview' section shows the newly created VPC 'Pradeep-vpc' and its four subnets: 'ap-south-1a' (containing 'Pradeep-subnet-public1-ap-south-1a' and 'Pradeep-subnet-private1-ap-south-1a') and 'ap-south-1b' (containing 'Pradeep-subnet-public2-ap-south-1b' and 'Pradeep-subnet-private2-ap-south-1b'). The interface includes standard browser navigation and AWS service icons at the top.

This screenshot is identical to the one above, showing the AWS VPC settings page. It displays the creation of a VPC named 'Pradeep-vpc' with four subnets across two availability zones ('ap-south-1a' and 'ap-south-1b'). The configuration details, including the CIDR block '10.0.0.0/16' and auto-generated name 'Pradeep', are visible on the left. The preview on the right shows the same subnet structure. The browser interface and system status bar at the bottom are also present.

2. Search for VPC in the search bar and open the VPC dashboard page.
3. To create a new VPC, click “Create VPC” and go to the Create VPC page and select the following configurations with appropriate VPC name.



4. Click on “Create VPC” button and wait for your VPC to be created.



5. Click on “View VPC” to view your VPC details.

Subnets (17) Info

Name	Subnet ID	State	VPC	IPv4 CIDR
projectvpc-subnet-...	subnet-002afafe2634275f	Available	vpc-02a44a4e1d7c9600c pro...	10.0.128.0/20
projectvpc-subnet-...	subnet-0045efad28fcf308f	Available	vpc-02a44a4e1d7c9600c pro...	10.0.16.0/20
chinta pradeep-sub...	subnet-002743a2235dc1e32	Available	vpc-0792d70c11f176b29 chi...	10.0.16.0/20
Pradeep-subnet-pr...	subnet-0ac6a2b1997bc2f2c	Available	vpc-0f46bb3bb1772184c Pra...	10.0.144.0/20

6. Click on “Subnets” in the left side menu to view subnets of your VPC.

Route tables (15) Info

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
Pradeep-rtb-public	rtb-054196b79227b2451	2 subnets	-	No	vpc-0f46b...
chinta pradeep-rtb...	rtb-0de6a339faae418ed	2 subnets	-	No	vpc-0792...
Pradeep-rtb-privat...	rtb-00b8497f35a3e83a1	subnet-0a838d2f9987a...	-	No	vpc-0f58...
-	rtb-01a9ffa544813d096	-	-	Yes	vpc-0f58...

7. Click on “Route Tables” in the left side menu to view route tables of your VPC.

Route tables (15) Info

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
Pradeep-rtb-public	rtb-054196b79227b2451	2 subnets	-	No	vpc-0f46b...
chinta pradeep-rtb...	rtb-0de6a339faae418ed	2 subnets	-	No	vpc-0792...
Pradeep-rtb-privat...	rtb-00b8497f35a3e83a1	subnet-0a838d2f9987a...	-	No	vpc-0f58...
-	rtb-01a9ffa544813d096	-	-	Yes	vpc-0f58...

8. Click on “Network ACL’s” in the left side menu to view ACLs of your VPC.

Network ACLs (5) Info

Name	Network ACL ID	Associated with	Default	VPC ID
-	acl-02bb4ae584d0e3d...	4 Subnets	Yes	vpc-02a44...
-	acl-0842d68937f7e6964	4 Subnets	Yes	vpc-0f585...
-	acl-0df7e3e6209195a61	3 Subnets	Yes	vpc-03fbdc...
-	acl-05619494687abeece	4 Subnets	Yes	vpc-0f46bt...

RESULT: A Virtual Private Cloud (VPC) was successfully created and troubleshooted.

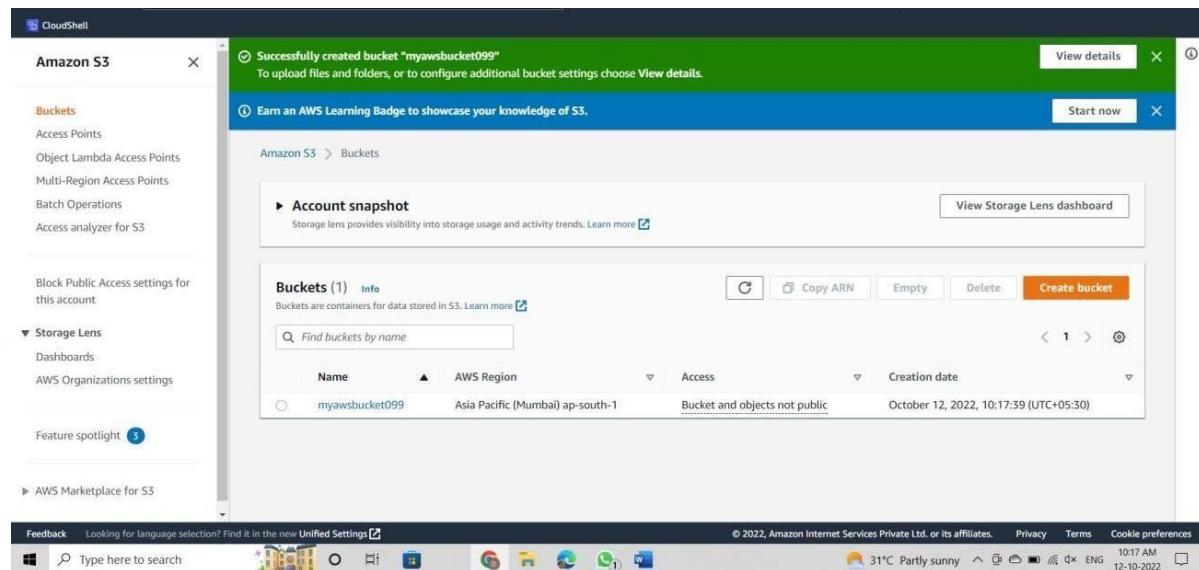
EXPERIMENT 4:

CREATING A STATIC WEBSITE ON AMAZON S3 USING THE AWS SYSTEM MANAGER

AIM: To create a static website on Amazon S3 using AWS system manager.

PROCEDURE:

- 1) Sign in to the AWS Management Console and click on Amazon S3 Console at <https://console.aws.amazon.com/s3/> 2) Click on create a bucket.
- 3) Enter the name of the bucket and the region you want to create it in
- 4) Accept the default settings and then click on create a bucket age.



- 5) In the buckets list, click on the bucket you just created and choose properties, then under static website hosting click the enable option.

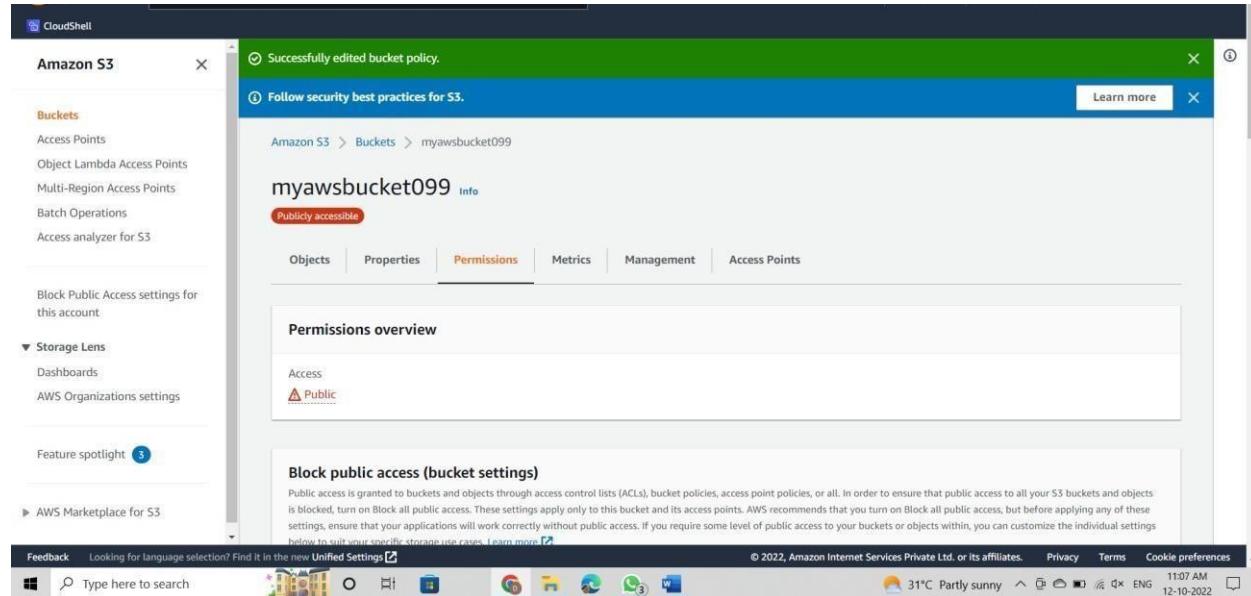
The screenshot shows the AWS S3 console with the path: Amazon S3 > Buckets > myawsbucket099 > Edit static website hosting. A green success message at the top says "Successfully edited static website hosting." Below it, a blue banner encourages earning an AWS Learning Badge. The main configuration area shows "Requester pays" set to "Disabled". Under "Static website hosting", "Enabled" is selected for "Static website hosting" and "Bucket hosting" for "Hosting type". The "Bucket website endpoint" is listed as <http://myawsbucket099.s3-website.ap-south-1.amazonaws.com>. The status bar at the bottom indicates it's from December 10, 2022, at 10:29 AM.

