

**DEPARTMENT OF NETWORKING AND
COMMUNICATIONS**

LAB RECORD

Academic Year: 2022-23

ODD SEMESTER

Programme (UG/PG) : UG

Semester : V

Course Code : 18CSC310J

**Course Title: DATA CENTRIC NETWORKING AND
SYSTEM DESIGN**

Student Name :

Register Number :

Branch with Specialization : CSE- CLOUD COMPUTING

Section : K2



**SCHOOL OF COMPUTING
FACULTY OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

SRM Nagar, Kattankulathur- 603203

DEPARTEMNT 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



Name- Year-3rd Semester-5th

Branch-CSE-CLOUD COMPUTING

University Register No.-

Certified that this is a Bonafide Record work done by the above student in the year 2022 - 2023

Signature of the Faculty

Signature of Head of the Department

DATE: _____

INTERNAL EXAMINER

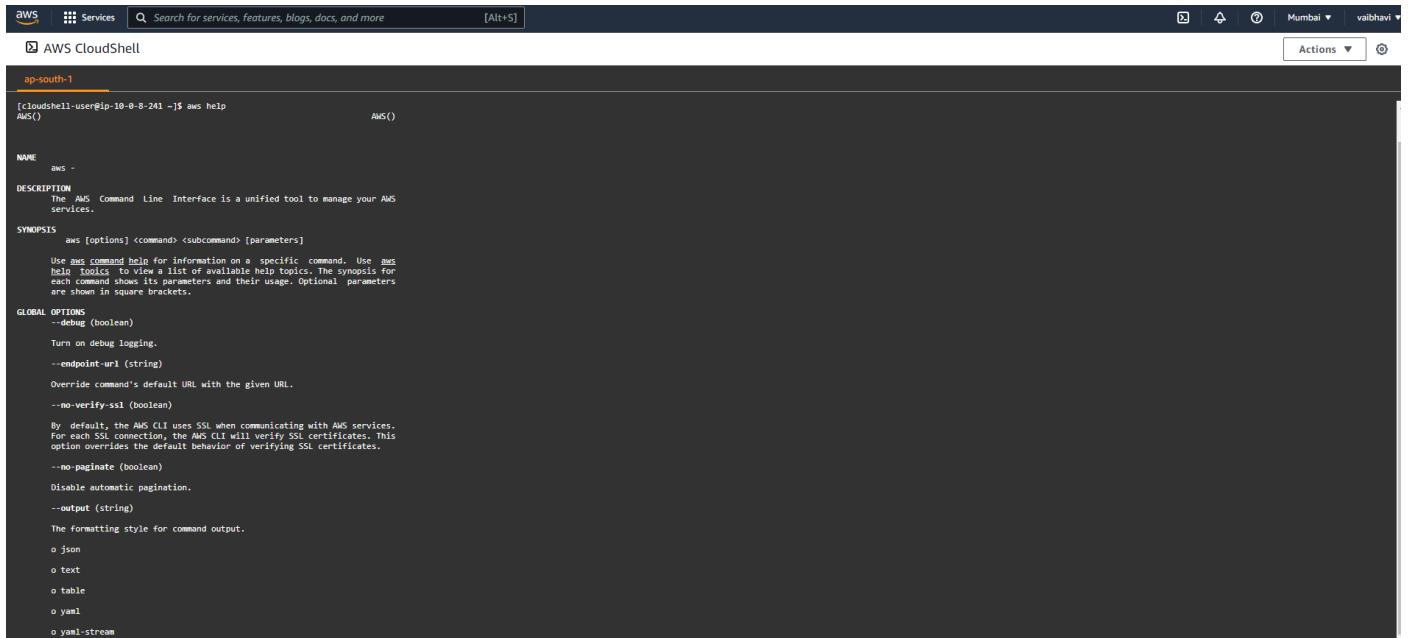
EXTERNAL EXAMINER

Table of Contents

DCN EX1.

Aim: Install and learn using AWS COMMANDS

AWS HELP



```
[cloudshell-user@ip-10-0-8-241 ~]$ aws help
aws()

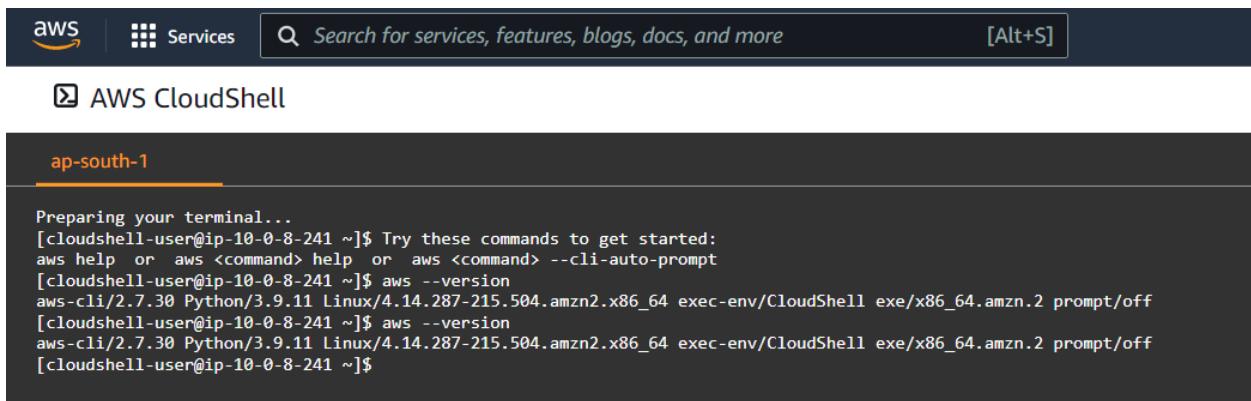
NAME      aws -
DESCRIPTION
The AWS Command Line Interface is a unified tool to manage your AWS services.

SYNOPSIS
aws [options] <command> [<subcommand>] [parameters]

Use aws --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.

GLOBAL OPTIONS
--debug (boolean)
    Turn on debug logging.
--endpoint-url (string)
    Override command's default URL with the given URL.
--no-validate (boolean)
    By default, the AWS CLI uses SSL when communicating with AWS services. For each SSL connection, the AWS CLI will verify SSL certificates. This option overrides the default behavior of verifying SSL certificates.
--no-paginate (boolean)
    Disable automatic pagination.
--output (string)
    The formatting style for command output.
    o json
    o text
    o table
    o yaml
    o yaml-stream
```

Aws—version



```
Preparing your terminal...
[cloudshell-user@ip-10-0-8-241 ~]$ Try these commands to get started:
aws help or aws <command> help or aws <command> --cli-auto-prompt
[cloudshell-user@ip-10-0-8-241 ~]$ aws --version
aws-cli/2.7.30 Python/3.9.11 Linux/4.14.287-215.504.amzn2.x86_64 exec-env/CloudShell exe/x86_64.amzn.2 prompt/off
[cloudshell-user@ip-10-0-8-241 ~]$ aws --version
aws-cli/2.7.30 Python/3.9.11 Linux/4.14.287-215.504.amzn2.x86_64 exec-env/CloudShell exe/x86_64.amzn.2 prompt/off
[cloudshell-user@ip-10-0-8-241 ~]$
```

Aws configure

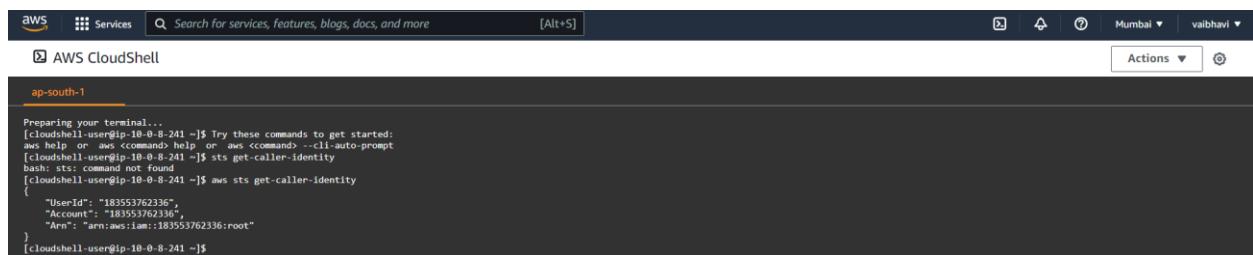


AWS CloudShell

```
ap-south-1

Preparing your terminal...
[cloudshell-user@ip-10-0-8-241 ~]$ Try these commands to get started:
aws help or aws <command> help or aws <command> --cli-auto-prompt
[cloudshell-user@ip-10-0-8-241 ~]$ aws configure
AWS Access Key ID [None]: AKIASVPFIEQDV3555HU
AWS Secret Access Key [None]: WRU3aR/vSGABwE3FdWGL2HNTPFi0/UJiJKorNvhU
Default region name [None]: mumbai
Default output format [None]: json
[cloudshell-user@ip-10-0-8-241 ~]$
```

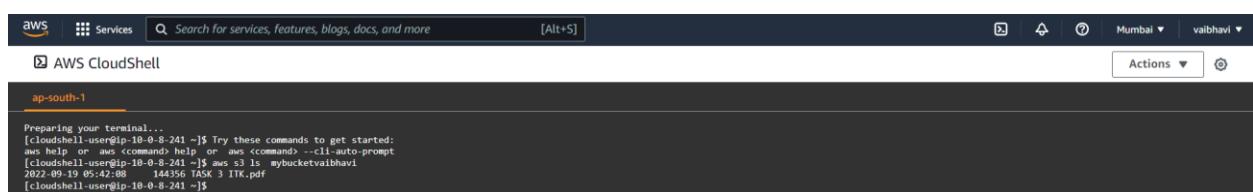
aws sts get-caller-identity



aws s3 ls



aws s3 ls bucketName



remove file from bucket



Delete bucket



The screenshot shows the AWS CloudShell interface. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar containing 'Search for services, features, blogs, docs, and more [Alt+S]', and account information for 'Mumbai' and 'valbhavi'. Below the navigation bar is a toolbar with icons for copy, paste, refresh, and actions, along with a dropdown menu labeled 'Actions ▾'. The main area is titled 'ap-south-1' and contains a terminal window. The terminal output shows the execution of AWS CLI commands to delete an S3 bucket:

```
[cloudshell-user@ip-10-0-8-241 ~]$ aws s3 rb s3://mybucketvalbhavi --force
delete: s3://mybucketvalbhavi/TASK 3 ITK.pdf
remove bucket: mybucketvalbhavi
[cloudshell-user@ip-10-0-8-241 ~]$ aws s3 ls
[cloudshell-user@ip-10-0-8-241 ~]$
```

EXPERIMENT 2

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

A] Creating Amazon EC2 Instance

Procedure :

Steps to get the Amazon AWS access key ID and secret key

1. Go to the IAM Console and click on Users.
2. Click on the User that you want to create the access key for.
Click on the actual row and not the check box.
3. On the next screen, click on "Create Access Key". If you have created keys before you should be able to see them (but can't download them again)
4. You will see a popup that allows you to download the access ID and key. The keys can be downloaded only once so make sure you save it in a safe place. You can, however, create another key later on.
5. The downloaded CSV has both the access id and key.

Configure Amazon CLI (Command Line Interface)

1. Type in "aws configure" on the command line.
2. Enter the Access ID, key and the default region.
3. That configures the CLI. This creates a directory called .aws in home. This directory has the credentials and the config file.
4. To test the configuration we will create a security group and then delete it from the AWS console. To create the security group type in

```
aws ec2 create-security-group --group-name my-sg --description "My security group"
```

This will create a new security group. Logon to AWS console to double check if you can see the security group (under EC2). You can then delete the group.

create AWS EC2 instance using CLI

We now finally look at how to create the EC2 instance using CLI. The CLI command for creating instance is called `run-instances`. When you create an instance from the console, you go through seven steps of configuration. All of that can be done using specific parameters on the CLI. While creating the instance we want to be able to select the AMI (machine image); select the instance type (hardware); set the VPC, IAM role, and other configuration parameters; configure additional block storage; add tags; add security groups and then launch one or multiple instances.

```
aws ec2 create-security-group --group-name EC2SecurityGroup --description "Security Group for EC2 instances to allow port 22"
```

```
aws ec2 authorize-security-group-ingress --group-name EC2SecurityGroup - protocol tcp --port 22 --cidr 0.0.0.0/0 aws ec2 describe-security-groups -- group-names EC2SecurityGroup
```

Commands for EC2 :

1) AWS Configure

```
C:\Users\Admin>aws configure
AWS Access Key ID [*****FL4K]: AKIA2TKXESQCM3L3OVFF
AWS Secret Access Key [*****HQE0]: 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
Default region name [None]: us-east-1
Default output format [None]: json
```

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 create-key-pair --key-name demokeypair --query 'KeyMaterial' --output text >demokeypair.pem
```

2) Create – key value - pair

3) Describe key pair

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 describe-key-pairs --key-name demokeypair
{
    "KeyPairs": [
        {
            "KeyPairId": "key-098bf943f116ba623",
            "KeyFingerprint": "cd:f2:d6:4e:cb:97:8f:d4:19:c2:22:a3:65:fd:b6:8f:ce:ed:37:3e",
            "KeyName": "demokeypair",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2022-08-26T18:24:31+00:00"
        }
    ]
}
```

4) Create security group

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 create-security-group --group-name demo-sg --description "Demo CLI Security Group" --vpc-id vpc-042b52670a9bad
{
    "GroupId": "sg-0f94d19f489ddf564"
}
```

5) Authorize security group

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 authorize-security-group-ingress --group-id sg-0f94d19f489ddf564 --protocol tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0c14c2eacd6e6a450",
            "GroupId": "sg-0f94d19f489ddf564",
            "GroupOwnerId": "728716579844",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

6) Create EC2

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 run-instances --image-id ami-05fa00d4c63e32376 --instance-type t2.micro --subnet-id subnet-0e6f00f98e52d0971 --count 1 --security-group-ids sg-0f94d19f489ddf564 --key-name demokeypair
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-05fa00d4c63e32376",
            "InstanceId": "i-003b25d102610cc17",
            "InstanceType": "t2.micro",
            "KeyName": "demokeypair",
            "LaunchTime": "2022-08-26T19:07:22+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1c",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-94-42.ec2.internal",
            "PrivateIpAddress": "172.31.94.42",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-0e6f00f98e52d0971",
            "VpcId": "vpc-042b52670a9bac0cb",
-- More -- |
```

