

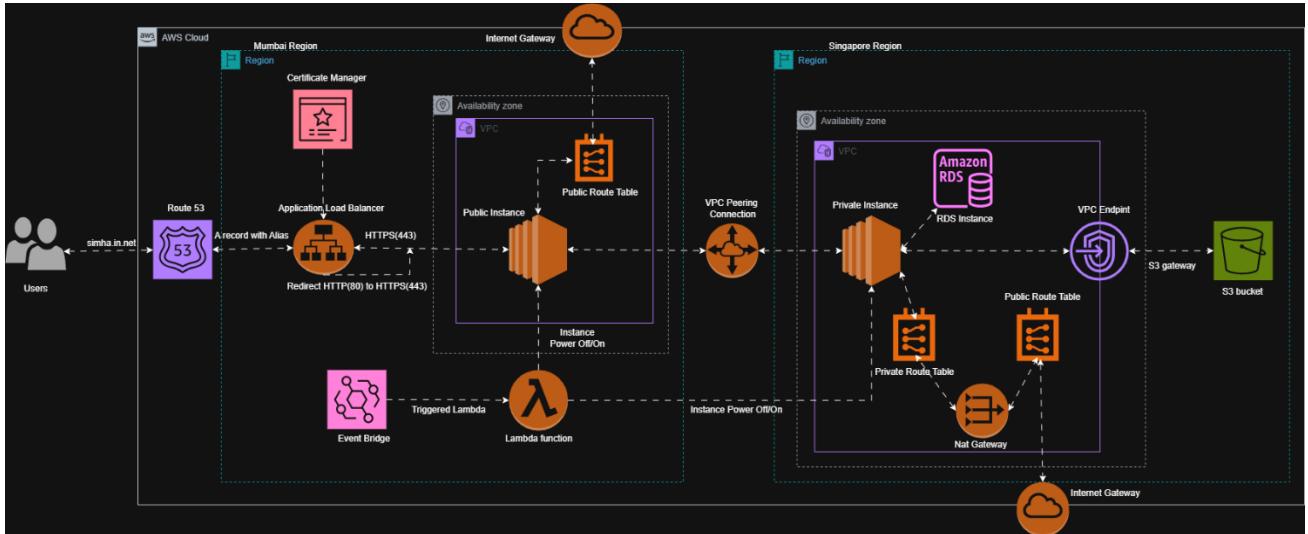
A Secure Multi-Tier Web Application Architecture on AWS

Table of Contents

1. Introduction.....	2
2. Pre-requisites.....	2
3. Summary.....	2
4. Procedure module wise:	
1. Set up a secure connection between the public instance and the private instance located in different regions.....	3
2. Install the required packages and clone the code on both the public instance and the private instance.....	12
3. Create an RDS instance, set up the connection to the private instance and configuration accordingly.....	15
4. Configure the S3 bucket for the private instance using a VPC endpoint.....	23
5. Configure automatic power off and on for an EC2 instance using AWS Lambda and Event Bridge.....	29
6. Setting up a secure web application using Route 53, a Load Balancer, and ACM.....	34
5. Overall, of this project.....	45

1. Introduction:

This document outlines the steps to configure and establish a Secure Multi-Tier Web Application Architecture on AWS.



2. Pre-requisites

- Appropriate Access/Credentials to login to the AWS Console.
- AmazonEC2FullAccess
- AmazonVPCFullAccess
- AmazonRDSFullAccess
- AmazonS3FullAccess
- AWSCertificateManagerFullAccess
- AWSLambda_FullAccess
- CloudWatchEventsFullAccess
- AmazonRoute53FullAccess

3. Summary:

The architecture ensures secure and efficient operations by maintaining a strict separation between public and private resources. It leverages key AWS services such as Amazon RDS for reliable database management, S3 with VPC endpoints for secure data storage, and ACM for SSL certification to enhance security. Amazon Route 53 is ensuring reliable domain name resolution and traffic routing. VPC peering enables seamless and secure communication across AWS regions, while a NAT Gateway in the private subnet ensures controlled internet access for updates and external communications without exposing private instances. An Elastic Load Balancer (ELB) ensures efficient traffic distribution and availability. Automation is implemented using AWS Lambda and Event Bridge, enabling cost optimization by powering instances on and off during non-business hours.