- 6) Then click on the upload option after selecting your bucket and upload your project files(index.html, style.css,main.js). Also, make sure that the project files you will upload have proper names as they are case-sensitive.
- 7) After configuring the website, you can use the endpoint to test the website out.

The screenshot shows the AWS S3 console with the path: CloudShell > myawsbucket099 > Upload succeeded. It displays a summary of the upload: 1 file, 1.1 KB (100.00%) succeeded. The "Files and folders" tab is selected, showing a table with one item: index.html (text/html, 1.1 KB, Succeeded). The status bar at the bottom indicates it's from December 10, 2022, at 10:31 AM.

8) choose the name of the bucket you just configured and then choose permissions under that block public access will be visible, click on the edit option and save the changes.

9) under the bucket permissions there will be a bucket policy, click on that and paste this code into the bucket policy editor:



{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "*",

"Action": ["s3:GetObject"],

"Resource": ["arn:aws:s3:::myawss3bucket1/*"

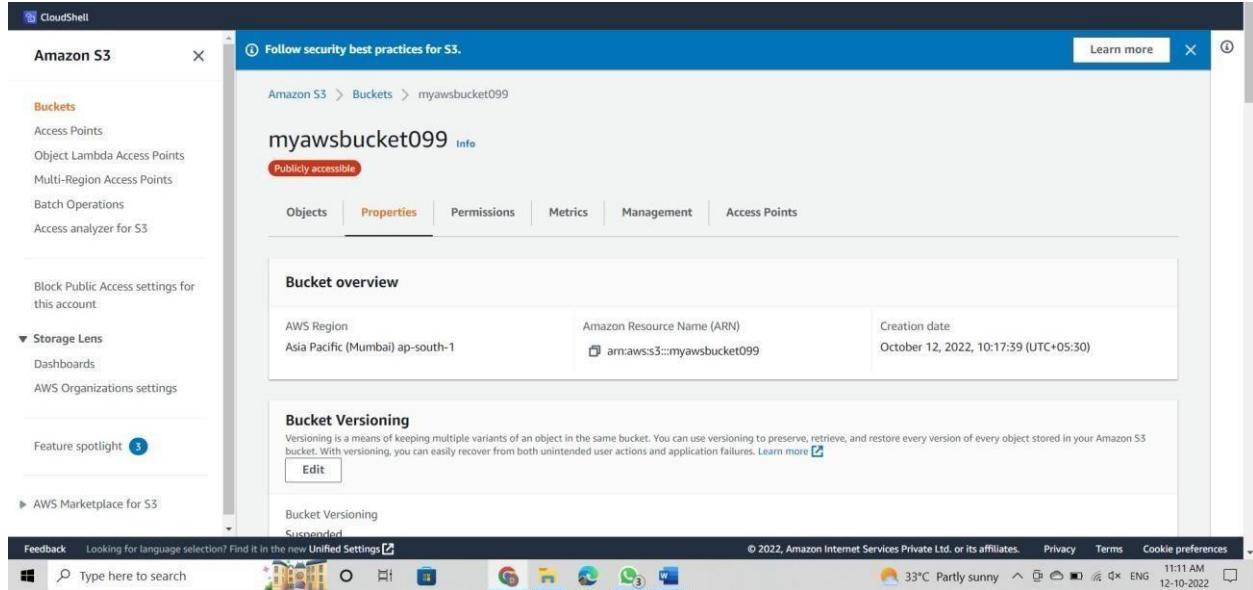
]

}

]

}

10) Click on save changes



11) Under the buckets, choose the name of your bucket which you just created, and in the properties, choose a static website hosting - then choose your bucket endpoint

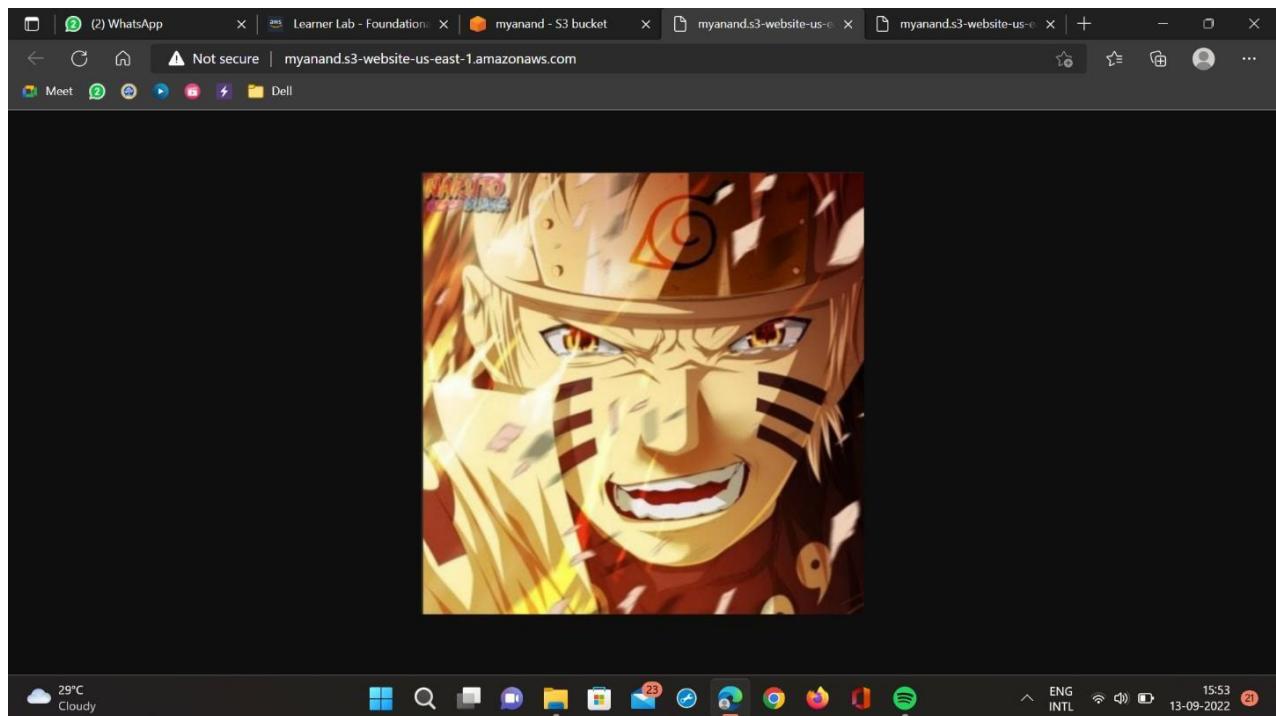
12) Now you can access your website as it has successfully been hosted.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and Access analyzer for S3. Below that are sections for Block Public Access settings and Storage Lens. A Feature spotlight section is also present. The main area shows the 'myawsbucket099' bucket details, including its name, type (Publicly accessible), and a list of 7 objects. The objects are:

Name	Type	Last modified	Size	Storage class
image.png	png	October 12, 2022, 11:14:24 (UTC+05:30)	435.9 KB	Standard
index.html	html	October 12, 2022, 10:31:46 (UTC+05:30)	1.1 KB	Standard
logo.png	png	October 12, 2022, 11:14:25 (UTC+05:30)	6.3 KB	Standard
pause.png	png	October 12, 2022, 11:14:25 (UTC+05:30)	5.4 KB	Standard

At the bottom, there are links for Feedback, Unified Settings, and cookie preferences. The status bar at the bottom right shows the date (12-10-2022), time (11:20 AM), and location (ENG).

<http://myawsbucket099.s3-website.ap-south-1.amazonaws.com/>



13) Make sure to clean up after this experiment by deleting the AWS Resources that have been just allocated . After you have deleted these resources, your website will no longer be available .