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 run-instances --image-id ami-05fa00d4c63e32376 --instance-type t2.micro --subnet-id subnet-0e6f00f98e52d0971 --count 1 --security-group-ids sg-0f94d19f489ddf564 --key-name demokeypair
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-05fa00d4c63e32376",
            "InstanceId": "i-003b25d102610cc17",
            "InstanceType": "t2.micro",
            "KeyName": "demokeypair",
            "LaunchTime": "2022-08-26T19:07:22+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1c",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-94-42.ec2.internal",
            "PrivateIpAddress": "172.31.94.42",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-0e6f00f98e52d0971",
            "VpcId": "vpc-042b52670a9bac0cb",
-- More -- |
```

7) Describe Instance EC2

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 describe-instances --instance-ids i-003b25d102610cc17
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-05fa00d4c63e32376",
                    "InstanceId": "i-003b25d102610cc17",
                    "InstanceType": "t2.micro",
                    "KeyName": "demokeypair",
                    "LaunchTime": "2022-08-26T19:07:22+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1c",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-172-31-94-42.ec2.internal",
                    "PrivateIpAddress": "172.31.94.42",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-3-87-190-140.compute-1.amazonaws.com",
                    "PublicIpAddress": "3.87.190.140",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    }
                }
            ]
        }
    ]
}
-- More -- |
```

8) Terminate Instances

```
D:\clg_\sem 5\DCN\lab\ss\exp2>aws ec2 terminate-instances --instance-ids i-003b25d102610cc17
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-003b25d102610cc17",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

B] Creating a LAMP Instance in the AWS CLI

Procedure :

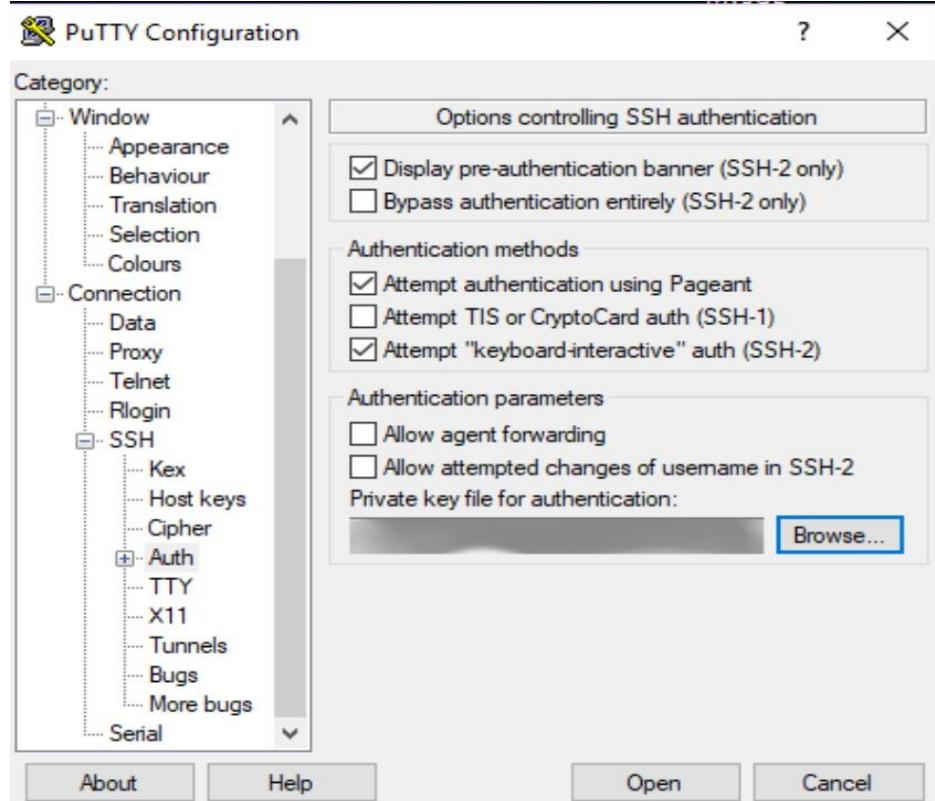
We will add four total rules. So go ahead and add three other rules. Set the types for all of the rules to:

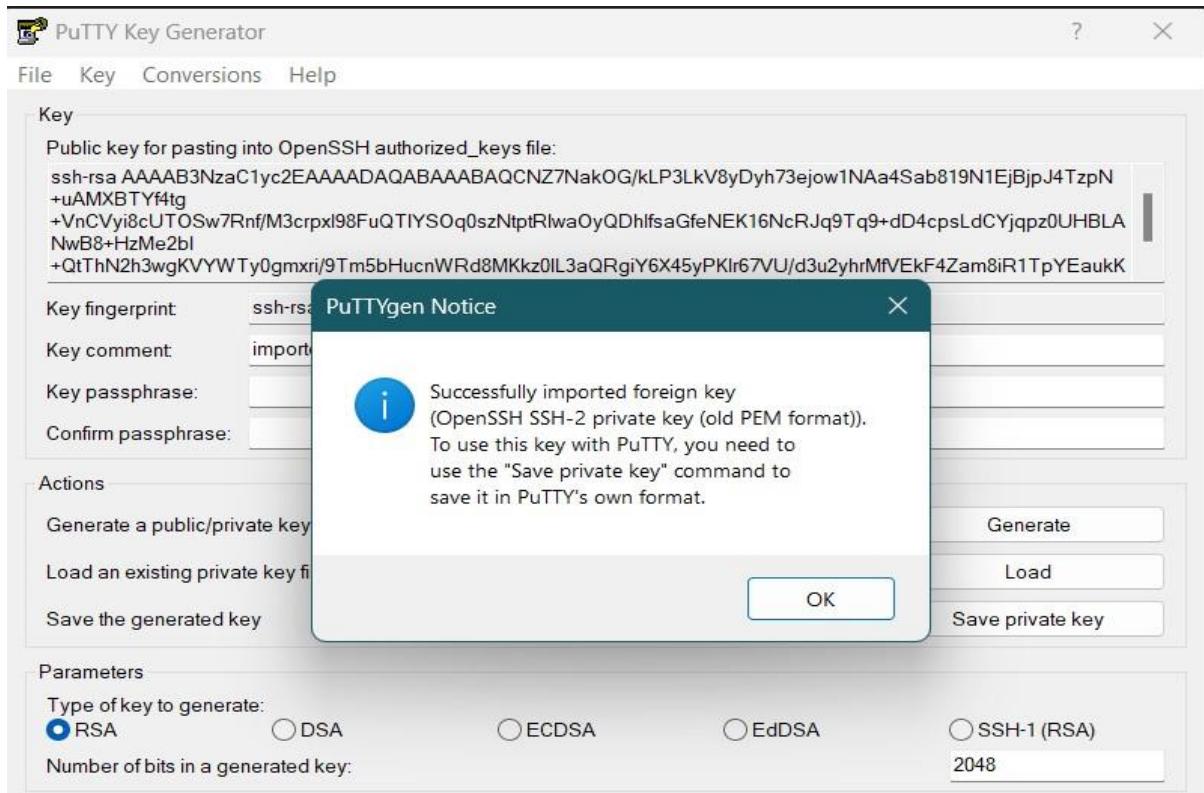
- SSH
- MySQL/Aurora
- HTTP
- HTTPS

If you're on Windows, download the latest version of PuTTY to save yourself some time and heartache. If you're using MacOS, you're more than welcome to follow along, but you can SSH into your instance from the native command line.

1. Hit the windows Key, and search for PuTTYgen. Run this program and hit "Load".
2. Search for the Private Key File you just downloaded.
3. Click "Save Private Key", give it a name, and save it in a secure location.
4. You do not need a username and password attached to the file.
5. Now, open up PuTTY, and on the left hand pane, expand the tab "SSH", and then click on "Auth".
6. Set the "Private key file for authentication" to the private key you saved from PuTTYgen.
7. Scroll back up to "Session", and enter the host name for the new instance. Just copy and paste the IP address that we allocated with Elastic IP, enter a name in the "Saved Sessions" box, and click save, so that you can SSH in with just two clicks!
8. You can now double click the saved session, just confirm the next dialog box, and the command line will now prompt you for a user, type "ec2user" and hit enter

Verification :





```

root@ip-172-31-32-239:/home/ec2-user
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Sat Aug 27 04:29:33 2022 from 106.208.16.252

              _\   _ )   Amazon Linux 2 AMI
             __| \__|_|_|

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[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]#

```

- Ec2 instance is connected. 10. Type the commands to intall LAMP (linux, apache, mysql, php) server.
- Firstly, type sudo su to become the root user.
- 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-openssh-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 0:4.0-3.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 12:4.2.5-79.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

```

- 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
=====
```

- 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]# 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
```

- 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-mysqlnd, trying to install php-mysqlnd-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.a
mzn2.0.2.x86_64
--> Processing Dependency: php-common(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-4
6.amzn2.0.2.x86_64
--> Package php-mysqlnd.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-mysqlnd-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.x
86_64
```

- 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]# yum search php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
=====
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-mysqlnd.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-xmllrpc.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
```

- 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]#
```

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