This project represents a robust, scalable, and cost-effective solution for deploying secure web applications. The architecture is future-ready, with provisions for advanced features such as auto-scaling, ensuring it can adapt to growing business demands while maintaining optimal security and performance.

Module 1:

Set up a secure connection between the public instance and the private instance located in different regions.

1. Create a VPC with a CIDR block of “172.31.0.0/16” in the Mumbai region.

The screenshot shows the AWS VPC console interface. At the top, there is a navigation bar with icons for Home, Services, and Regions (Mumbai selected). The main header reads "Your VPCs (1/1) Info". Below the header, there is a search bar and a table with the following data:

Name	VPC ID	State	Block Public...	IPv4 CIDR
PublicVPC	vpc-0ab9222cd31d02273	Available	Off	172.31.0.0/16

Below the table, the details for the VPC are shown:

vpc-0ab9222cd31d02273 / PublicVPC

Details | Resource map | CIDRs | Flow logs | Tags | Integrations

Details

VPC ID vpc-0ab9222cd31d02273	State Available	Block Public Access Off	DNS hostnames Enabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-003a667ff807d0eaf	Main route table rtb-0de2905abfc36a848
Main network ACL arn:aws:acel:ap-south-1:123456789012:ace-0ab9222cd31d02273	Default VPC Yes	IPv4 CIDR 172.31.0.0/16	IPv6 pool -

2. Create a subnet within the VPC and modify its settings to enable the assignment of public IP addresses.

Subnets (1/3) [Info](#)

Last updated less than a minute ago | [Actions](#) | [Create subnet](#)

Name	Subnet ID	State	VPC	Block Public
PublicSubnet	subnet-0c77452e16c19ded6	Available	vpc-0ab9222cd31d02273 Pub...	Off
-	subnet-053819e25328c54e9	Available	vpc-0ab9222cd31d02273 Pub...	Off

subnet-0c77452e16c19ded6 / PublicSubnet

[Details](#) | [Flow logs](#) | [Route table](#) | [Network ACL](#) | [CIDR reservations](#) | [Sharing](#) | [Tags](#)

Details

Subnet ID subnet-0c77452e16c19ded6	Subnet ARN arn:aws:ec2:ap-south-1:590183945701:subnet/subnet-0c77452e16c19ded6	State Available	Block Public Access Off
IPv4 CIDR 172.31.0.0/20	IPv6 CIDR -	IPv6 CIDR association ID -	Network border group ap-south-1
Availability Zone ap-south-1b	Available IPv4 addresses 4090	Availability Zone ID aps1-az3	VPC vpc-0ab9222cd31d02273 PublicVPC
Route table	Default subnet		

3. Create an Internet Gateway and attach it to the VPC.

Internet gateways (1/1) [Info](#)

[Actions](#) | [Create internet gateway](#)

Name	Internet gateway ID	State	VPC ID
PublicVPC's_IG	igw-0c9ab6a110953517d	Attached	vpc-0ab9222cd31d02273 PublicVPC

igw-0c9ab6a110953517d / PublicVPC's_IG

[Details](#) | [Tags](#)

Details

Internet gateway ID igw-0c9ab6a110953517d	State Attached	VPC ID vpc-0ab9222cd31d02273 PublicVPC	Owner 590183945701
--	-----------------------------------	---	---------------------------------------

4. Update the VPC's route table to associate it with the subnet and add a route to the Internet Gateway linked to that VPC.

Route tables (1/1) Info

Last updated less than a minute ago

rtb-0de2905abfc36a848 / PublicinstanceRT

Subnet associations

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
PublicSubnet	subnet-0c77452e16c19ded6	172.31.0.0/20	-

Subnets without explicit associations (2)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Route tables (1/1) Info

Last updated 1 minute ago

rtb-0de2905abfc36a848 / PublicinstanceRT

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0c9ab6a110953517d	Active	No
172.31.0.0/16	local	Active	No

5. Create a security group for the VPC with inbound rules that allow HTTP and HTTPS traffic from anywhere, as well as SSH and ICMP (IPv4) for the specified IP range.