RESULT:

A static website has been successfully created on Amazon S3 using AWS System Manager.

EXPERIMENT NO: 5

Automation and Optimization with Amazon S3

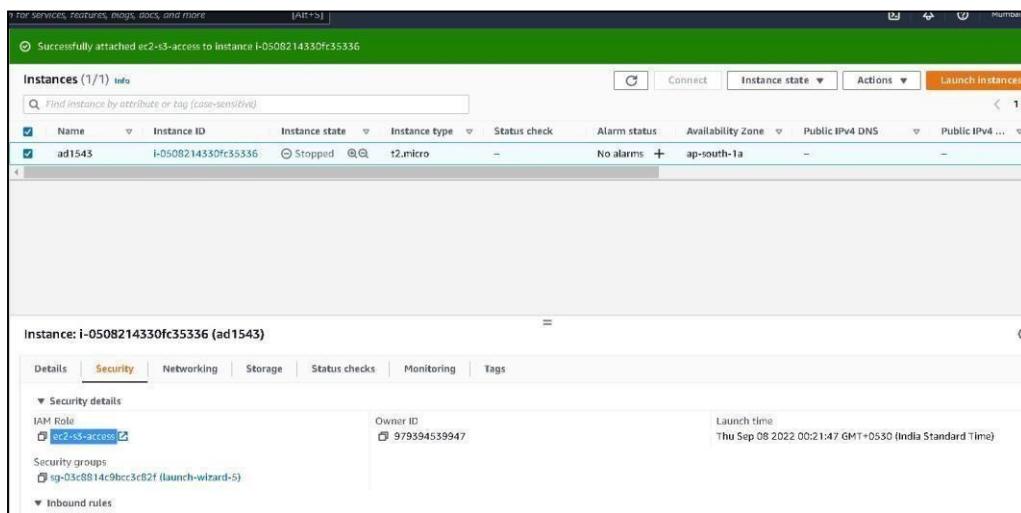
Aim: Automate Files backup to aws S3 bucket on Linux machine.

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

Procedure:

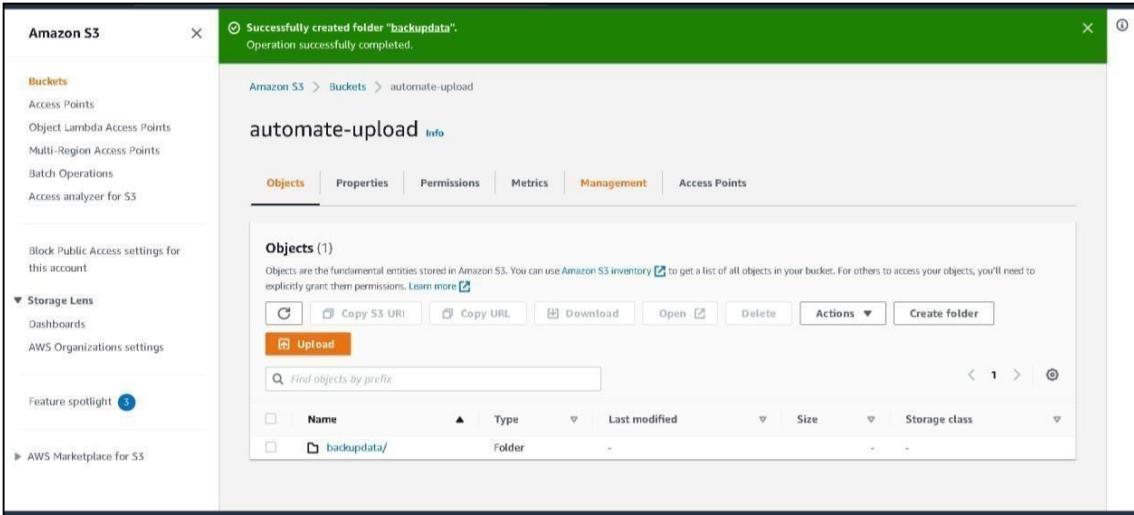
Steps:

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



(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

```
2022-09-19 08:27:17 paint-ed1543
[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 sync 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
  cp
  rm
  ab
  presign
  website
  sv
  sync
  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: ./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

The screenshot shows the AWS S3 console interface. At the top, there's a green banner indicating "Successfully deleted objects". Below it, the navigation path is "Amazon S3 > Buckets > automate-upload". The main area displays the contents of the "automate-upload" bucket. A table lists three objects: "a", "b", and "c". All three objects were uploaded on September 22, 2022, at 09:34:00 UTC+05:30. They are all 0 B in size and have the Standard storage class. The table has columns for Name, Type, Last modified, Size, and Storage class.

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	a	-	September 22, 2022, 09:34:00 (UTC+05:30)	0 B	Standard
<input type="checkbox"/>	b	-	September 22, 2022, 09:34:00 (UTC+05:30)	0 B	Standard
<input type="checkbox"/>	c	-	September 22, 2022, 09:34:00 (UTC+05:30)	0 B	Standard

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
ab           | 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/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 a d file e have been updated.

Name	Type	Last modified	Size	Storage class
a	-	September 22, 2022, 09:34:00 (UTC+05:30)	0 B	Standard
b	-	September 22, 2022, 09:34:00 (UTC+05:30)	0 B	Standard
c	-	September 22, 2022, 09:34:00 (UTC+05:30)	0 B	Standard
d	-	September 22, 2022, 09:43:03 (UTC+05:30)	0 B	Standard
e	-	September 22, 2022, 09:43:03 (UTC+05:30)	0 B	Standard

Result:

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

EXPERIMENT NO: 6

Querying Data in S3 with Amazon Athena

Aim: Querying Data in S3 with Amazon Athena

Step 1:

Go to buckets and create two buckets.

The screenshot shows the Amazon S3 console interface. On the left, there's a sidebar with options like 'Buckets', 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main area is titled 'Amazon S3' and shows a message about design updates. Below that is a table titled 'Buckets (2)'. The table has columns for 'Name', 'Region', 'Access', and 'Bucket created'. It lists two buckets: 'aws-simplified-athena-demo' (Region: US East (N. Virginia) us-east-1, Access: Not Public, Created: 2020-03-20T23:48:56.000Z) and 'aws-simplified-results' (Region: US East (N. Virginia) us-east-1, Access: Not Public, Created: 2020-03-20T23:40:43.000Z). There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket' at the top right of the table.

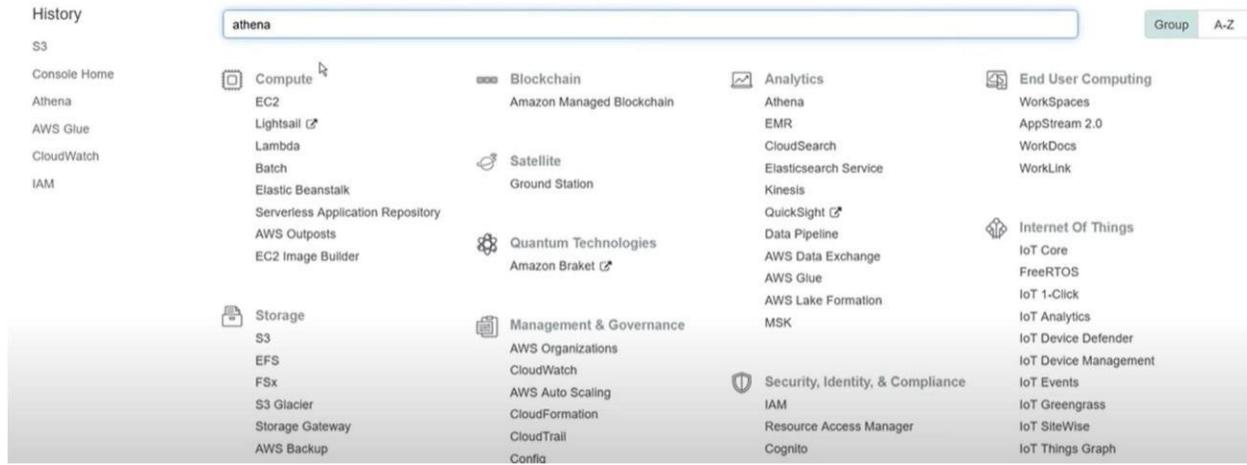
Step 2 :

After clicking onto the bucket add files to it.

The screenshot shows the 'aws-simplified-athena-demo' bucket details. At the top, there are tabs for 'Overview', 'Properties', 'Permissions', 'Management', and 'Access points'. The 'Overview' tab is selected. Below the tabs is a search bar with placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' Underneath is a toolbar with 'Upload', '+ Create folder', 'Download', and 'Actions' dropdown. To the right, it shows the region as 'US East (N. Virginia)' and a note 'Viewing 1 to 1'. A table lists the file 'transactions.json' with columns for Name, Last modified, Size, and Storage class. The file was last modified on Mar 20, 2020 at 7:49:26 PM GMT-0400, is 78.1 KB in size, and is in the Standard storage class.

Step 3 :

Now go to Amazon athena.



Step 4 :

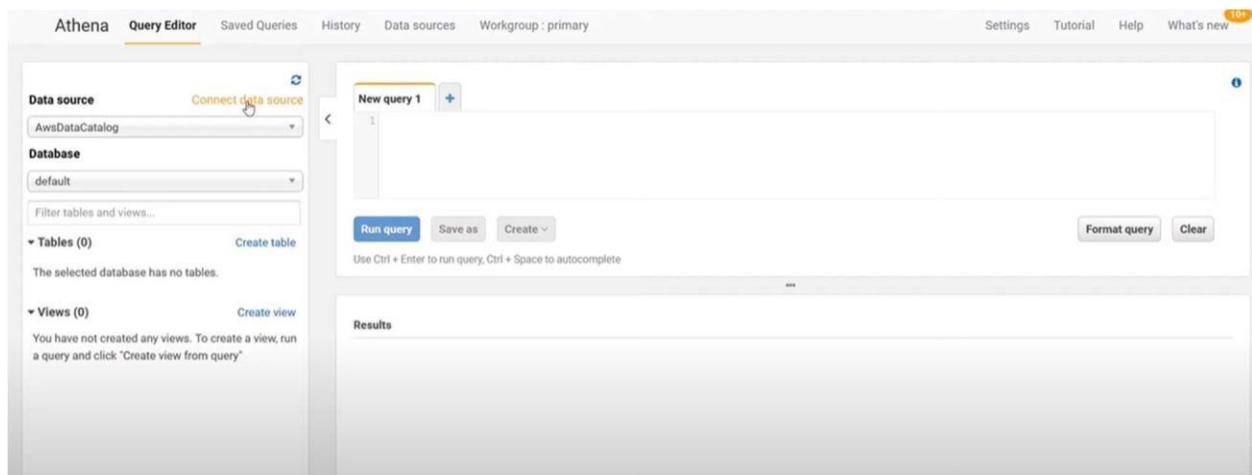
Select AwsDataCatalog in the left side which is present in the data source tab.

Step 5 :

After that go to settings and specify an output path.

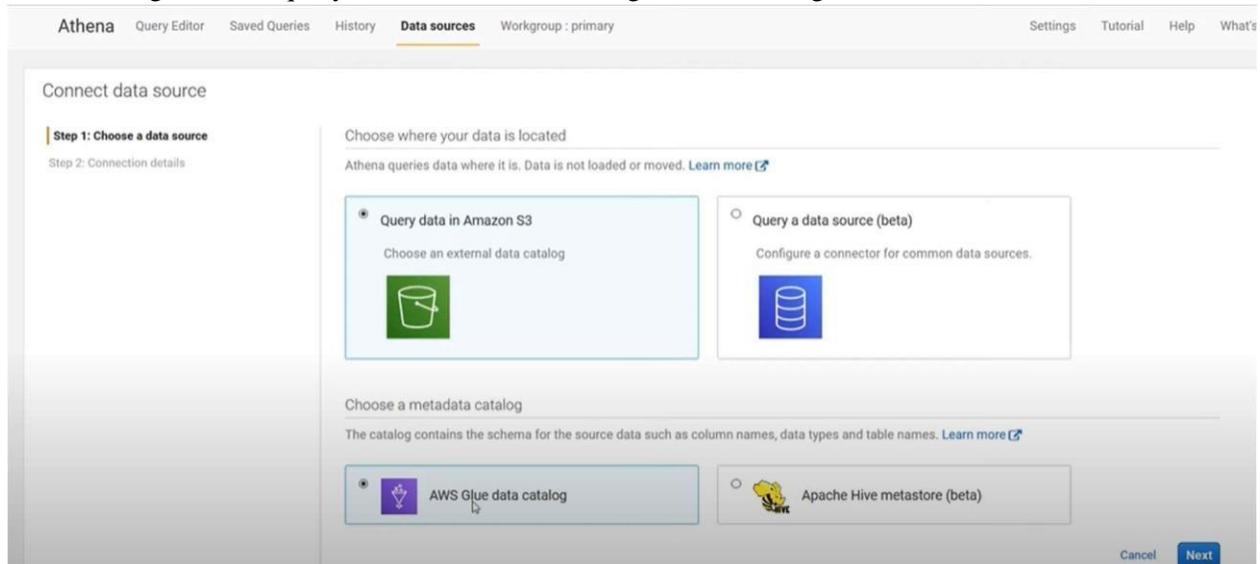
Step 6:

Click on connect data source.



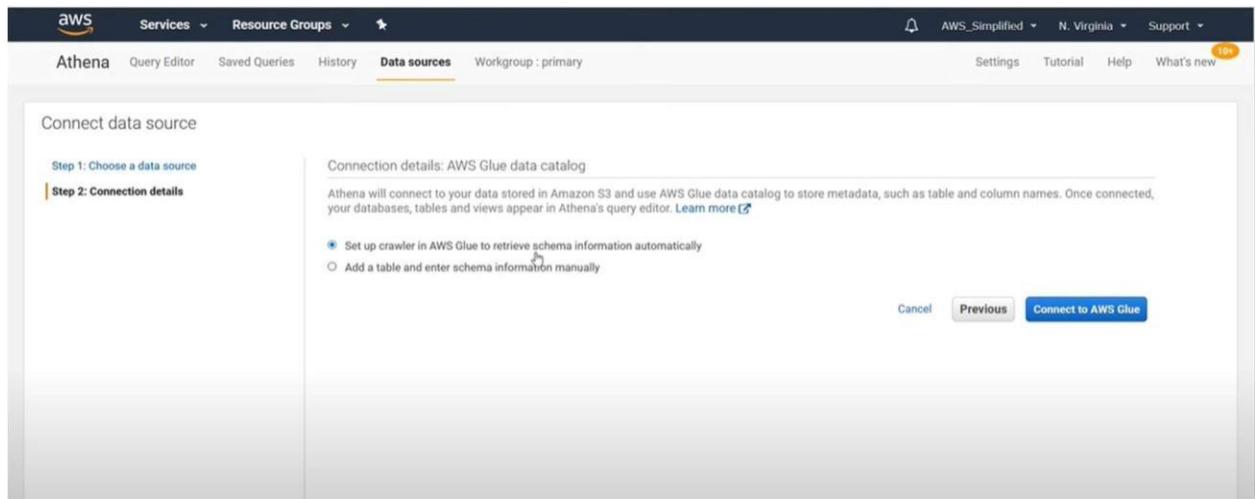
Step 7:

After clicking choose a query in amazon s3 and Aws glue data catalog.

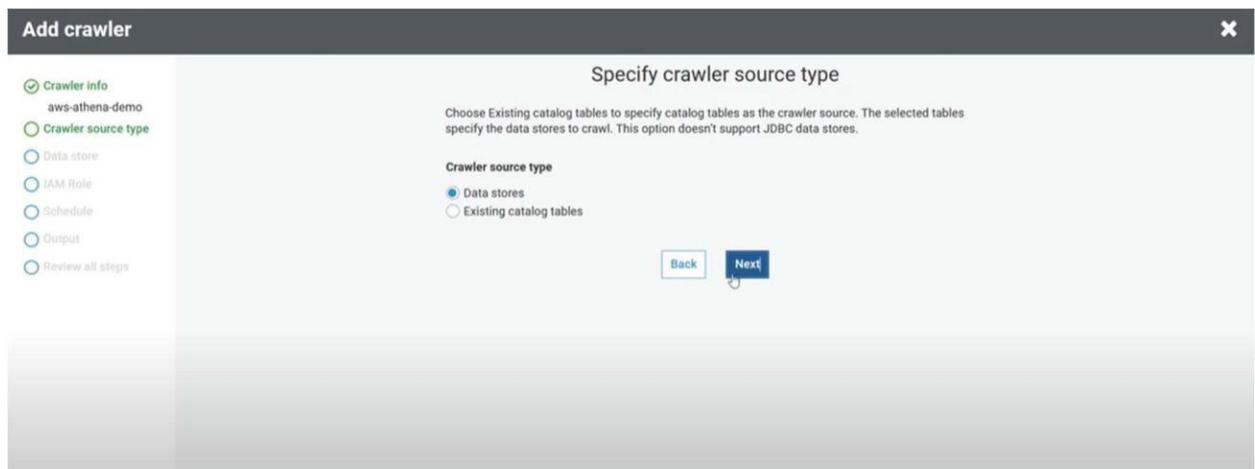
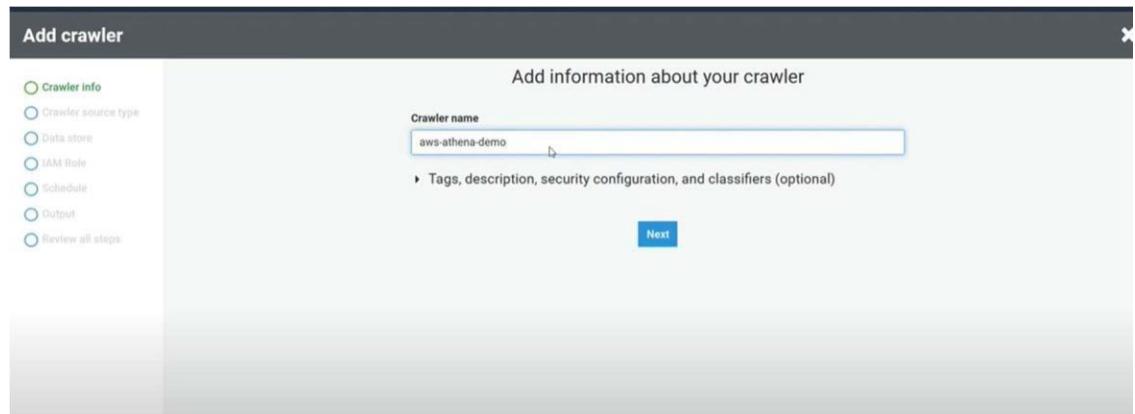


Step 8:

Click on next and select setup a crawler in AWS glue to retrieve schema information automatically.



Step 9: After selecting that it will redirect to a new page and add crawler and follow below steps to add a new crawler after setting up click on finish.



Add crawler

Crawler info
aws-athena-demo

Crawler source type
Data stores

Data store

IAM Role

Schedule

Output

Review all steps

Add a data store

Choose a data store: S3

Crawl data in:

- Specified path in my account
- Specified path in another account

Include path: s3://bucket/prefix/object

Exclude patterns (optional)

[Back](#) [Next](#)

Add crawler

Crawler info
aws-athena-demo

Crawler source type
Data stores

Data store
S3: s3://aws-simplifi...

IAM Role

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role

Choose an existing IAM role

Create an IAM role

IAM role: AWSGlueServiceRole- demo

To create an IAM role, you must have **CreateRole**, **CreatePolicy**, and **AttachRolePolicy** permissions.

Create an IAM role named "AWSGlueServiceRole-rolename" and attach the AWS managed policy, **AWSGlueServiceRole**, plus an inline policy that allows read access to:

- s3://aws-simplified-athena-demo/

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

Add crawler

Crawler info
aws-athena-demo

Crawler source type
Data stores

Data store
S3: s3://aws-simplifi...

IAM Role
arn:aws:iam::398447858632:role/service-role/AWSGlueServiceRole-demo

Schedule

Output

Review all steps

Create a schedule for this crawler

Frequency: Run on demand

[Back](#) [Next](#)

Add crawler

Crawler info
aws-athena-demo

Crawler source type
Data stores

Data store
S3: s3://aws-simpli...

IAM Role
arn:aws:iam::3984478
58632:role/service-
role/AWSGlueService-
Role-demo

Schedule
Run on demand

Output
default

Review all steps

Crawler info

Name: aws-athena-demo
Tags: -

IAM role

IAM role: arn:aws:iam::398447858632:role/service-role/AWSGlueServiceRole-
demo

Schedule

Schedule: Run on demand

Output

Database: default
Prefix added to tables (optional):
Create a single schema for each S3 path: false

Step 10:

Crawler is successfully created and now click on the crawler and click run crawler.

	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input checked="" type="checkbox"/>	aws-athena-demo		Ready		0 secs	0 secs	0	0

Step 11:

After running the crawler go back to athena you will see a table created on table column select that and click on preview table.

Athena Query Editor Saved Queries History Data sources Workgroup: primary Settings Tutorial Help What's new

Data source: AwsDataCatalog Connect data source

Database: default

Tables (1) Create table
aws_simplified_athena_demo

Views (0) Create view

You have not created any views. To create a view, run a query and click 'Create view from query'

New query 1

Preview table Show properties Delete table Generate Create Table DDL

Format query Clear

Results

Step 12:

Now the query can be executed.

Data source: AwsDataCatalog

Database: default

Tables (1): aws_simplified_athena_demo

Views (0):