```

root@ip-172-31-32-239:/var/www/html
[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
[green bar]
~
```

PHP Version 5.4.16	
System	Linux ip-172-31-32-239 ap-south-1.compute.internal 5.10.130-118.517.amzn2.x86_64 #1 SMP Wed Jul 13 16:51:52 UTC 2022 x86_64
Build Date	Oct 31 2019 18:35:17
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/mysqlind.ini, /etc/php.d/mysqlnd_myisam.ini, /etc/php.d/mysqlnd_myisqldini, /etc/php.d/pdo_mysqlind.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/sqlite3.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API20100525.NTS
PHP Extension Build	API20100525.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled

Result:

LAMP server is successfully created using 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 Console Home page. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar containing 'Search for services, features, blogs, docs, and more', and a keyboard shortcut '[Alt+S]'. On the right, it shows the region 'Mumbai' and a user 'vaibhavi'. Below the navigation bar, the main content area has two sections: 'Recently visited' on the left and 'Welcome to AWS' on the right. The 'Recently visited' section lists links to VPC, CloudShell, S3, IAM, EC2, and AWS Budgets. The 'Welcome to AWS' section features three cards: 'Getting started with AWS' (with a rocket icon), 'Training and certification' (with a diploma icon), and 'What's new with AWS?' (with a lightbulb icon). At the bottom of the main content area, there's a 'View all services' link.

2. Search for VPC in the search bar and open the VPC dashboard page.

The screenshot shows the VPC Dashboard page. The left sidebar includes a 'New VPC Experience' section, a 'VPC dashboard' section with an 'EC2 Global View' link, and a 'Virtual private cloud' section listing 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'DHCP Option Sets', 'Elastic IPs', 'Managed prefix lists', 'Endpoints', 'Endpoint services', 'NAT gateways', and 'Peering connections'. The main content area has several sections: 'Create VPC' and 'Launch EC2 Instances' buttons at the top; a 'Resources by Region' table showing counts for VPCs (2), Subnets (7), Route Tables (5), Internet Gateways (2), Egress-only Internet Gateways (0), DHCP option sets (1), and Elastic IPs (0); a 'Service Health' section showing 'Amazon EC2 - Asia Pacific' is operating normally; a 'Settings' section with 'Zones' and 'Console Experiments'; an 'Additional Information' section with links to 'VPC Documentation', 'All VPC Resources', 'Forums', and 'Report an Issue'; an 'AWS Network Manager' section with a brief description and a 'Get started with Network Manager' link; and a 'Site-to-Site VPN Connections' section.

3. To create a new VPC, click “Create VPC” and go to the Create VPC page

The screenshot shows the AWS VPC "Create VPC" page. On the left, under "VPC settings", there are three radio button options: "VPC only", "VPC and more" (which is selected), and "VPC with FlowLogs". Below these are sections for "Name tag auto-generation" (with "Auto-generate" checked and "project" selected) and "IPv4 CIDR block" (set to "10.0.0.16 / 16"). On the right, the "Preview" section shows a hierarchical network structure. A central "VPC" node labeled "project-vpc" branches down to four "Subnets (4)" under "ap-south-1a" and "ap-south-1b". These subnets further branch to "project-subnet-public1-ap-south-1a", "project-subnet-private1-ap-south-1a", "project-subnet-public2-ap-south-1b", and "project-subnet-private2-ap-south-1b". Finally, these subnets connect to three "Route tables (3)" labeled "project-rtb-public", "project-rtb-private1-ap-south-1a", and "project-rtb-private2-ap-south-1b". A tooltip at the top right says: "Introducing the new create VPC experience. We've designed the new create VPC to make it easier to use. The changes include a new visualization of the resources that will be created. Let us know what you think."

4. Select the following configurations with appropriate VPC name

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

The screenshot shows the "Create VPC workflow" page after the VPC has been created. It displays a success message: "Creating VPC Resources" and "Thank you for using the new create VPC experience. Let us know what you think." Below this, the "Success" section lists the completed steps: "Create VPC: vpc-00dd3bbf5601bcac3", "Enable DNS hostnames", "Enable DNS resolution", "Verifying VPC creation: vpc-00dd3bbf5601bcac3", "Create S3 endpoint: vpce-015cafcbe3af6ae5", "Create subnet: subnet-0b244c5aa7663a665", "Create subnet: subnet-0598bd280bddfad65", "Create subnet: subnet-0780c2e6f13dd1cb0", "Create subnet: subnet-08028b27313ba4a525", "Create internet gateway: igw-0cdefc3ef77e39793", and "Attach internet gateway to the VPC".

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

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

Subnets (7) Info

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available
—	subnet-00e29325ec87b6c35	Available	vpc-0dfd47eb6b9f4fd3d	172.31.32.0/20	—	4090
project-test1-subne...	subnet-0b244c5aa7663a665	Available	vpc-00dd3bbf5601bcac3 pro...	10.0.0.0/20	—	4091
—	subnet-02b132e3475520db	Available	vpc-0dfd47eb6b9f4fd3d	172.31.0.0/20	—	4090
project-test1-subne...	subnet-0780c2e6f13dd1cb0	Available	vpc-00dd3bbf5601bcac3 pro...	10.0.128.0/20	—	4091
project-test1-subne...	subnet-0598bd280bddfd65	Available	vpc-00dd3bbf5601bcac3 pro...	10.0.16.0/20	—	4091
project-test1-subne...	subnet-08028b27313ba4525	Available	vpc-00dd3bbf5601bcac3 pro...	10.0.144.0/20	—	4091
—	subnet-071a6c369259a82fb	Available	vpc-0dfd47eb6b9f4fd3d	172.31.16.0/20	—	4091

Select a subnet

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

Route tables (5) Info

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
—	rtb-04fce30dcc1f7de8	—	—	Yes	vpc-0dfd47eb6b9f4fd3d	183553762336
project-test1-rtb-pr...	rtb-0fa5346c6c57bdabf	subnet-08028b27313ba...	—	No	vpc-00dd3bbf5601bcac3 pro...	183553762336
—	rtb-028777be449a183c1	—	—	Yes	vpc-00dd3bbf5601bcac3 pro...	183553762336
project-test1-rtb-pr...	rtb-05c66d80317c087ee	subnet-0780c2e6f13dd...	—	No	vpc-00dd3bbf5601bcac3 pro...	183553762336
project-test1-rtb-p...	rtb-095d3090b287a4767	2 subnets	—	No	vpc-00dd3bbf5601bcac3 pro...	183553762336

Select a route table

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

The screenshot shows the AWS Management Console with the "Services" bar at the top. The "Network ACLs" service is selected under the "Security" category in the left sidebar. The main pane displays a table titled "Network ACLs (2) Info" with the following data:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count
-	acl-0119975ef1f4295dd	4 Subnets	Yes	vpc-00dd3bbf5601bcac3 / project-test1-vpc	2 Inbound rules
-	acl-0c01a8b4a6282ed63	3 Subnets	Yes	vpc-0dfd47eb6b9f4fd3d	2 Inbound rules

Below the table, there is a section titled "Select a network ACL" with three small icons.

The screenshot shows the AWS Management Console with the "Services" bar at the top. The "Your VPCs" service is selected under the "Virtual private cloud" category in the left sidebar. The main pane displays a table titled "Your VPCs (2) Info" with the following data:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set
project-test1-vpc	vpc-00dd3bbf5601bcac3	Available	10.0.0.0/16	-	dopt-025aff884a7e20...
-	vpc-0dfd47eb6b9f4fd3d	Available	172.31.0.0/16	-	dopt-025aff884a7e20...

Below the table, there is a section titled "Select a VPC above" with three small icons.

RESULT:

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

EXPERIMENT 4: CREATING A STATIC

WEBSITE ON AMAZON S3-USING 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