Security Groups (1/2) Info

Export security groups to CSV

Inbound rules (4)

version	Type	Protocol	Port range	Source	Description
v4	HTTP	TCP	80	0.0.0.0/0	-
v4	All ICMP - IPv4	ICMP	All	122.166.215.32/32	-
v4	SSH	TCP	22	122.166.215.32/32	-
v4	HTTPS	TCP	443	0.0.0.0/0	-

6. Launch a web server (public instance) within the public VPC to host the web application.

The screenshot shows the AWS EC2 Instances page. At the top, there are filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. A search bar at the top left contains the placeholder "Find Instance by attribute or tag (case-sensitive)". Below the filters, a table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Web_server	i-0a9cff391a0905f79	Running	t2.micro	Initializing	View alarms +	ap-south-1b

Below the table, the instance details for "i-0a9cff391a0905f79 (Web_server)" are shown. The "Instance summary" section includes fields for Instance ID (i-0a9cff391a0905f79), Public IPv4 address (13.203.76.114), Private IPv4 address (172.31.2.80), Public IPv4 DNS (ec2-13-203-76-114.ap-south-1.compute.amazonaws.com), and Instance state (Running).

7. Similar, create another VPC for a private network, ensuring that the subnet does not have a public IP enabled and that no Internet Gateway is required.

The screenshot shows the AWS VPCs page. At the top, there are filters for Name, VPC ID, State, Block Public..., IPv4 CIDR, and IPv6 CIDR. A search bar at the top left contains the placeholder "Search". Below the filters, a table lists one VPC:

Name	VPC ID	State	Block Public...	IPv4 CIDR
PrivateVPC	vpc-04c463f0d67a6044e	Available	Off	10.0.0.0/24

Below the table, the details for "vpc-04c463f0d67a6044e / PrivateVPC" are shown. The "Details" tab is selected, displaying the following configuration:

VPC ID	State	Block Public Access	DNS hostnames
vpc-04c463f0d67a6044e	Available	Off	Enabled
DNS resolution	Tenancy	DHCP option set	Main route table
Enabled	default	dopt-01d274818061ed760	rtb-0e7aaafcfc225906bd
Main network ACL	Default VPC	IPv4 CIDR	IPv6 pool
arn:aws:acm-pca:ap-south-1:123456789012:certificate/12345678901234567890123456789012	N/A	10.0.0.0/24	

Subnets (1/2) Info

Name	Subnet ID	State	VPC
private_subnet	subnet-03d70a7602345fc2b	Available	vpc-04c463f0d67a6044e PrivateVPC
RDS-Pvt-subnet-1	subnet-0a046ef6cadd554fb	Available	vpc-04c463f0d67a6044e PrivateVPC

subnet-03d70a7602345fc2b / private_subnet

Details

Subnet ID subnet-03d70a7602345fc2b	Subnet ARN arn:aws:ec2:ap-southeast-1:590183945701:subnet/subnet-03d70a7602345fc2b	State Available	Block Public Access Off
IPv4 CIDR 10.0.0.0/28	Available IPv4 addresses 11	IPv6 CIDR -	IPv6 CIDR association ID -
Availability Zone ap-southeast-1a	Availability Zone ID apse1-az1	Network border group ap-southeast-1	VPC vpc-04c463f0d67a6044e PrivateVPC
Route table		Default subnet	

Route tables (1/3) Info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main
Private_route_table	rtb-0e7aafcf225906bd	subnet-03d70a7602345fc...	-	Yes
RDS-Pvt-rt	rtb-056c8e5f16c8a3c50	subnet-0a046ef6cadd55...	-	No

rtb-0e7aafcf225906bd / Private_route_table

Details

Subnet associations

Explicit subnet associations (1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
private_subnet	subnet-03d70a7602345fc2b	10.0.0.0/28	-

Subnets without explicit associations (0)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Route tables (1/3) Info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main
Private_route_table	rtb-0e7aafcf225906bd	subnet-03d70a7602345fc...	-	Yes
RDS-Pvt-rt	rtb-056c8e5f16c8a3c50	subnet-0a046ef6cadd55...	-	No

rtb-0e7aafcf225906bd / Private_route_table

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No

- Create a security group for the private instance that allows traffic only from the public instance's CIDR block for MySQL, SSH, and ICMP ports.