```
1 SELECT * FROM "default"."aws_simplified_athena_demo" WHERE type='PURCHASE' and amount >= 50;
```

Run query | **Save as** | **Create** | (Run time: 1.81 seconds, Data scanned: 78.08 KB)

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Learner Lab - Foundational Services

Logs

arn:aws:lambda:us-east-1:211621018401:function:LearnerLabFoundationalServices

2022-10-12T14:36:27.846Z | INFO | 12-10-2022 | [Lambda] User [vetahsuser211621018401] has triggered a new Lambda function execution.

RESULT:

Querying Data in S3 with Amazon Athena is done and output is verified.

7. Procedure to Create an AWS Lambda Function to Email Daily Reports

Short description

To send email from a Lambda function using Amazon SES, do the following:

1. Create an AWS Identity and Access Management (IAM) policy and execution role for Lambda to run the API call.
2. Verify your [Amazon SES identity](#) (domain or email address).
3. Create or update a Lambda function that includes logic for sending email through Amazon SES.

Note: To include a PDF attachment in your emails, you must use the [Amazon SES Send Raw Email](#) API operation. For more information, see [Sending raw email using the Amazon SES API](#).

Resolution

Note: The example [Node.js Lambda function code](#) in this article is provided as-is. Adapt the example to your use case, or design your own in your [preferred programming language](#).

Create an IAM policy and execution role for Lambda to run the API call

1. [Create an IAM policy using the JSON policy editor](#). When you create the policy, paste the following JSON policy document into the policy editor:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ses:SendEmail",  
                "ses:SendRawEmail"  
            ],  
            "Resource": "*"  
        }  
    ]}
```

Note: For more information and examples of how to restrict access to this policy, see [Example IAM policies for Amazon SES](#).

2. [Attach the IAM policy](#) to an IAM role. For instructions, see the To use a managed policy as a permissions policy for an identity (console) section in [Adding IAM identity permissions \(console\)](#).

Note: You will assign this IAM role to your Lambda function in the following steps.

Verify your Amazon SES identity (domain or email address)

To verify a domain, see [Verifying domains in Amazon SES](#).

To verify an email address, see [Verifying email addresses in Amazon SES](#).

Create or update a Lambda function that includes logic for sending email through Amazon SES

1. If you haven't done so already, [create a Lambda function](#).

Note: You can create a Lambda function by [using the Lambda console](#) or by [building and uploading a deployment package](#).

2. In the [Lambda console](#), in the left navigation pane, choose Functions.
3. Choose the name of your function.
4. On the Configuration tab, in the Permissions pane, look at the function's Execution Role. Verify that the IAM role with Amazon SES permissions that you created earlier is listed. If the correct IAM role isn't listed, assign the correct role to the function.
5. Under Function code, in the [editor pane](#), paste the following example function code:

Important: Replace us-west-2 with the [AWS Region](#) that your verified Amazon SES identity is in. Replace "RecipientEmailAddress", ... with the email address or addresses that you want to send the email to. Replace SourceEmailAddress with your Amazon SES-verified sender email address, or any email address from an [Amazon SES-verified domain](#). Optionally, edit the message body ("Test") and subject line ("Test Email").

```
// Copyright 2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.//  
SPDX-License-Identifier: Apache-2.0
```

```
var aws = require("aws-sdk");  
var ses = new aws.SES({ region: "us-west-2" });  
exports.handler = async function (event) {  
  var params = { Destination: {  
    ToAddresses: ["RecipientEmailAddress", ...],  
  },  
    Message: {  
      Body: {  
        Text: { Data: "Test" },  
      },  
      Subject: { Data: "Test Email" },  
    },  
    Source: "SourceEmailAddress",  
  };  
  return ses.sendEmail(params).promise();  
};
```

For more information on using the sendEmail API, see the [AWS SDK for JavaScript documentation](#).

5. Choose Deploy.

Send a test email

1. In the [Lambda console](#), [configure a test event](#) for your function.

Note: The test payload is required but isn't used for this code example.