The screenshot shows the AWS S3 console interface. At the top, there's a blue header bar with the AWS logo, a search bar, and navigation links. Below the header, a banner says "We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback." On the left, there's a sidebar with "Storage" and "Amazon S3" sections. The main content area has a large "Create a bucket" button. To the right of the button, there's a section about object storage and a "Create bucket" button. Below the main area, there are three tabs: "Pricing", "How it works", and "Resources". The "How it works" tab is currently active, displaying a video player with the title "Introduction to Amazon S3". The "Pricing" tab contains information about no minimum fees and a link to the AWS Simple Monthly Calculator. The "Resources" tab has a "User guide" link.

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more [\[?\]](#)

General configuration

Bucket name Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming [\[?\]](#)

AWS Region

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Successfully created bucket "mybucketvaibhavi"
To upload files and folders, or to configure additional bucket settings choose [View details](#).

Follow security best practices for S3.

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 [\[?\]](#)

Name	AWS Region	Access	Creation date
mybucketvaibhavi	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	October 12, 2022, 11:18:31 (UTC+05:30)

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

AWS Services Search for services, features, blogs, docs, and more [Alt+S] Global valbhav

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.

Amazon S3 > Buckets > mybucketvalbhav > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 1600B, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more [\[?\]](#)

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

Files and folders (2 Total, 503.8 KB)
All files and folders in this table will be uploaded.

Name	Folder	Type	Size
8.jpg	-	image/jpeg	503.3 KB
index.html	-	text/html	454.0 B

Destination

Destination
s3://mybucketvalbhav

▶ Destination details
Bucket settings that impact new objects stored in the specified destination.

▶ Permissions
Grant public access and access to other AWS accounts.

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.

Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://mybucketvalbhav	2 files, 503.8 KB (100.00%)	0 files, 0 B (0%)

Files and folders (2 Total, 503.8 KB)

Name	Folder	Type	Size	Status	Error
8.jpg	-	image/jpeg	503.3 KB	Success	-
index.html	-	text/html	454.0 B	Success	-

Amazon S3

Services Search for services, features, blogs, docs, and more [Alt+S]

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Successfully edited static website hosting.

How to optimize your costs on S3.

Requester pays

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more [i]

Requester pays

Disabled

Edit

Static website hosting

Use this bucket to host a website or redirect requests. Learn more [i]

Static website hosting

Enabled

Hosting type

Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more [i]

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

Edit

AWS Marketplace for S3

Amazon S3

Services Search for services, features, blogs, docs, and more [Alt+S]

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Successfully edited bucket policy.

How to optimize your costs on S3.

Amazon S3 > Buckets > mybucketvaibhavi

mybucketvaibhavi info

Publicly accessible

Objects Properties Permissions Metrics Management Access Points

Permissions overview

Access

Public

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more [i]

Edit

Block all public access

Off

Individual Block Public Access settings for this bucket

AWS Marketplace for S3

HEY , THIS IS Vaibhavi HERE 🙋

I am pursuing CSE in CC from SRMIST



Result: Static website has been created in S3 using Amazon System Manager

Vaibhavi Tandon
RA2011028010087
Experiment : 5

Title : Automation and Optimization with Amazon S3

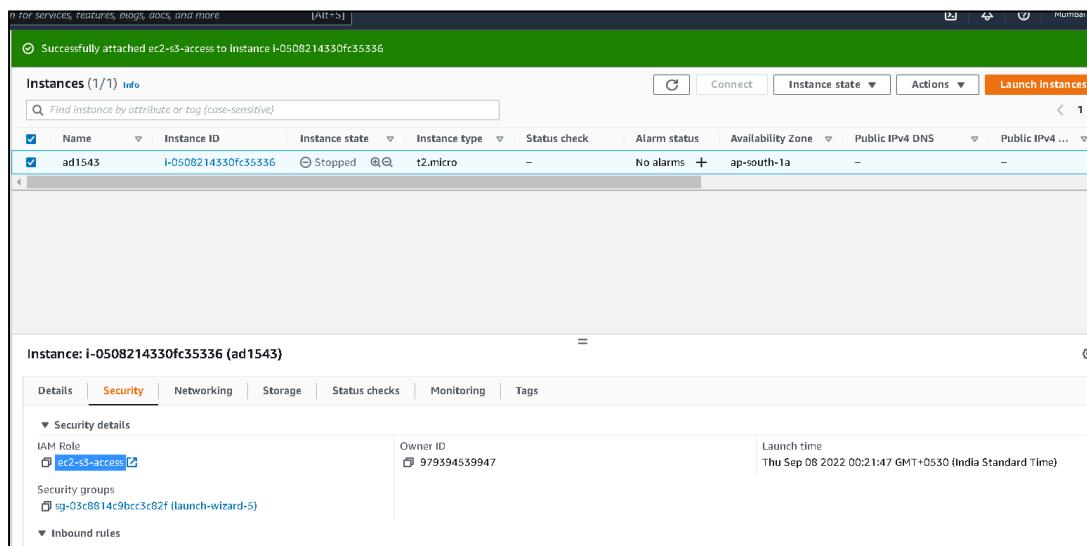
Aim : Automate Files backup to aws S3 bucket on Linux machine.

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

Procedure :

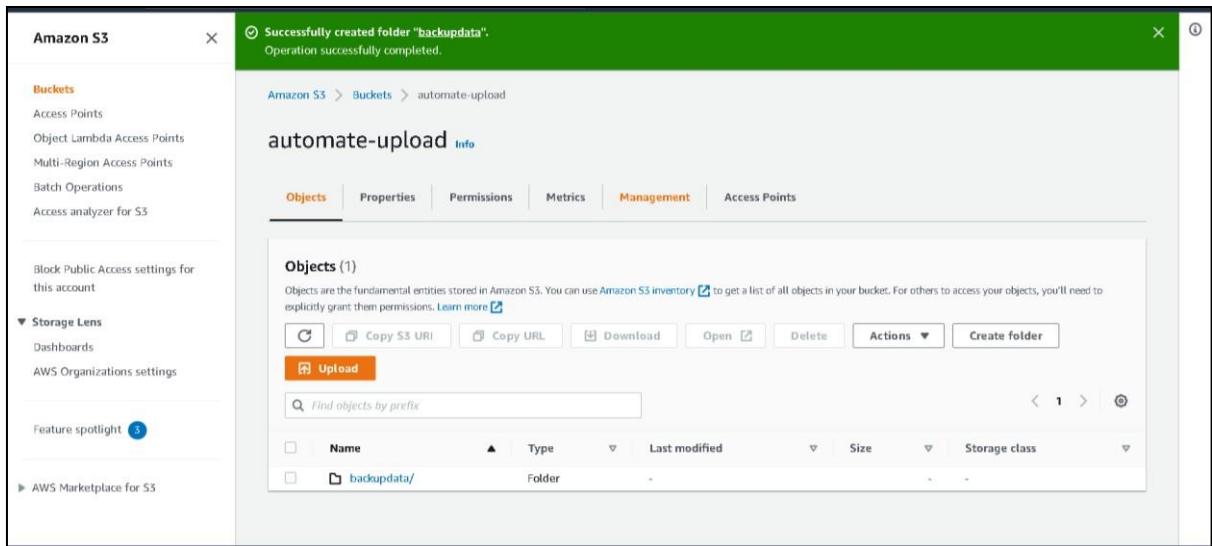
Steps:

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



(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-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
cp
rm
sb
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

Objects (3)

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

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 sync /home/ec2-user/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]# 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.

The screenshot shows the Amazon S3 console interface. On the left, the navigation pane includes 'Buckets', 'Storage Lens', and 'AWS Marketplace for S3'. The main area displays the 'automate-upload' bucket. The 'Objects' tab is active, showing 5 objects: 'a', 'b', 'c', 'd', and 'e'. Each object has a timestamp of 'September 22, 2022, 09:34:00 (UTC+05:30)' and a size of '0 B'. The 'Storage class' for all objects is 'Standard'. Action buttons like 'Upload', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', and 'Actions' are visible above the object list.

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, 09:43:05 (UTC+05:30)	0 B	Standard
e	-	September 22, 2022, 09:43:05 (UTC+05:30)	0 B	Standard

Result:

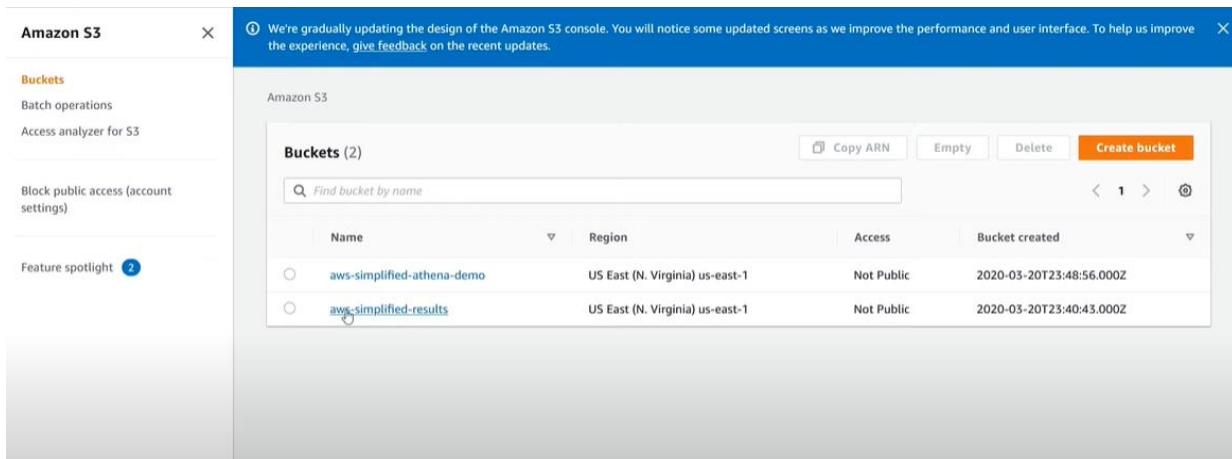
We have successfully automated our local files/directory backup on Amazon S3 buckets using crontab.

Vaibhavi Tandon
RA2011028010087

EXP NO : 6 Querying Data in S3 with Amazon Athena

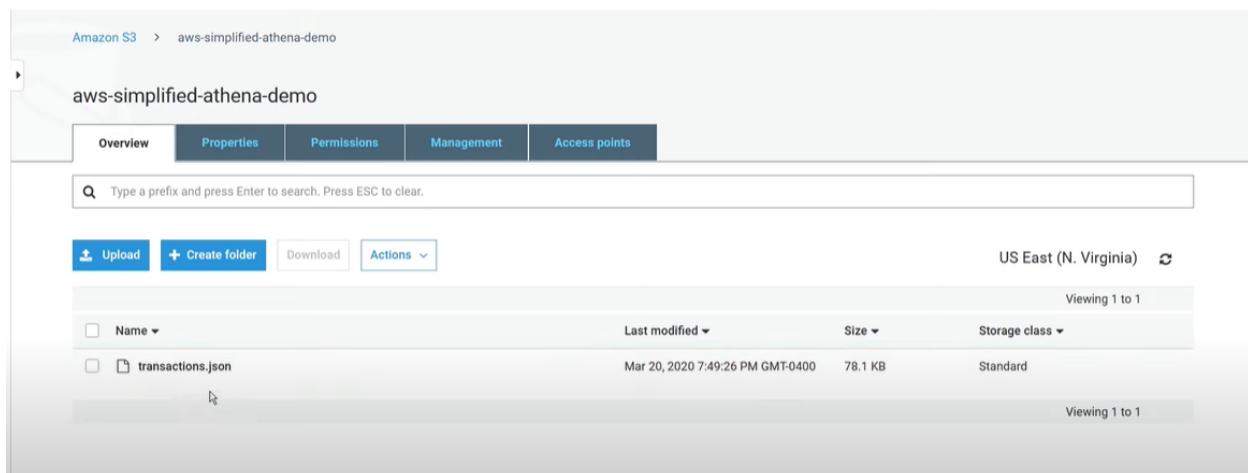
Aim: Querying Data in S3 with Amazon Athena

Step 1:
Go to buckets and create two buckets.



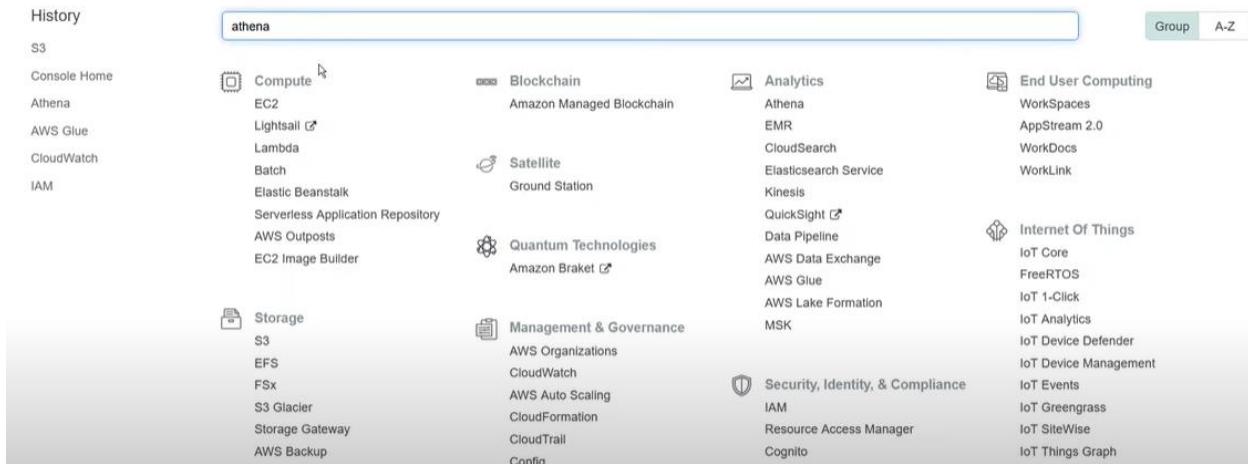
The screenshot shows the Amazon S3 console interface. On the left, there's a sidebar with links for 'Buckets', 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and a 'Feature spotlight' section. The main area is titled 'Amazon S3' and shows a table of 'Buckets (2)'. 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, Bucket created: 2020-03-20T23:48:56.000Z) and 'aws_simplified-results' (Region: US East (N. Virginia) us-east-1, Access: Not Public, Bucket 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.

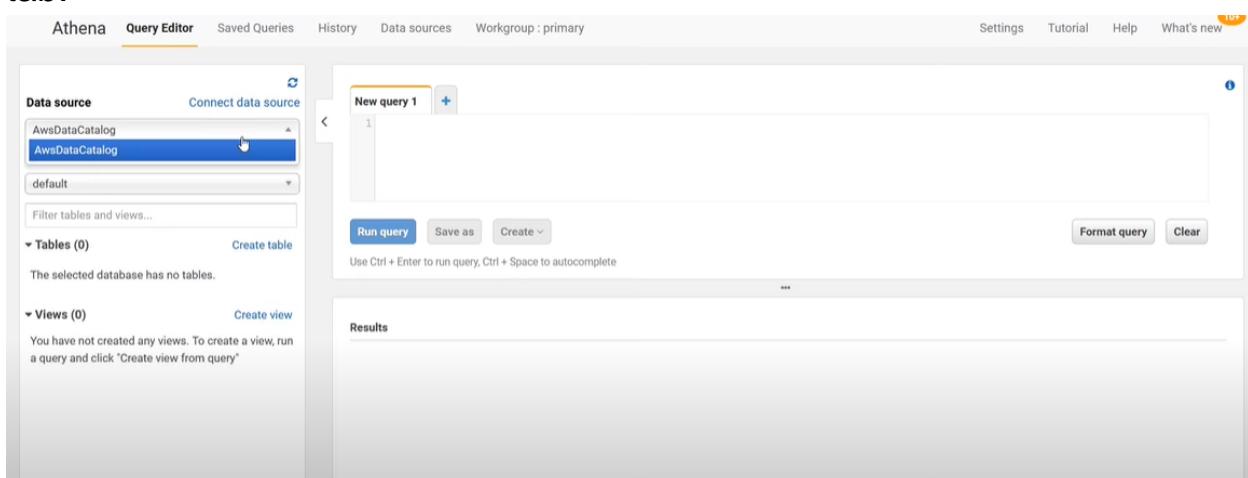


The screenshot shows the 'aws-simplified-athena-demo' bucket details. At the top, there's a breadcrumb navigation: 'Amazon S3 > aws-simplified-athena-demo'. Below it, the bucket name 'aws-simplified-athena-demo' is displayed. A navigation bar with tabs 'Overview' (selected), 'Properties', 'Permissions', 'Management', and 'Access points' is present. A search bar with placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' is below the navigation bar. Under the search bar, there are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. To the right, the region 'US East (N. Virginia)' and a note 'Viewing 1 to 1' are shown. The main content area displays a table of files. The table has columns for 'Name', 'Last modified', 'Size', and 'Storage class'. One file, 'transactions.json', is listed with the following details: Name: transactions.json, Last modified: Mar 20, 2020 7:49:26 PM GMT-0400, Size: 78.1 KB, Storage class: Standard. Another note 'Viewing 1 to 1' is located at the bottom right of the file list.

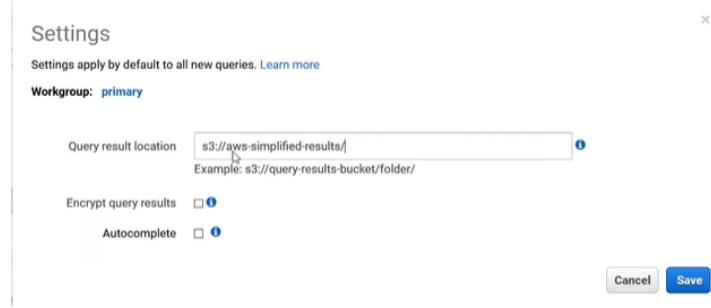
Step 3 :
Now go to Amazon athena.



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.

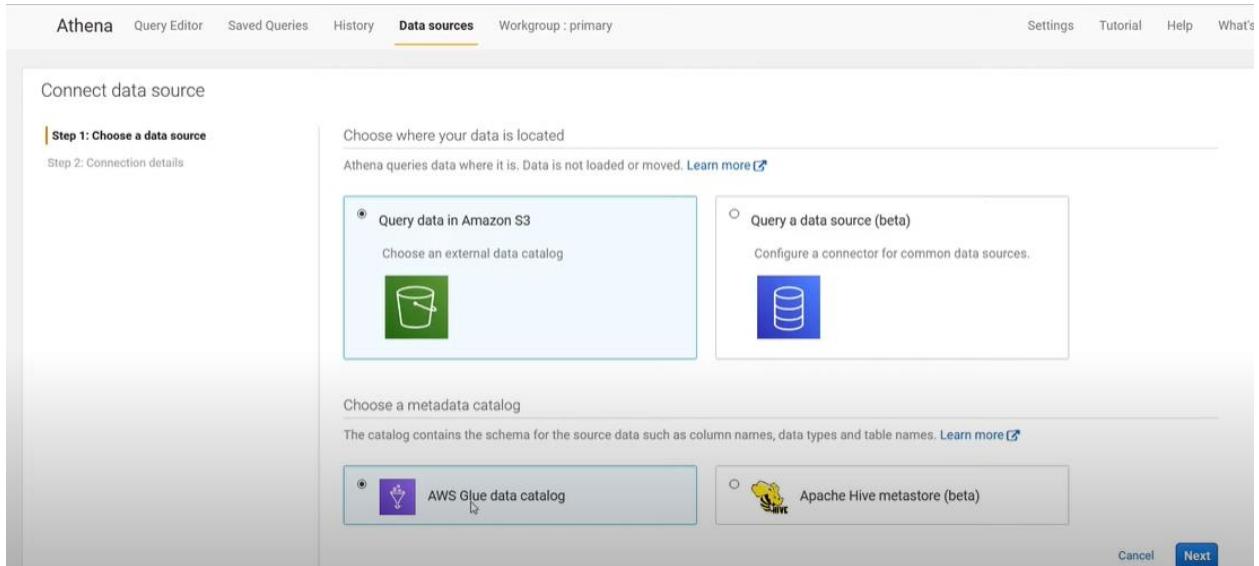


Step 6:

Click on connect data source.

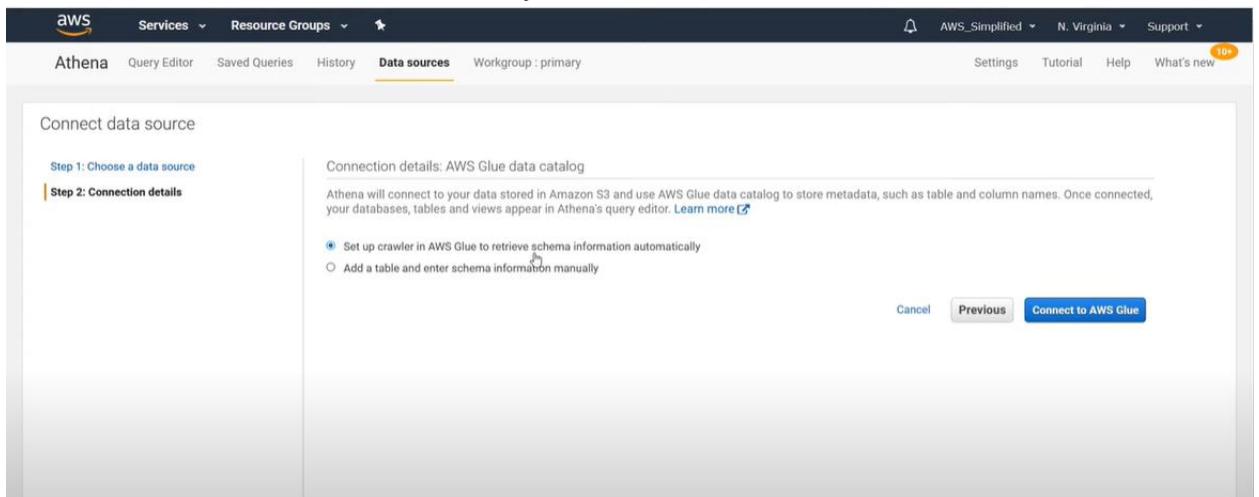
Step 7:

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

Add information about your crawler

Crawler name
aws-athena-demo

Tags, description, security configuration, and classifiers (optional)

Next

Crawler info
 Crawler source type
 Data store
 IAM Role
 Schedule
 Output
 Review all steps

Add crawler

Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type
 Data stores
 Existing catalog tables

Back Next

Crawler info
aws-athena-demo
 Crawler source type
 Data store
 IAM Role
 Schedule
 Output
 Review all steps

Add crawler

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

Crawler info
aws-athena-demo
 Crawler source type
Data stores
 Data store
 IAM Role
 Schedule
 Output
 Review all steps

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

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](#)

Create a schedule for this crawler

Frequency

[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
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.

The screenshot shows the AWS Glue console. On the left, there's a sidebar with navigation links: Data catalog, Databases, Tables, Connections, Crawlers (which is selected), Classifiers, Settings, ETL, Workflows, Jobs, ML Transforms, Triggers, Dev endpoints, and Notebooks. The main area has a title 'Crawlers' with a sub-instruction: 'A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.' Below this, a message says 'Attempting to run crawler "aws-athena-demo"...'. There are buttons for 'Add crawler' and 'Run crawler' (which is highlighted). A search bar and a 'User preferences' section are also present. A table lists one crawler: 'aws-athena-demo' (Name), 'Ready' (Status), '0 secs' (Last runtime), '0 secs' (Median runtime), '0' (Tables updated), and '0' (Tables added). The table has columns for Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added.

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.

The screenshot shows the Amazon Athena Query Editor. At the top, it says 'Athena' and 'Query Editor'. It also shows 'Saved Queries', 'History', 'Data sources', and 'Workgroup : primary'. On the right, there are 'Settings', 'Tutorial', 'Help', and 'What's new' buttons. The left sidebar shows 'Data source' set to 'AwsDataCatalog', 'Database' set to 'default', and a list of 'Tables (1)' which includes 'aws_simplified_athena_demo'. Below that is a list of 'Views (0)'. The main area has a 'New query 1' tab. A context menu is open over the 'aws_simplified_athena_demo' table, with options like 'Preview table' (which is highlighted in yellow), 'Show properties', 'Delete table', and 'Generate Create Table DDL'. Below the table list, there's a 'Results' section. At the bottom, there are buttons for 'Format query' and 'Clear'.

Step 12:

Now the query can be executed.

This screenshot is similar to the previous one but shows a query being run. The 'New query 2' tab is active, containing the SQL command: 'SELECT * FROM "default"."aws_simplified_athena_demo" WHERE type='PURCHASE' and amount >= 50;'. Below the query, it says '(Run time: 1.81 seconds, Data scanned: 78.08 KB)'. At the bottom, there are buttons for 'Run query', 'Save as', 'Create', 'Format query', and 'Clear'.

Result :

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

Vaibhavi Tandon
RA2011028010087

Experiment : 7

Title : Creating a lambda function in AWS to email daily reports

Aim : Automate Sending Emails at a Specific Time with AWS Lambda, CloudWatch and SES

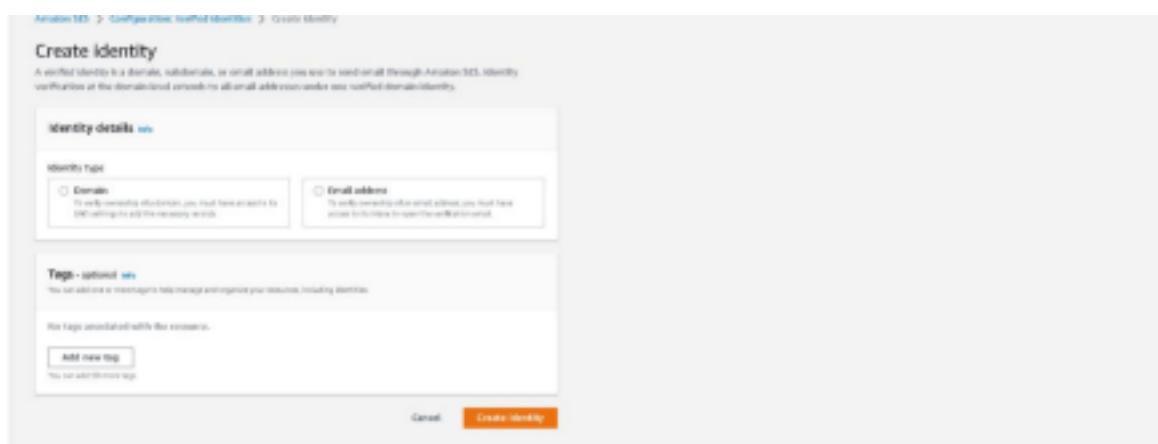
Pre-requisites : AWS Console, Amazon SES, Amazon Lambda, Amazon CloudWatch.

Procedure :

We are going automate sending email to a person or a group of people. AWS **Cloudwatch** is used to setup a schedule to trigger AWS **Lambda** function and then its going to use AWS **SES (Simple Email Service)** to send out emails to people.

Steps:

1. Go to AWS SES (Simple email service), click on “Create Identity”. Use email address as a type and type the email address.



2. Verify the email address that reviewed an email from aws to tell you to verify that.

Verified identities										
A verified identity is a domain, subdomain, or email address you want to send or call through Amazon SES. Learn more										
<p>Note: Identity status update</p> <p>The Identity Status value represents the explicit verification of the Identity itself. For the domain identities this means verifying ownership through updating the DNS records, and for the email address identities, this means updating the verification email aws-verify@amazon.com and selecting the link to complete the verification process. Learn more</p>										
<p>Verified identities (0) View</p> <p>Search all domains and email address identities</p> <table><thead><tr><th><input type="checkbox"/> Identity</th><th>Identity type</th><th><input type="checkbox"/> Identity status</th></tr></thead><tbody><tr><td><input type="checkbox"/> awsidentity001@123@gmail.com</td><td>email address</td><td><input checked="" type="checkbox"/> not yet verified</td></tr><tr><td><input type="checkbox"/> awsidentity002@gmail.com</td><td>email address</td><td><input checked="" type="checkbox"/> not yet verified</td></tr></tbody></table>		<input type="checkbox"/> Identity	Identity type	<input type="checkbox"/> Identity status	<input type="checkbox"/> awsidentity001@123@gmail.com	email address	<input checked="" type="checkbox"/> not yet verified	<input type="checkbox"/> awsidentity002@gmail.com	email address	<input checked="" type="checkbox"/> not yet verified
<input type="checkbox"/> Identity	Identity type	<input type="checkbox"/> Identity status								
<input type="checkbox"/> awsidentity001@123@gmail.com	email address	<input checked="" type="checkbox"/> not yet verified								
<input type="checkbox"/> awsidentity002@gmail.com	email address	<input checked="" type="checkbox"/> not yet verified								

3. Create two identities (email address).
One for sending emails and another for receiving.
 4. Create an IAM role.
Give Use case as lambda and give full access to cloudwatch, SES.
 5. Go to Lambda Service, create a lambda function.
Give name, runtime as NodeJS, execution role as created IAM role previously.

Basic Information

Function name:
 Give a name that describes the purpose of your function.

Handler code:
 Choose the language to use for the function's Lambda. Note that the console currently supports only Node.js, Python, and Java.

Authorizer: [Info](#)
 Choose the permission set or identity you want for your function code.

[AWS Lambda](#)

Permissions: [Info](#)
 My default Lambda automatically assumes its own permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role and when adding triggers.

Change default execution role:

Execution role: [Info](#)
 Choose the role that references permissions of this function. To create a custom role, go to the [AWS Lambda](#) console.

[Create a new role with basic Lambda permissions](#)

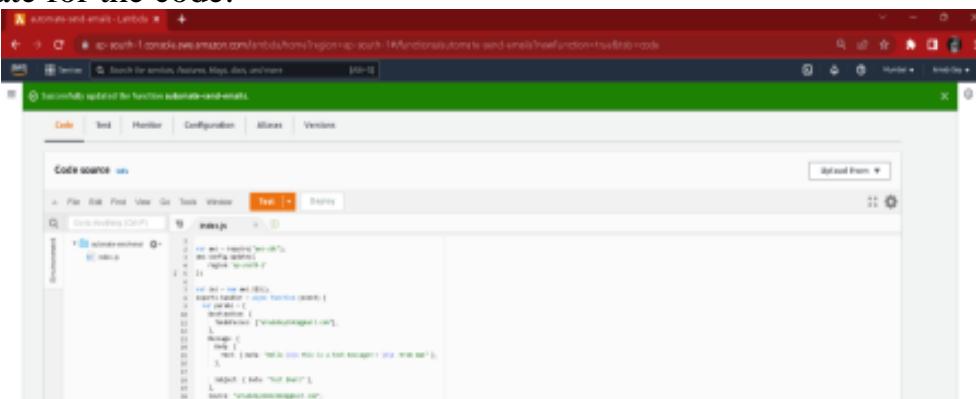
[Use an existing role](#)

[Create a new role from AWS policy templates](#)

Editing role:
 Choose the editing role that you've created to be used with this Lambda function. You must have permission to upload logs to Amazon CloudWatch Logs.

Authorizers and Lambda functions:

- ## 6. Use this template for the code:



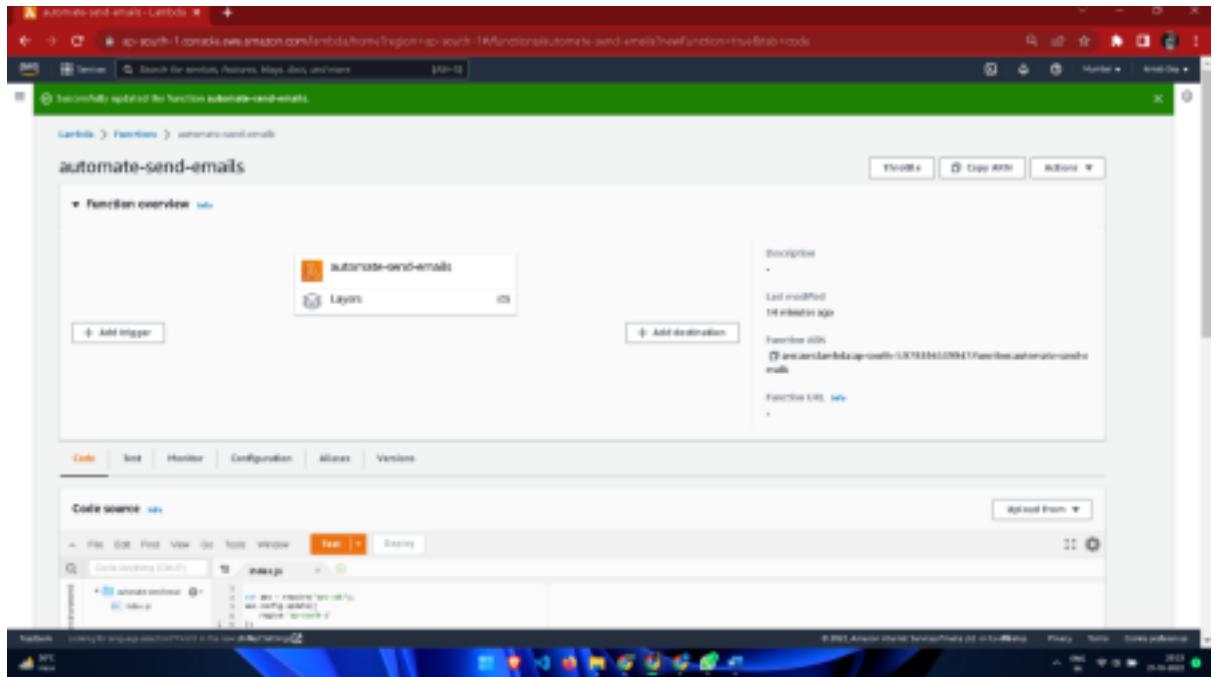
The screenshot shows the AWS Lambda console with the 'lambda-send-email' function selected. The 'Code source' tab is active, displaying a Python script named 'lambda_function.py'. The code uses the `smtplib` and `email.mime.text` modules to send an email via Gmail's SMTP server. It defines a function `lambda_handler` that takes event and context parameters. Inside, it sets up the SMTP host, port, user, password, and recipient. It creates a `MIMEText` message with the subject 'Lambda Test Email' and body 'This is a test message.'. It then connects to the SMTP server, logs in, and sends the message. Finally, it returns a success response with status code 200 and a message indicating the email was sent successfully.

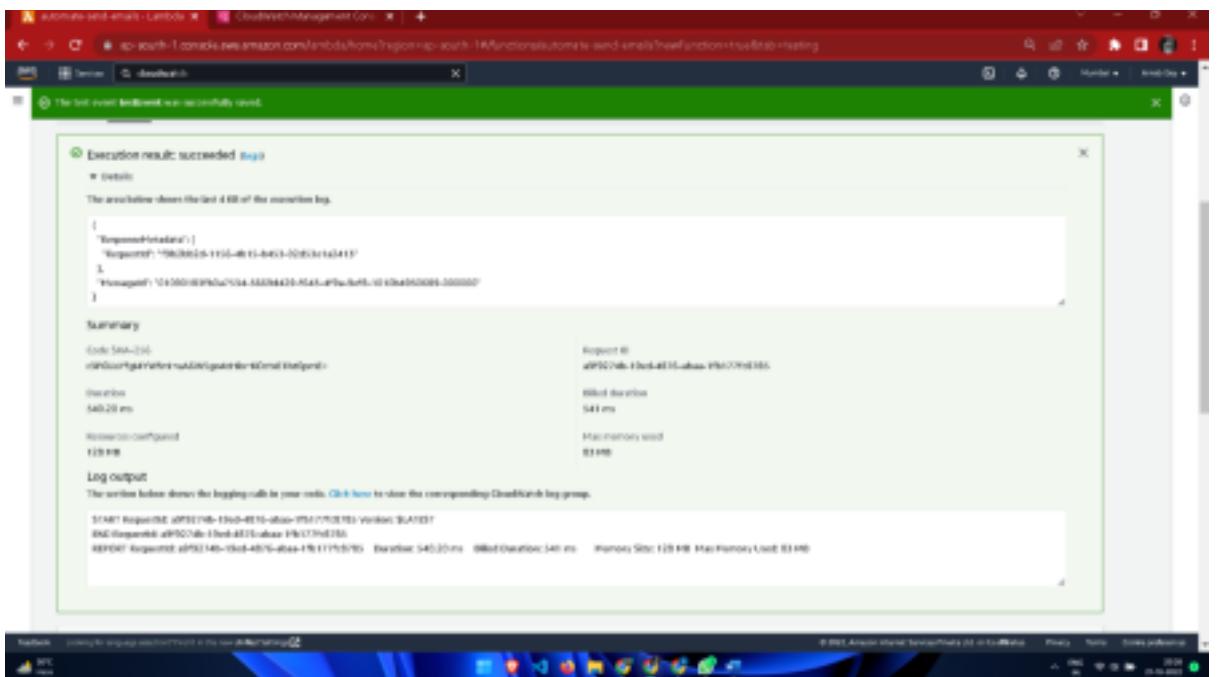
```