- Choose Test. Lambda uses Amazon SES to send the test email to your recipient.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and navigation links for Services, Resource Groups & Tag Editor, and IAM. The main area displays a success message: "The test event anandemail was successfully saved." Below this, there are tabs for "Code source" and "Info". Under "Code source", there's a file tree showing "index.js" under the "anand" folder. The "Execution results" tab is selected, showing the following details:

- Test Event Name:** anandemail
- Response:**

```
{
  "statusCode": 200,
  "body": "\nHello from Lambda!\n"
}
```
- Function Logs:**

```
START RequestId: 87c73d71-f6a3-4cdc-8d25-e46909d8e606 Version: $LATEST
END RequestId: 87c73d71-f6a3-4cdc-8d25-e46909d8e606
REPORT RequestId: 87c73d71-f6a3-4cdc-8d25-e46909d8e606 Duration: 23.25 ms Billed Duration: 24 ms Memory Size: 128 MB Max Memory Used: 57 MB
Request ID
87c73d71-f6a3-4cdc-8d25-e46909d8e606
```

At the bottom, there are links for Feedback, © 2022, Amazon Internet Services Private Ltd. or its affiliates., Privacy, Terms, and Cookie preferences.

This screenshot shows the AWS Lambda console with the same interface as the previous one, but the "Execution results" tab is not selected. Instead, the "Code source" tab is selected, displaying the contents of "index.js". The code is as follows:

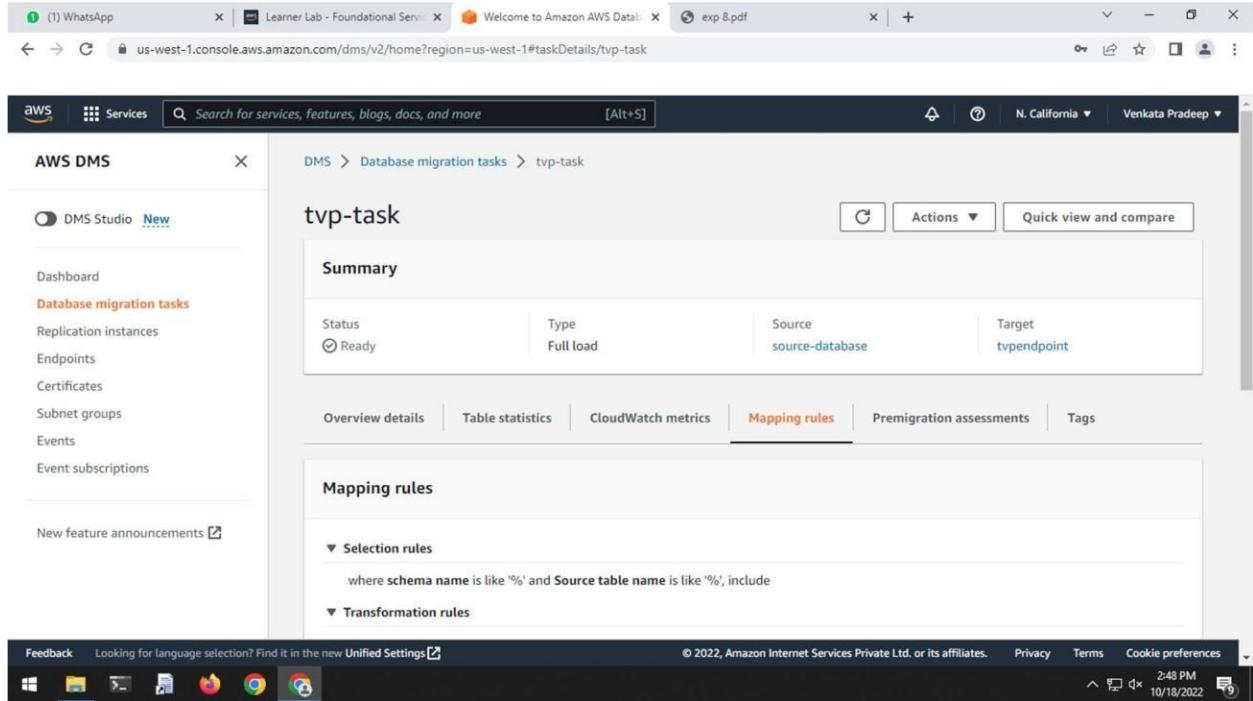
```

1 // Copyright 2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
2 // SPDX-License-Identifier: Apache-2.0
3
4 var aws = require("aws-sdk");
5 var ses = new aws.SES({ region: "us-west-2" });
6 exports.handler = async function (event) {
7   var params = {
8     Destination: {
9       ToAddresses: ["tv6624@srmist.edu.in"],
10    },
11    Message: {
12      Body: {
13        Text: { Data: "Test" },
14      },
15      Subject: { Data: "Test Email" },
16    },
17    Source: "ba9564@srmist.edu.in",
18  };
19
20  i 21 return ses.sendEmail(params).promise()
21 };
```

At the bottom, there are links for Feedback, © 2022, Amazon Internet Services Private Ltd. or its affiliates., Privacy, Terms, and Cookie preferences.

8. PROCEDURE OF MIGRATE TO AMAZON RDS

1. Login to your AWS account



The screenshot shows the AWS DMS (Database Migration Service) console. A specific migration task named 'tvp-task' is being viewed. The 'Mapping rules' tab is currently selected. The 'Summary' section indicates the task is 'Ready' (Status), 'Full load' (Type), and involves 'source-database' as the Source and 'tvpendpoint' as the Target. Below the summary, the 'Mapping rules' tab is active, showing sections for 'Selection rules' and 'Transformation rules'. The selection rules define a pattern where schema name is like '%` and source table name is like %` include. The transformation rules section is collapsed.

2. Search for DATABASE MANAGEMENT SERVICE

The screenshot shows the AWS DMS console interface. On the left, there's a navigation sidebar with options like Dashboard, Database migration tasks, Replication instances, Endpoints, Certificates, Subnet groups, Events, and Event subscriptions. The main area displays a 'tvp-task' under 'Database migration tasks'. The 'Summary' section shows the task status as 'Ready', type as 'Full load', source as 'source-database', and target as 'tvpendpoint'. Below the summary, there are tabs for Overview details, Table statistics, CloudWatch metrics, Mapping rules (which is currently selected), Premigration assessments, and Tags. The 'Mapping rules' section contains two expandable sections: 'Selection rules' (with the note 'where schema name is like '%' and Source table name is like '%', include') and 'Transformation rules'.

3. Open REPLICATION INSTANCES

The screenshot shows the AWS DMS (Database Migration Service) console. The left sidebar has 'AWS DMS' selected under 'Database migration tasks'. The main area shows a 'tvp-task' migration task. The 'Summary' tab is active, displaying basic information: Status (Ready), Type (Full load), Source (source-database), and Target (tvpendpoint). Below the summary, there are tabs for Overview details, Table statistics, CloudWatch metrics, Mapping rules (which is selected), Premigration assessments, and Tags. Under 'Mapping rules', sections for Selection rules and Transformation rules are visible. The bottom of the screen shows the standard AWS navigation bar with links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences, along with system status icons.

4. Click on CREATE REPLICATION INSTANCE
5. Enter your replication instance NAME, DESCRIPTION
6. Give the required details and click CREATE

The screenshot shows the AWS DMS console with the URL us-west-1.console.aws.amazon.com/dms/v2/home?region=us-west-1#endpointDetails/source-database. The left sidebar is titled 'AWS DMS' and includes 'Endpoints' under the 'Replication instances' section. The main content area is titled 'source-database' and shows the 'Overview details' tab selected. It displays various endpoint details:

Detail	Value
Endpoint ARN	arn:aws:dms:us-west-1:864550822886:endpoint:ECOXLOWU2FLFFKOZOPNXL2CDITFDMUH2X3MMIY
Status	Active
SSL mode	None
Endpoint engine	MySQL
Server name	source-database.c39ashxlumfg.us-west-1.rds.amazonaws.com
Port	3306
User name	venkata pradeep
Extra connection attributes	(empty)

The screenshot shows the AWS DMS console with the URL us-west-1.console.aws.amazon.com/dms/v2/home?region=us-west-1#replicationInstances. The left sidebar is titled 'AWS DMS' and includes 'Replication instances' under the 'Replication instances' section. The main content area is titled 'Replication instances (1)' and shows a single instance named 'pradeep-instance' listed in a table:

Name	Status	VPC	Class	Engine version	Availability zone
pradeep-instance	Available	vpc-0f779c7749d989185	dms.t3.medium	3.4.7	us-west-1a

The screenshot shows the AWS DMS (Database Migration Service) console. The left sidebar has a navigation menu with options like DMS Studio, Dashboard, Database migration tasks, Replication instances, Endpoints (which is highlighted in orange), Certificates, Subnet groups, Events, and Event subscriptions. Below the sidebar is a "New feature announcements" section. The main content area is titled "Endpoints (2)" and shows a table with two rows of endpoint information. The columns are Name, Type, Status, Engine, Server name, and Port. The first row is for a "source-database" source endpoint, and the second row is for a "tvpendpoint" target endpoint. Both endpoints are listed as Active and MySQL engines. The table includes a search bar at the top and a "Create endpoint" button at the top right.