1)
2) var aws = require("aws-sdk");
3) var ses = new aws.SES({ region: "us-west-2" });
4) exports.handler = async function (event) {
5)   var params = {
6)     Destination: [████████],
7)     ToAddresses: ["RecipientEmailAddress"],
8)   },
9)   Message: {
10)     Body: {
11)       Text: { Data: "Test" },
12)     },
13)   },
14)   Subject: { Data: "Test Email" },
15) },
16) Source: "SourceEmailAddress",
17) };
18) ██████████
19) return ses.sendEmail(params).promise()
20) };

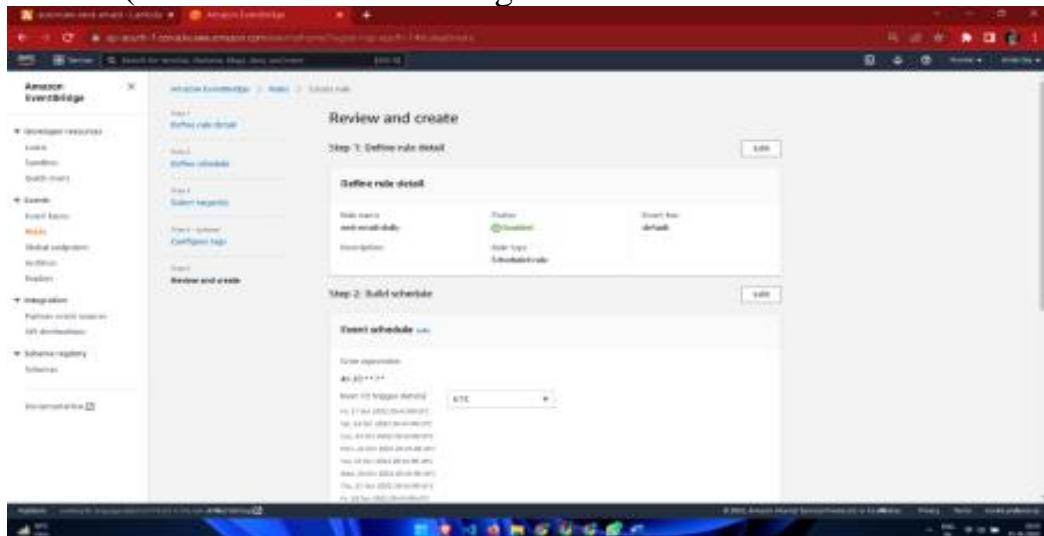
```

7. Click on Deploy and then TEST, you wil receive the message in your mentioned emails.

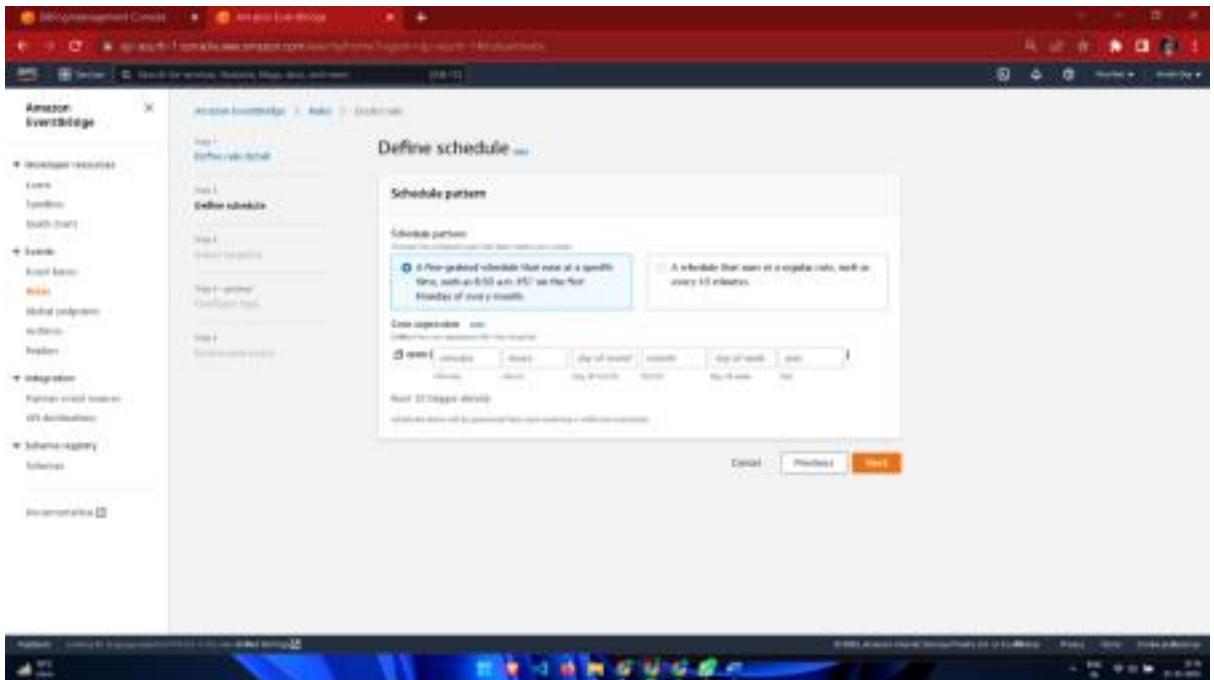




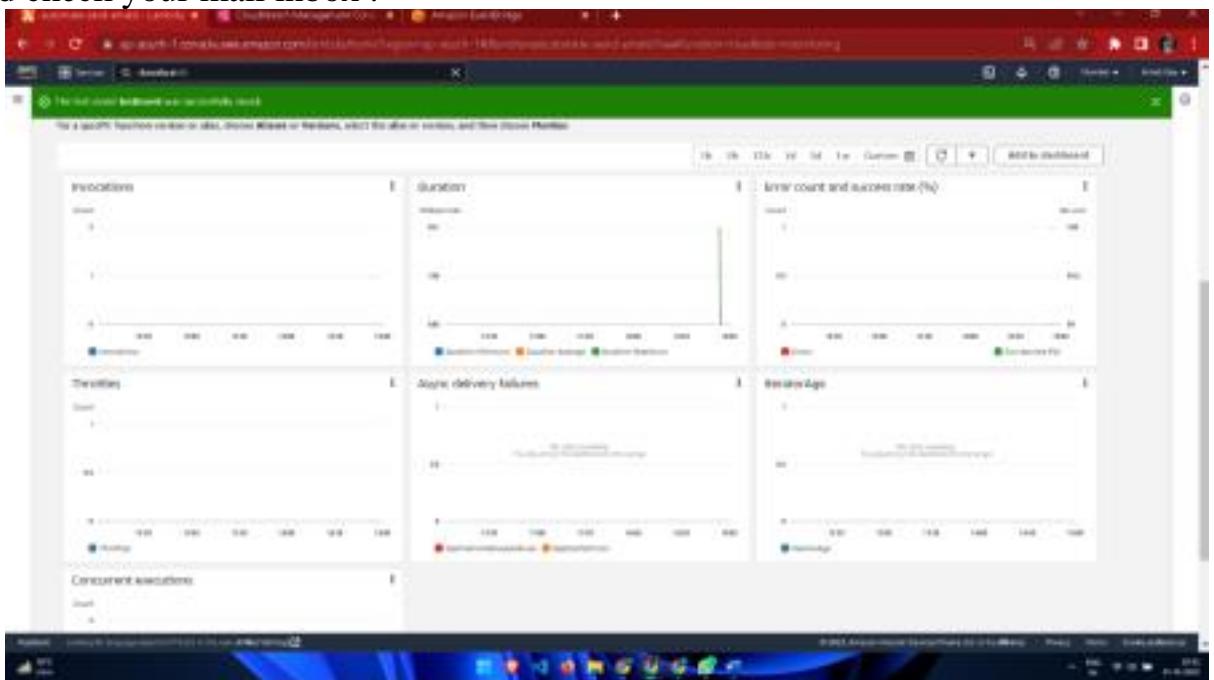
- For scheduled daily report, go to AWS Cloudwatch , navigate to rule section (now called as eventBridg

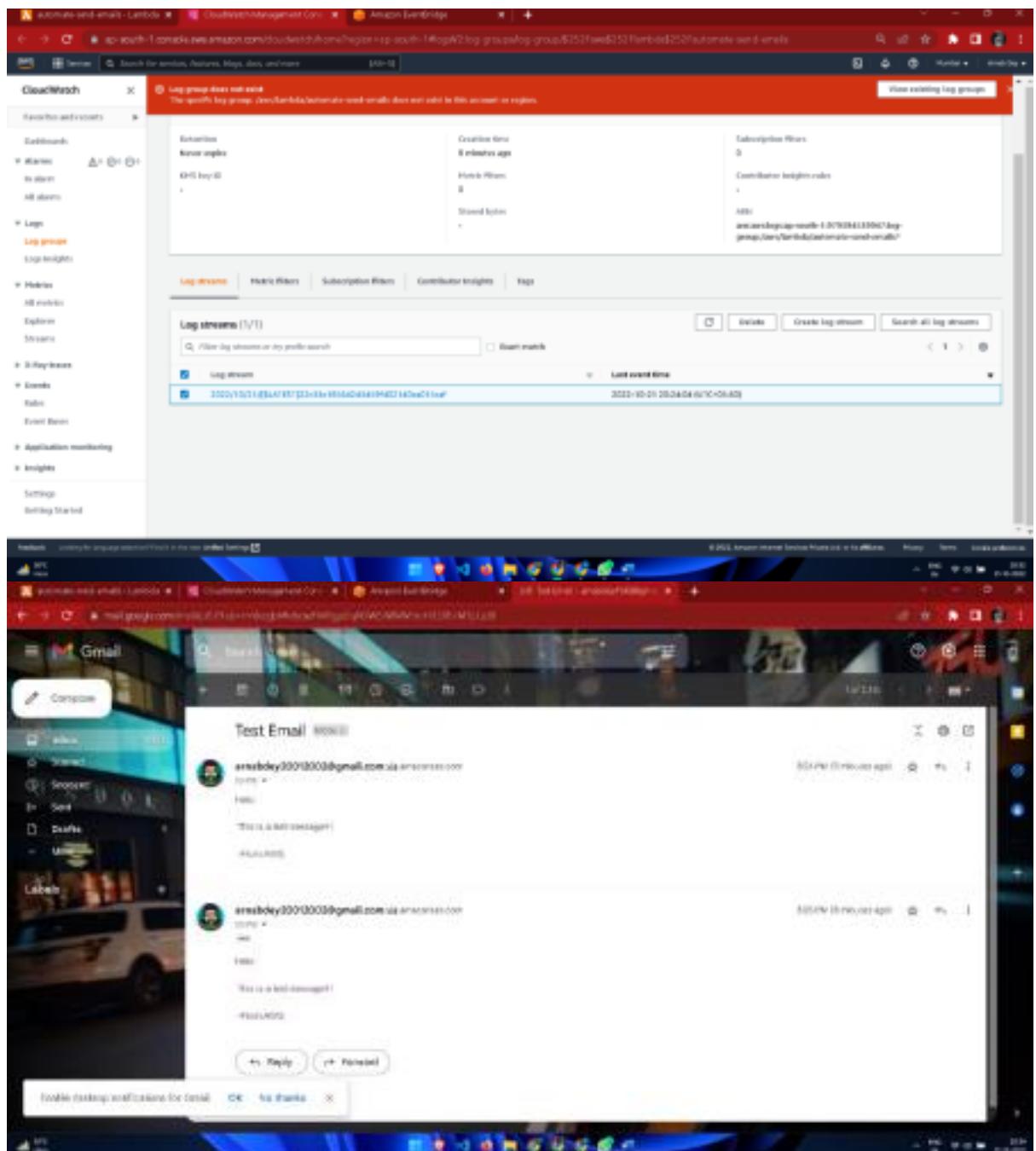


- Create rule- give name, ruletype- schedule, use cron expression for schedule pattern .
For e.g. : 15 19 * * ? *



10. Select Targets as lambda function, and use the above defined function.
11. Go to monitoring in Lambda service, click on View logs in cloudWatch and check your mail inbox .





Result:

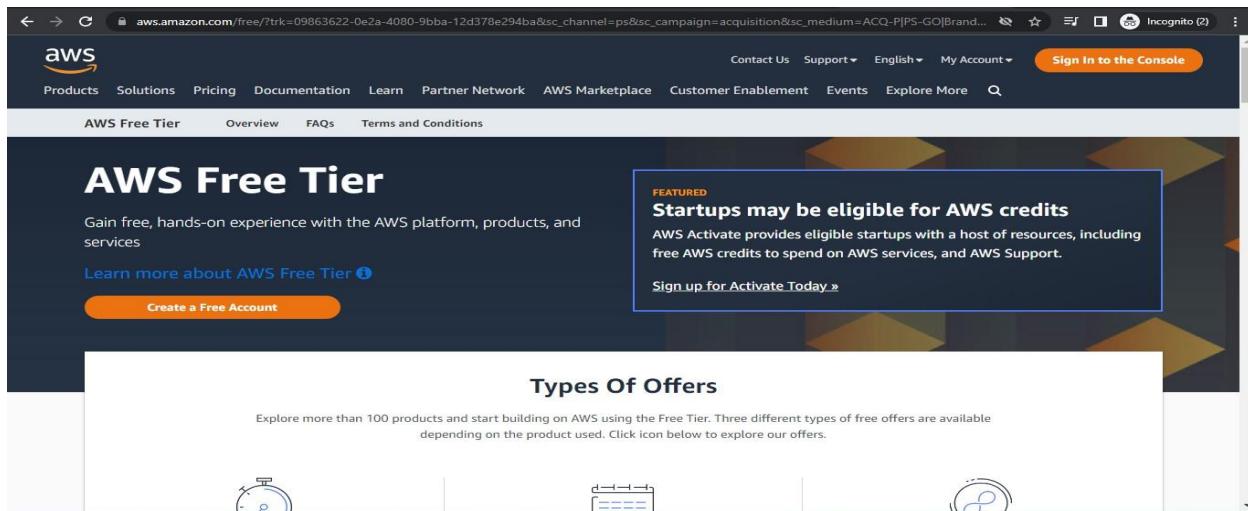
Hence, the lambda function is created and also implemented using SES, CloudWatch to schedule daily reports.

Vaibhavi Tandon
RA2011028010087
Experiment 8

Migrate to Amazon RDS

PROCEDURE:

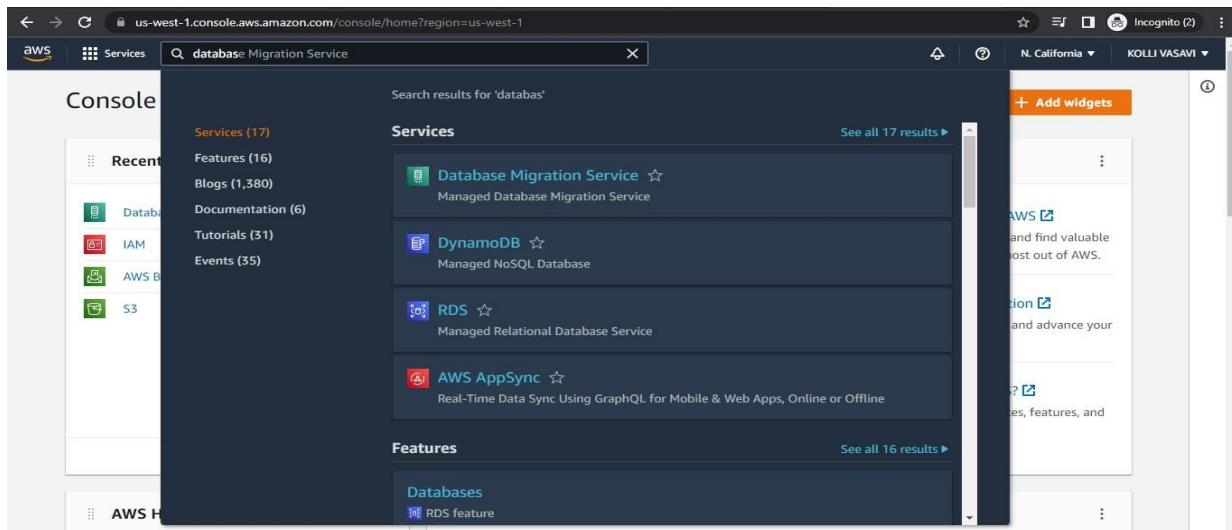
https://aws.amazon.com/free/?trk=09863622-0e2a-4080-9bba-12d378e294ba&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P|PS-GO|Brand|Desktop|SU|AWS|Core|IN|EN|Text&s_kwcid=AL!4422!3!453325185010!e!!g!!aws%20free&ef_id=Cj0KCQjw_7KXBhCoARIsAPdPTfiIj_nDXTj072T5S-Pc3j6qaBSDqVs-6FJI1WtuV8Eo3mdZUwcv5_8aArdoEALw_wcB:G:s&s_kwcid=AL!4422!3!453325185010!e!!g!!aws%20free&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&aws.Free%20Tier%20Types=all&awsFree%20Tier%20Categories=all



The screenshot shows the AWS Free Tier landing page. At the top, there's a navigation bar with links like 'Products', 'Solutions', 'Pricing', 'Documentation', 'Learn', 'Partner Network', 'AWS Marketplace', 'Customer Enablement', 'Events', 'Explore More', and a search bar. A prominent orange 'Create a Free Account' button is located on the left. On the right, there's a 'FEATURED' box for 'Startups' with a link to 'Sign up for Activate Today'. Below this, a section titled 'Types Of Offers' displays three small icons representing different service categories.

Login to your AWS account

2.



The screenshot shows the AWS Management Console search results for 'database Migration Service'. The search bar at the top has 'database Migration Service' typed into it. The results are categorized under 'Services' (17) and 'Features' (16). Under 'Services', 'Database Migration Service' is listed as a Managed Database Migration Service. Under 'Features', 'DynamoDB' is listed as a Managed NoSQL Database, 'RDS' as a Managed Relational Database Service, and 'AWS AppSync' as Real-Time Data Sync Using GraphQL for Mobile & Web Apps, Online or Offline.

Search for DATABASE MANAGEMENT SERVICE.

3.

The screenshot shows the AWS DMS console home page. On the left, there's a sidebar with options like 'AWS DMS', 'DMS Studio', 'New', 'Dashboard', 'Database migration tasks', 'Replication instances' (which is highlighted in orange), 'Endpoints', 'Certificates', 'Subnet groups', 'Events', and 'Event subscriptions'. A 'What's new' section with a '12' badge is also present. The main content area features the title 'AWS Database Migration Service' and the subtitle 'Migrate your databases to AWS with minimal downtime'. It includes a callout for 'Getting started' with a 'Create replication instance' button. Below this, there's a section titled 'How it works' and a note about migrating to a different database. At the bottom, there's a footer with links for 'Privacy', 'Terms', and 'Cookie preferences'.

Open REPLICATION INSTANCES

4.

The screenshot shows the 'Replication instances' page under the 'DMS' service. The top navigation bar includes 'Services' and a search bar. The main content area displays a table header with columns for 'Name', 'Status', 'VPC', 'Class', 'Engine version', 'Availability zone', 'Public', 'Public IP address', 'Private IP address', and 'Multi AZ'. Below the header, a message states 'Empty replication instance table' and 'You don't have any replication instances.' A 'Create replication instance' button is located at the bottom of this section. The footer contains links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

Click on CREATE REPLICATION INSTANCE

5.

The screenshot shows the 'Create replication instance' configuration page. The top navigation bar includes 'Services' and a search bar. The main content area is titled 'Create replication instance' and has a sub-section 'Create replication instance'. It contains several input fields: 'Name' (with 'test-instance' entered), 'Descriptive Amazon Resource Name (ARN) - optional' (with 'test-instance' entered), 'Description' (with 'test DMS replication instance' entered), and 'Instance class' (with 'Info' selected). A note below the instance class says 'Choose an appropriate instance class for your replication needs. Each instance class provides differing levels of compute, network and memory capacity. DMS pricing'. The footer contains links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

Enter your replication instance NAME, DESCRIPTION

6.

This screenshot shows the first step of creating a replication instance. It's a form-based interface with several dropdown menus and sections:

- Instance class:** A dropdown menu showing "dms.t3.medium" selected, which includes "2 vCPUs" and "4 GB Memory". There's also a checkbox for "Include previous-generation instance classes".
- Engine version:** A dropdown menu showing "3.4.7" selected, with a checkbox for "Include Beta DMS versions".
- Upgrades to versions 3.4.7 and higher:** A callout box with information about upgrading to AWS DMS versions 3.4.7 and higher, mentioning VPC endpoints and public routes. It includes a "View endpoints" button.
- Allocated storage (GiB):** A dropdown menu showing "20" selected.
- Feedback:** A link at the bottom left.
- Footer:** Standard footer links for 2022, Privacy, Terms, and Cookie preferences.

This screenshot shows the second step of creating a replication instance. It includes:

- VPC:** A dropdown menu showing "vpc-04c21674d0ef161c9" selected.
- Multi AZ:** A section explaining the difference between Multi-AZ and Single-AZ options.
- Workload type:** A dropdown menu showing "Dev or test workload (Single-AZ)" selected.
- Publicly accessible:** A checkbox that is checked, with a note explaining it assigns a public IP address.
- Advanced security and network configuration:** A section with two dropdown menus: "Replication subnet group" (set to "default-vpc-04c21674d0ef161c9") and "Availability zone" (set to "No Preference").
- Footer:** Standard footer links for 2022, Privacy, Terms, and Cookie preferences.

This screenshot shows the third step of creating a replication instance. It includes:

- VPC security group(s):** A dropdown menu showing "Use default" selected, with "default" listed below it.
- KMS key:** A dropdown menu showing "(Default) aws/dms" selected.
- Maintenance:** A section with a "▶" icon.
- Tags:** A section with a "▶" icon.
- Note:** A callout box stating "DMS requires access permissions to manage your VPC resources. By clicking Create replication instance, you grant permission for DMS to create a role that has this access."
- Buttons:** "Cancel" and "Create" buttons at the bottom right.
- Footer:** Standard footer links for 2022, Privacy, Terms, and Cookie preferences.

Give the required details and click CREATE

7.

The screenshot shows the AWS DMS console. On the left, there's a sidebar with options like 'Dashboard', 'Database migration tasks', 'Replication instances', 'Endpoints' (which is selected and highlighted in orange), 'Certificates', 'Subnet groups', 'Events', and 'Event subscriptions'. Below that is a 'What's new' section with a '12' badge. The main area is titled 'Endpoints' and shows a table with columns 'Name', 'Type', 'Status', 'Engine', and 'Server name'. A message at the top says 'Empty endpoint table' and 'You don't have any endpoints.' At the bottom right of the table area is a large orange button labeled 'Create endpoint'.

From the console choose ENDPOINTS and create endpoint

8.

The screenshot shows the 'Create endpoint' configuration screen. The left sidebar is identical to the one in step 7. The main area has a title 'Create endpoint'. Under 'Endpoint type', there are two options: 'Source endpoint' (selected) and 'Target endpoint'. Below each option is a brief description. There is also a checked checkbox for 'Select RDS DB instance'. A dropdown menu labeled 'RDS Instance' is shown, with the note 'Instances available only for current user and region'. At the bottom of the screen is a section titled 'Endpoint configuration'.

Select the ENDPOINT TYPE

9.

AWS DMS

Instances available only for current user and region
source-database

Endpoint configuration

Endpoint identifier Info
A label for the endpoint to help you identify it.
source-database

Descriptive Amazon Resource Name (ARN) - optional
A friendly name to override the default DMS ARN. You cannot modify it after creation.
source-db

Source engine
The type of database engine this endpoint is connected to.
MySQL

Access to endpoint database
 Choose AWS Secrets Manager
 Provide access information manually

Secret ID
Amazon Resource Name (ARN) of the secret used to manage access to your endpoint database. If you need a new secret ARN, create it from the AWS Secrets Manager console.



AWS DMS

source-db

Source engine
The type of database engine this endpoint is connected to.
MySQL

Access to endpoint database
 Choose AWS Secrets Manager
 Provide access information manually

Server name
source-database.c39ashxlmfg.us-west-1.rds.amazonaws.com

Port The port the database runs on for this endpoint. 3306	Secure Socket Layer (SSL) mode The type of Secure Socket Layer enforcement none
---	--

User name Info
admin

Password Info

Endpoint settings



AWS DMS

source-database.c39ashxlmfg.us-west-1.rds.amazonaws.com

Port The port the database runs on for this endpoint. 3306	Secure Socket Layer (SSL) mode The type of Secure Socket Layer enforcement none
---	--

User name Info admin	Password Info *****
--------------------------------	-------------------------------

Endpoint settings

Define additional specific settings for your endpoints using wizard or editor. [Learn more](#)

Wizard
Enter endpoint settings using the guided user interface.

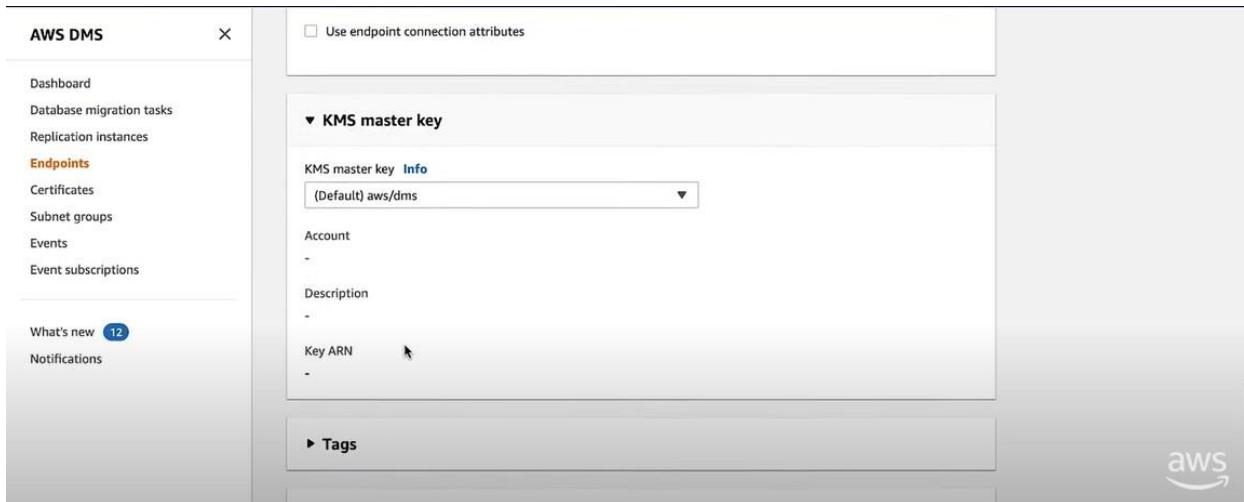
Editor