Name	Type	Status	Engine	Server name	Port
source-database	Source	Active	MySQL	source-database.c39ashxlumfg.us-west-1.rds.amazonaws.com	3306
tvpendpoint	Target	Active	MySQL	source-database.c39ashxlumfg.us-west-1.rds.amazonaws.com	3306

7. From the console choose ENDPOINTS and create endpoint

The image consists of three vertically stacked screenshots from the AWS DMS console.

Screenshot 1: Database migration tasks

- Shows the "Database migration tasks" page with one task named "tvp-task" listed.
- The task details: Identifier: tvp-task, Status: Ready, Progress: 0%, Type: Full load, Source: source-database, Target: tvpendpoint, Replication instance: pradeep-instance.

Screenshot 2: Replication instance summary

- Shows the "Replication instance summary" page for a replication instance named "pradeep-instance".
- Details: Class: dms.t3.medium, Engine version: 3.4.7, Status: Available, Associated migration tasks: 1.
- ARN: arn:aws:dms:us-west-1:1864550822886:rep:test-instance
- Status: Available
- Public IP address: (not explicitly shown)

Screenshot 3: Create task

- Shows the "Create task" page.
- Task identifier: tvp-task, Status: Ready, Progress: 0%, Type: Full load, Source: source-database, Target: tvpendpoint, Replication instance: pradeep-instance.

8. Select the ENDPOINT TYPE

9. Choose RUN TEST button and when the status is successful click on CREATE ENDPOINT

10. Repeat the above steps to create an instance but this ENDPOINT TYPE is Target endpoint

11. Choose the DATABASE MIGRATION TASKS on the console and choose CREATE TASK

12. Create the task by giving required details

13. Click on Migrate existing data for one time migration and in task setting update the fields as required.

14. After giving the required details choose create task.

15. Run the task.

16. Click on the task and click TABLE STATISTICS to view the successfully completed tasks.

RESULT:

Finally, we have done migration to Amazon RDS.

9.

Configure Failover Routing with Amazon Route 53

PROCEDURE:

The screenshot shows the AWS Free Tier landing page. At the top, there's a banner for 'Startups may be eligible for AWS credits'. Below it, a section titled 'Types Of Offers' lists three categories: a clock icon for 'AWS Compute Credits', a network icon for 'AWS Compute Credits', and a person icon for 'AWS Support'.

Login to your AWS account

1.

The screenshot shows the AWS Route 53 Hosted Zones interface. On the left, a sidebar lists options like Dashboard, Hosted zones (which is selected), Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, Pending requests, Resolver, VPCs, Inbound endpoints, and Outbound endpoints. The main area displays a table of record sets for the 'avinash.website' zone. The table has columns for Name, Type, Value, and Evaluate Target Health. Two entries are shown: one for NS type pointing to four IP addresses (ns-165.awsdns-20.com, ns-1919.awsdns-47.co.uk, ns-1090.awsdns-08.org, ns-709.awsdns-24.net) and another for SOA type with the same target values.

Go to Hosted zones.

2.

Dashboard

Hosted zones

Health checks

Traffic flow

Traffic policies

Policy records

Domains

Registered domains

Pending requests

Resolver

Welcome to Route 53 health checks

Route 53 health checks monitor the health and performance of your application's servers, or endpoints, from a network of health checkers in locations around the world. To specify either a domain name or an IP address and a port to create HTTP, HTTPS, and TCP health checks that check the health of the endpoint. To get started, click [Create health check](#).

Create health check

Health check concepts



Availability and performance monitoring



DNS failover

Go to health checks and create health check

3.

outage occurs.

Name [i](#)

What to monitor Endpoint [i](#)
 Status of other health checks (calculated health check) [i](#)
 State of CloudWatch alarm [i](#)

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy.
[Learn more](#)

Specify endpoint by IP address Domain name [i](#)

Protocol [i](#)

IP address * [i](#)

Host name [i](#)

Port * [i](#)

Path [i](#)

Give the required details.

4.

outage occurs.

Name [i](#)

What to monitor Endpoint [i](#)
 Status of other health checks (calculated health check) [i](#)
 State of CloudWatch alarm [i](#)

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy.
[Learn more](#)

Specify endpoint by IP address Domain name [i](#)

Protocol [i](#)

Domain name * [i](#)

Port * [i](#)

Path [i](#)

[Advanced configuration](#) [i](#)

Give the endpoint of which you want to monitor.

5.

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy.
Learn more

Specify endpoint by IP address Domain name

Protocol **HTTP**

Domain name * **mumbaiELB-25996257.ap-south-1.e**

Port * **80**

Path **/images**

[Advanced configuration](#)

URL **http://mumbaiELB-25996257.ap-south-1.elb.amazonaws.com:80/**

Health check type **Basic - no additional options selected (View Pricing)**

Copy paste the URL in a new tab to check if it is healthy.

6.

Create health check

Step 1: Configure health check

Step 2: Get notified when health check fails

Get notified when health check fails

If you want CloudWatch to send you an Amazon SNS notification, such as an email, when the status of the health check changes to unhealthy, create an alarm and specify where to send notifications.

Create alarm Yes No

CloudWatch sends you an Amazon SNS notification whenever the status of this health check is unhealthy for one minute.

Send notification to Existing SNS topic New SNS topic

Default_CloudWatch_Alarms_Topic

* Required Cancel Previous Create health check

If your health check fails then you can set notification and click on create health check.

7.

Dashboard
Hosted zones
Health checks
Traffic flow
Traffic policies
Policy records
Domains
Registered domains
Pending requests
Resolver
VPCs
Inbound endpoints
Outbound endpoints
Rules

Health check with id 9459b641-1d77-4853-b12e-6d9bd9d0d6b3 has been created successfully

Name	Status	Description	Alarms
prodthc	Unknown	http://mumbaiELB-25996257.ap-south-1....	⚠ 1 of 1 in INSUFFICIENT

No health check selected.

No health check selected.

Health check is created and status is unknown and soon it will turn healthy because it is healthy

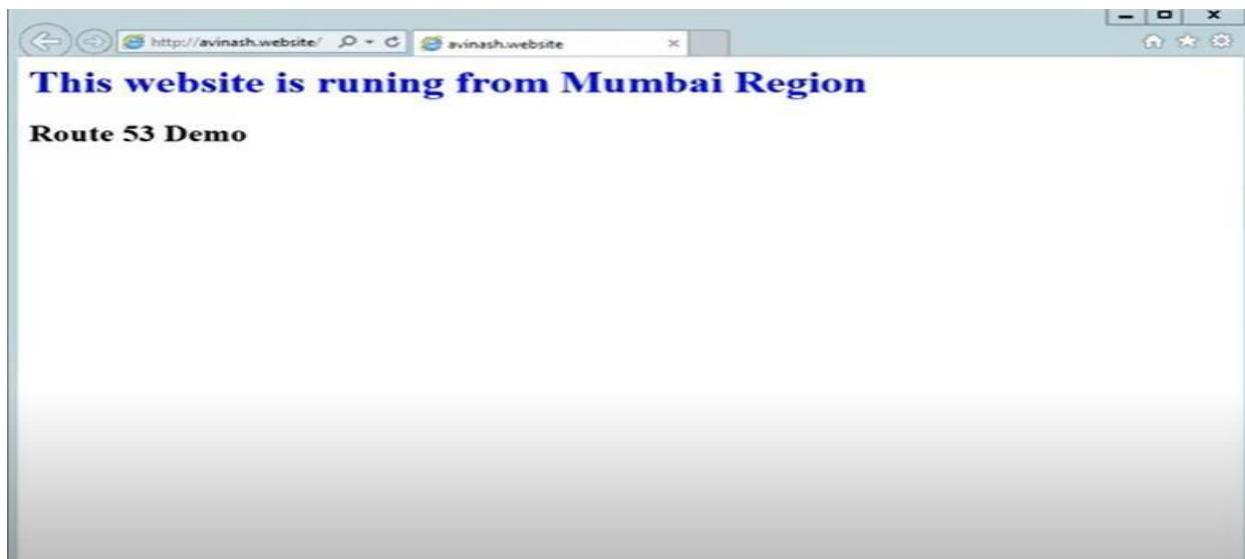
8.