Enter endpoint settings in JSON format.

Endpoint settings

Add new setting

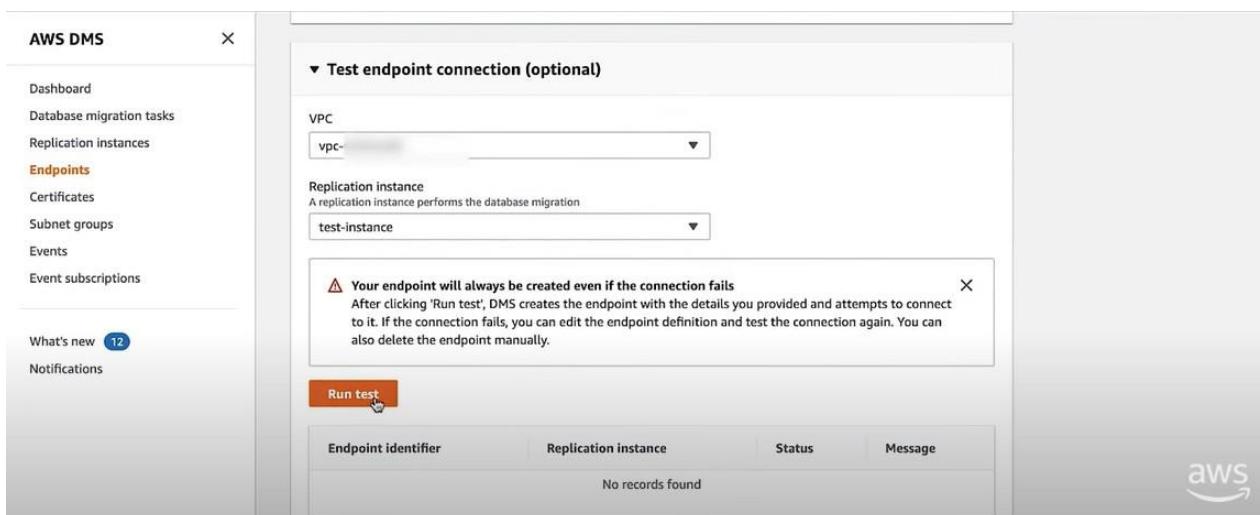
Use endpoint connection attributes





Give the required information

10.



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

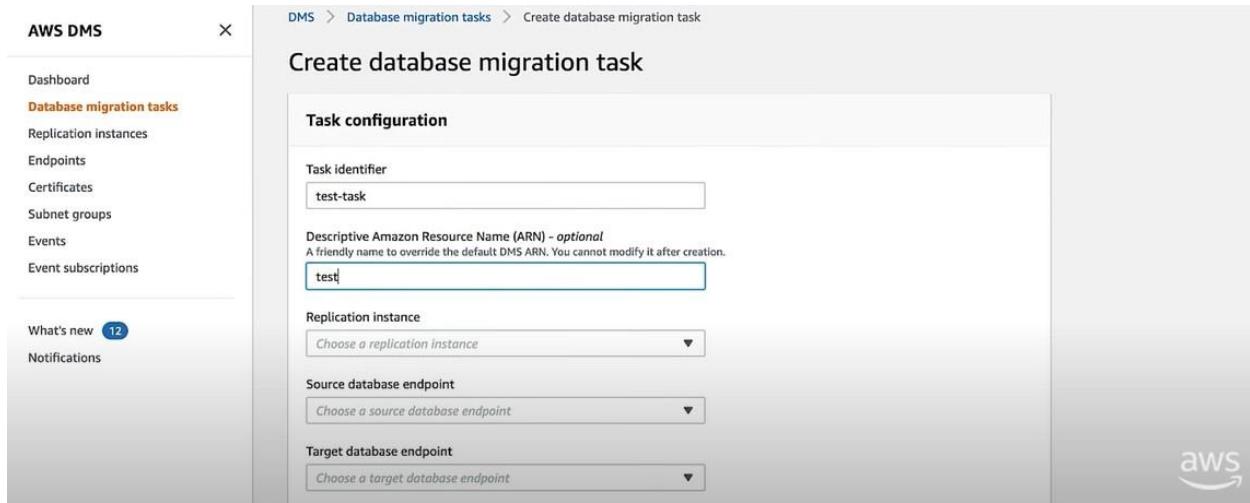
11.

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

12.

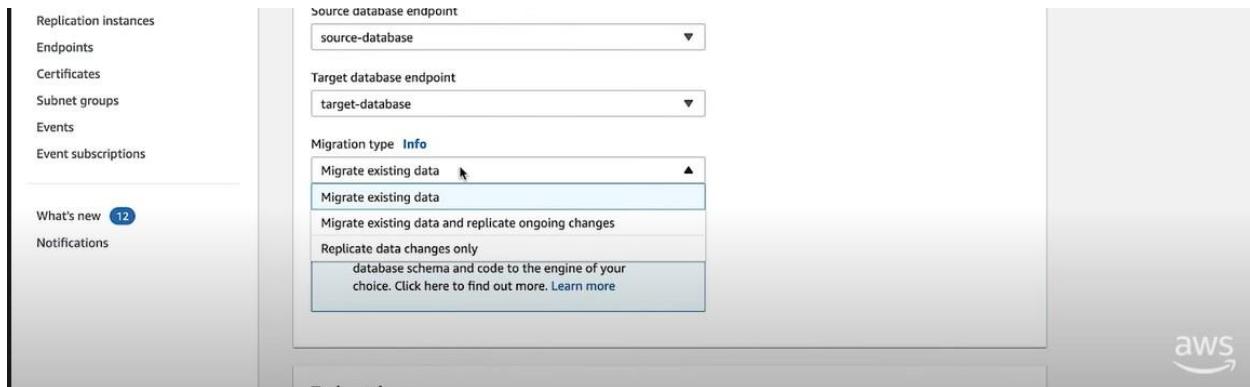
Choose the DATABASE MIGRATION TASKS on the console and choose CREATE TASK

13.



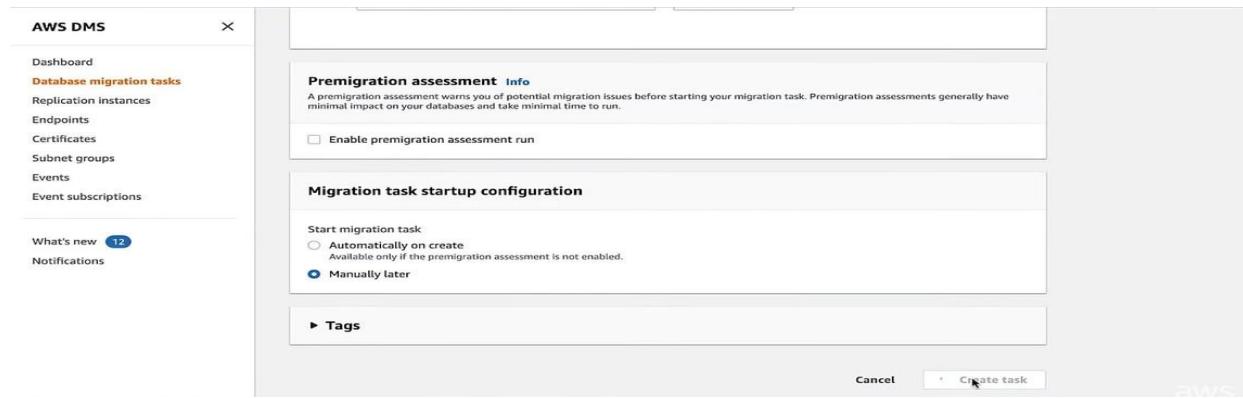
Create the task by giving required details

14.



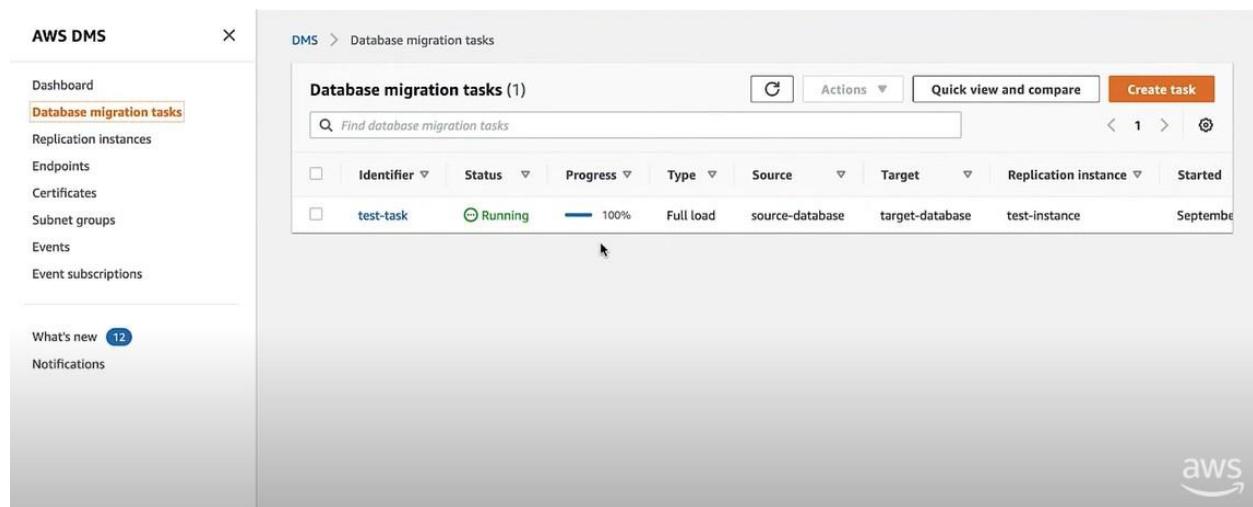
Click on Migrate existing data for one time migration and in task setting update the fields as required.

15.



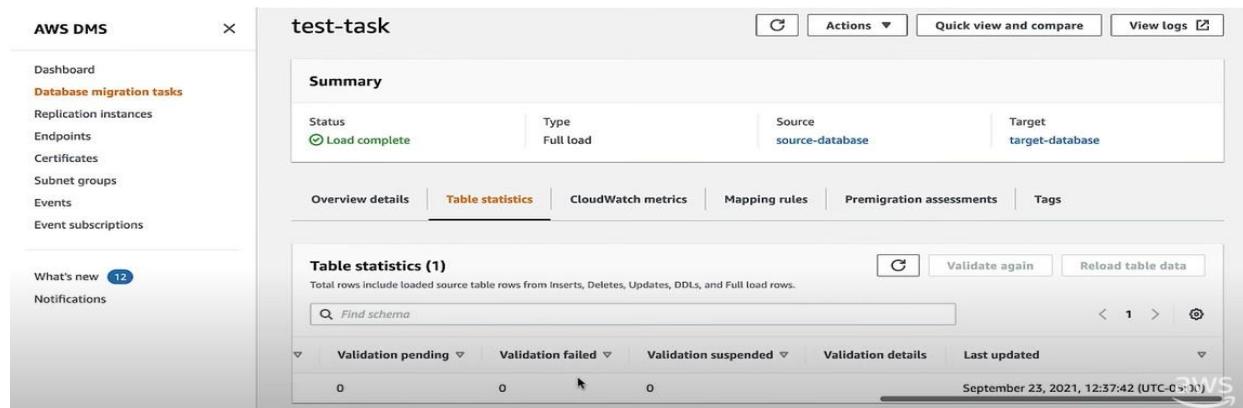
After giving the required details choose create task.

16.



Run the task.

17.



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

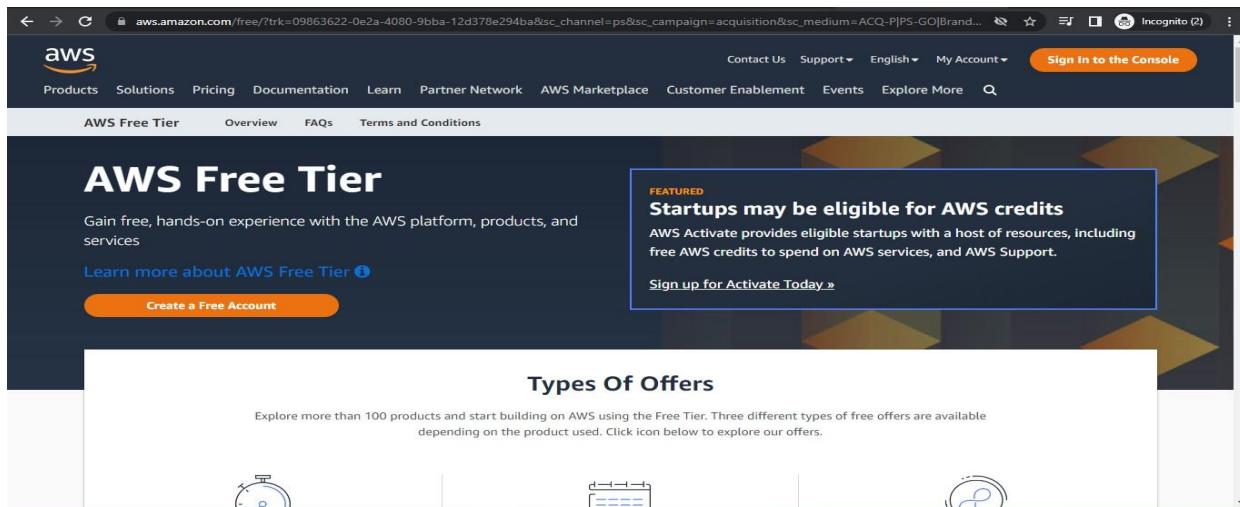
Vaibhavi Tandon

RA2011028010087

Ex-9Configure Failover Routing with Amazon Route 53

PROCEDURE:

https://aws.amazon.com/free/?trk=09863622-0e2a-4080-9bba-12d378e294ba&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P|PS-GO|Brand|Desktop|SU|AWS|Core|IN|EN|Text&s_kwcid=AL!4422!3!453325185010!e!!g!!aws%20free&ef_id=Cj0KCQjw_7KXBhCoARIsAPdPTfiIj_nDXTj072T5S-3j6qaBSDqVs-6FJI1WtuV8Eo3mdZUwcv5_8aArdoEALw_wcB:G:s&s_kwcid=AL!4422!3!453325185010!e!!g!!aws%20free&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all



Login to your AWS account

1.

A screenshot of the AWS Route 53 Hosted Zones interface. The left sidebar lists options like Dashboard, Hosted zones (which is selected and highlighted in yellow), Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, Pending requests, Resolver, VPCs, Inbound endpoints, and Outbound endpoints. The main pane shows a table of record sets for the 'avinash.website' hosted zone. The table has columns for Name, Type, Value, and Evaluate Target Health. Two entries are listed: one NS record 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 one SOA record for ns-165.awsdns-20.com with the value 'awsdns-hostmaster.amazon.com.'. Buttons at the top include Back to Hosted Zones, Create Record Set, Import Zone File, Delete Record Set, and Test Record Set.

Go to Hosted zones.

2.

Go to health checks and create health check

3.

Give the required details.

4.

Give the endpoint of which you want to monitor.

5.

The screenshot shows the 'Specify endpoint by' section with 'Domain name' selected. The 'Protocol' is set to 'HTTP', 'Domain name' is 'mumbaiELB-25996257.ap-south-1.elb.amazonaws.com', 'Port' is '80', and 'Path' is '/images'. Below this, there's an 'Advanced configuration' link and a URL field containing 'http://mumbaiELB-25996257.ap-south-1.elb.amazonaws.com:80/'. The 'Health check type' is listed as '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

This screenshot shows the 'Get notified when health check fails' step. It includes a note about creating an alarm for SNS notifications. A 'Create alarm' button has 'Yes' selected. Below it, a note says CloudWatch sends an SNS notification whenever the health check is unhealthy for one minute. The 'Send notification to' section shows 'Existing SNS topic' selected with 'Default_CloudWatch_Alarms_Topic' chosen. At the bottom are 'Cancel', 'Previous', and a prominent blue 'Create health check' button.

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

7.

The screenshot shows the 'Health checks' section of the AWS Route 53 dashboard. A success message at the top says 'Health check with id 9459b641-1d77-4853-b12e-6d9bd9d0d6b3 has been created successfully'. The main table lists a single health check named 'prodhc' with status 'Unknown'. The 'Description' column shows the URL 'http://mumbaiELB-25996257.ap-south-1.elb.amazonaws.com:80/'. The 'Alarms' column shows '1 of 1 in INSUFFICIENT'. Below the table are tabs for 'Info', 'Monitoring', 'Alarms', 'Tags', 'Health checkers', and 'Latency'. Both the 'Monitoring' and 'Alarms' tabs show 'No health check selected.'

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

8.

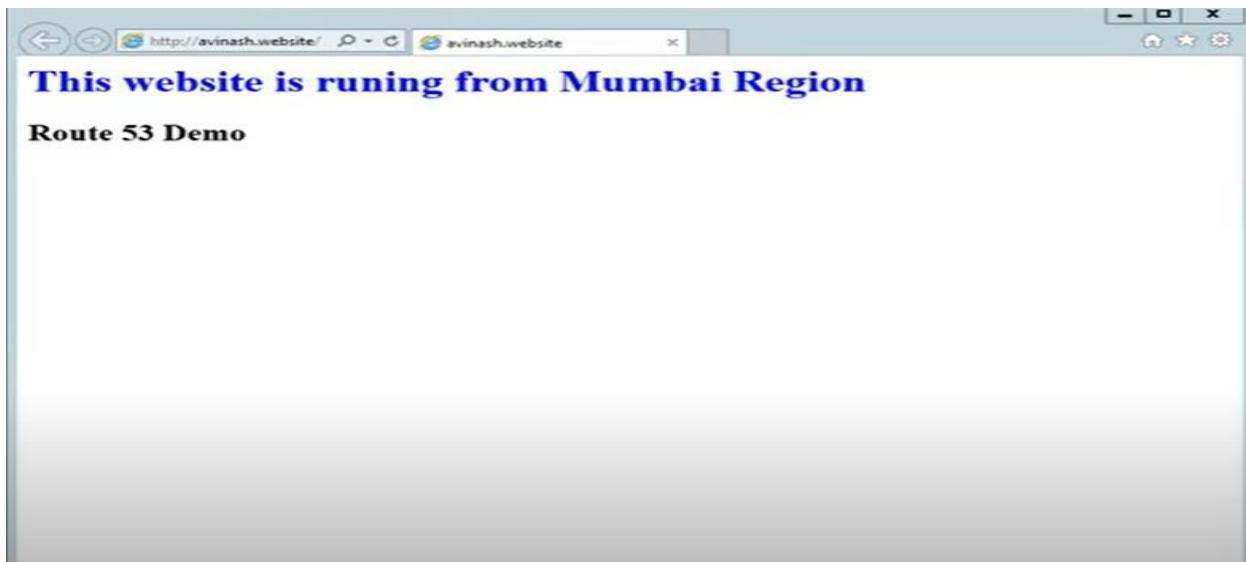
The screenshot shows the AWS Route 53 'Create Record Set' interface. On the left, there's a sidebar with navigation links like Dashboard, Hosted zones (which is selected), Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, Pending requests, Resolver, VPCs, Inbound endpoints, and Outbound endpoints. The main area has tabs for 'Back to Hosted Zones', 'Create Record Set' (which is active), 'Import Zone File', 'Delete Record Set', and 'Test Record Set'. A search bar at the top says 'Record Set Name' with a placeholder 'Any Type'. Below it, there are checkboxes for 'Aliases Only' and 'Weighted Only'. A table lists two record sets: 's-165.awsdns-20.com.' and 's-1919.awsdns-47.co.uk.'. The right side of the interface shows a 'Create Record Set' form with fields for 'Name' (set to 'avinash.website'), 'Type' (set to 'A - IPv4 address'), 'Alias' (radio button set to 'Yes'), and 'Alias Target' (set to 'dualstack.mumbai.ELB-25996257.ap-south-1.elb.amazonaws.com'). A note below the target field provides examples of what can be used as targets.

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

9.

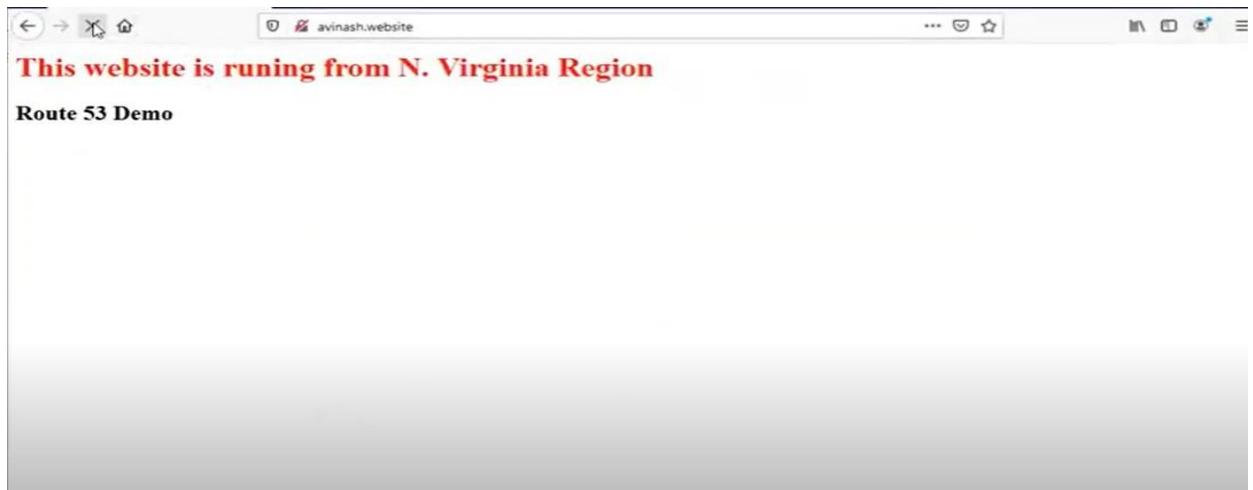
Repeat the same steps for the secondary set ID.

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.