In the hosted zones, create a record set and give the required information with routing policy as failover and click on create.

9.

Repeat the same steps for the secondary set ID.

10.



As it is set as primary set ID.

11.



When the load on primary set ID increases it routes the traffic to secondary set ID.

10. Optimize AWS Resource Utilization

AIM: To optimize AWS resource utilization with the help of AWS Cost Explorer.

PROCEDURE:

1. Firstly, open the AWS console homepage on browser (<https://aws.amazon.com/console/>).

The screenshot shows the AWS Console Home page. At the top, there's a header with "Console Home" and a "Info" link. Below the header, a message box says "Introducing the new widget Recent AWS blog posts. Find it at the bottom of your Console Home." On the left, there's a "Recently visited" section with links to S3, IAM, VPC, WorkSpaces, EC2, CloudShell, Amazon Simple Email Service, and EC2 Image Builder. To the right, there's a "Welcome to AWS" section with three items: "Getting started with AWS" (with a rocket icon), "Training and certification" (with a person icon), and "What's new with AWS?" (with a lightbulb icon). At the bottom, there are links for "AWS Health" and "Cost and usage". The footer includes "Feedback", "Unified Settings", "© 2022, Amazon Internet Services Private Ltd. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

2. Go to AWS Cost Management Console.

The screenshot shows the AWS Cost Management console. The left sidebar has a "Home" button and links for "Cost Explorer", "Reports", "Budgets", "Cost Anomaly Detection", and "Rightsizing recommendations". It also has sections for "Savings Plans" (Overview, Inventory, Recommendations, Purchase Savings Plans, Utilization report, Coverage report, Cart 0) and "Reservations" (Overview, Recommendations, Utilization report). A central panel says "Welcome to AWS Cost Management" and "Since this is your first visit, it will take some time to prepare your cost and usage data. Please check back in 24 hours." The footer includes "Feedback", "Unified Settings", "© 2022, Amazon Internet Services Private Ltd. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

3. In the navigation pane, choose Rightsizing Recommendations.

Potential resource savings: \$110 (Monthly savings based on 3 resources)

Potential reservation savings: \$1,019 (Monthly savings based on 10 reservations)

Resource optimization recommendations (Last updated: 2019-07-10 2:05PM):

- EC2 rightsizing opportunities found:** 3 instances found. Taking action could save you an estimated \$110 monthly.
- Idle instances detected:** 0 idle instances detected. Terminating these instances could save you an estimated \$0 monthly.
- Underutilized instances detected:** 3 underutilized instances detected. Modifying these instances could save you an estimated \$110 monthly.

Reservation purchase recommendations:

Service	Purchase recommendations	Estimated monthly savings
EC2	7	\$421.05
RDS	3	\$597.89

4. Click on “View” to view each recommendation.

3 Optimization opportunities \$110 Estimated monthly savings 50.00% Estimated savings (%)

Based on the last 14 days, we have identified 3 instances that have been idle and underutilized. Taking action on these instances could help you save an estimated \$110 monthly (50.00% of the EC2 On-Demand instance costs associated with these instances).

Recommendation	Instance ID	Account ID	Tag(s)	CPU (%)	Monthly estimated savings	
Modify instance	i-0b18d304a1...	AWS Insights Demo...	3	6.6%	\$72	<button>View</button>
Modify instance	i-0196e32825...	AWS Insights Demo...	2	4.0%	\$33	<button>View</button>
Modify instance	i-0a9909f442...	AWS Insights Demo...	2	7.5%	\$4	<button>View</button>

< Viewing 1 to 3 of 3 recommendations >

*Estimated Annual Savings and Purchase Recommendations are based on your past usage history and the relevant EC2, RDS, ElastiCache, Redshift, or Elasticsearch pricing. If your usage patterns change, it may affect the accuracy of the estimates and the purchase recommendations.
**To maximize savings, On-Demand usage associated with instance families eligible for size flexible PIs is auto-detected, analyzed, and shown as a purchase recommendation for the smallest instance size available in that instance family. [Learn More](#)
***Please note that for RDS recommendations with MySQL and/or Oracle Database Engines, Cost Explorer will display the associated cost and usage inclusive of all database editions and/or license models for that Database Engine.

5. At the top of “Rightsizing Recommendations” page, you can filter your recommendations such as Idle instances, underutilized instances etc.

Underutilized Amazon EC2 instance details

Details							
Account name	AWS Insights Demo	Account ID	i-0b18d304	Instance type	m4.xlarge	Region	US East (N. Virginia)
Tag(s) 3							

Utilization ?							
CPU	Disk	Memory	Network capacity ?	RI hours	On-Demand hours	Total running hours ?	
7%	-	-	High	0	332	332	

Recommended action

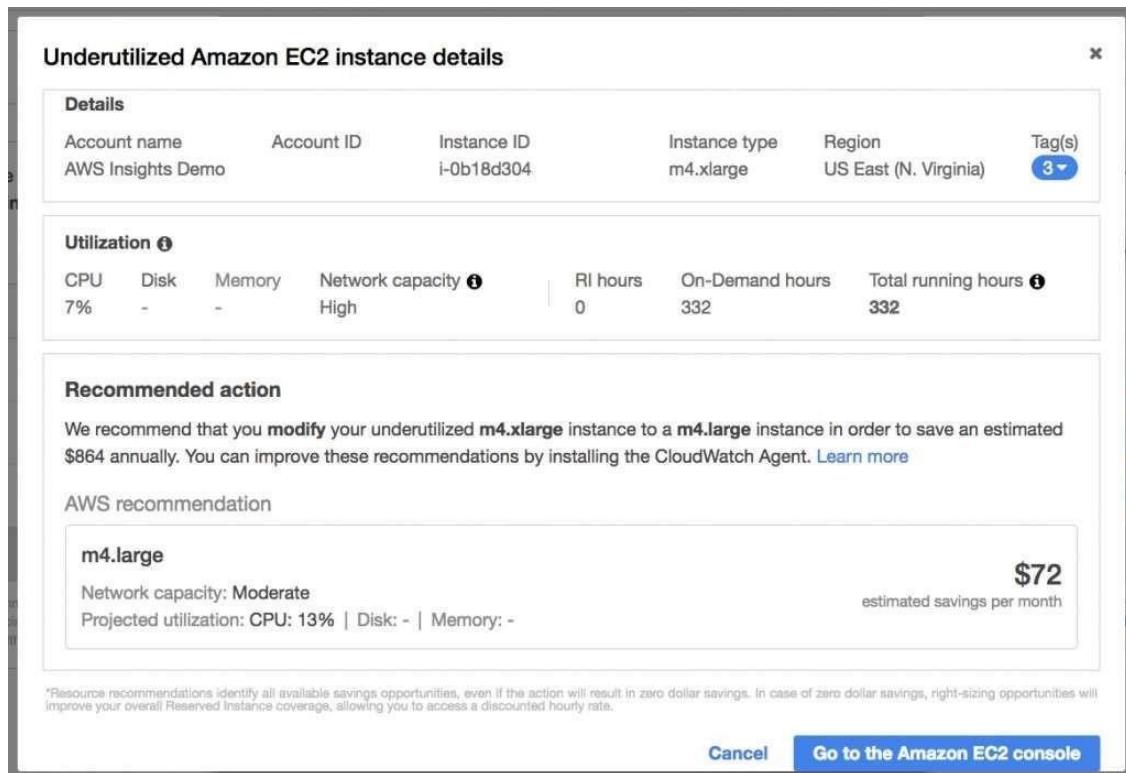
We recommend that you **modify** your underutilized **m4.xlarge** instance to a **m4.large** instance in order to save an estimated \$864 annually. You can improve these recommendations by installing the CloudWatch Agent. [Learn more](#)

AWS recommendation

m4.large	\$72
Network capacity: Moderate	estimated savings per month
Projected utilization: CPU: 13% Disk: - Memory: -	

*Resource recommendations identify all available savings opportunities, even if the action will result in zero dollar savings. In case of zero dollar savings, right-sizing opportunities will improve your overall Reserved Instance coverage, allowing you to access a discounted hourly rate.

[Cancel](#) [Go to the Amazon EC2 console](#)



RESULT:

Different AWS resource utilization was viewed and optimized with the help of AWS Cost Explorer.