The screenshot shows the AWS Security Groups console with the following details:

- Security Groups (1/3) Info**: Shows 1 security group named "Privateecc2instanceSG" with ID "sg-054a9be0b2065a766".
- Inbound rules (3)**: Lists three rules:

Type	Protocol	Port range	Source
MySQL/Aurora	TCP	3306	172.31.0.0/16
All ICMP - IPv4	ICMP	All	172.31.0.0/16
SSH	TCP	22	172.31.0.0/16

9. Launch an App Server (private instance) under the private VPC to configure RDS and S3 privately.

The screenshot shows the AWS Instances console with the following details:

- Instances (1/1) Info**: Shows 1 instance named "App_Server" with ID "i-03c712f819be636e7".
- i-03c712f819be636e7 (App_Server)**: Instance summary details:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-03c712f819be636e7	-	10.0.0.10
IPv6 address	Instance state	Public IPv4 DNS
-	Running	-
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-10-0-0-10.ap-southeast-1.compute.internal	ip-10-0-0-10.ap-southeast-1.compute.internal	

10. Now establish a VPC peering connection between the public instance and the private instance by providing the request and acceptor VPC ID:

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

Peering connection settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
connectionfrompublicttoprivateinstance

Select a local VPC to peer with

VPC ID (Requester)
vpc-0ab9222cd31d02273 (PublicVPC)

VPC CIDRs for vpc-0ab9222cd31d02273 (PublicVPC)

CIDR	Status	Status reason
172.31.0.0/16	Associated	-

Select another VPC to peer with

Account
 My account
 Another account

Region
 This Region (ap-south-1)
 Another Region
Asia Pacific (Singapore) (ap-southeast-1)

VPC ID (Acceptor)
vpc-04c463f0d67a6044e

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Actions ▾ [Create peering connection](#)

- [View details](#)
- [Accept request](#)
- [Reject request](#)
- [Edit DNS settings](#)
- [Manage tags](#)
- [Delete peering connection](#)

11. Update the route table of both instances to allow the peering connection by adding routes with the opposite CIDR block as the destination.

Edit routes

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	Active	No
10.0.0.0/24	Peering Connection	-	No

[Add route](#) [Cancel](#) [Preview](#) [Save changes](#)

The screenshot shows the AWS VPC Route Tables interface. A new route is being added to a route table named 'rtb-0e7aafcf225906bd'. The destination is '10.0.0.0/24' and the target is set to 'local'. The status is 'Active' and it is not propagated. A new route is being added with a destination of '172.31.0.0/16' and a target of 'Peering Connection pnx-023d26e7a763ec8df'. The status is '-' and it is not propagated. There is a 'Remove' button next to this route. At the bottom, there are 'Add route', 'Cancel', 'Preview', and 'Save changes' buttons.

12. After establishing the peering connection, test the connectivity between the public and private instances using ping and ssh with the .pem file.

- Copy the PEM file of the private instance to the public instance for login access.
- chmod 400 "singapore-key.pem"
- ssh -i "singapore-key.pem" ubuntu@10.0.0.10

```
ubuntu@ip-172-31-2-80:~$ ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=62.3 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=62.6 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=62.5 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=63.0 ms
^C
--- 10.0.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 62.348/62.638/63.032/0.250 ms
ubuntu@ip-172-31-2-80:~$ 
```

i-0a9cff391a0905f79 (Web_server)

Public IPs: 13.203.76.114 Private IPs: 172.31.2.80

```

ubuntu@ip-172-31-2-80:~$ sudo -i
root@ip-172-31-2-80:~# vi kep.pem
root@ip-172-31-2-80:~# chmod 400 "kep.pem"
root@ip-172-31-2-80:~# ssh -i kep.pem ubuntu@10.0.0.10
The authenticity of host '10.0.0.10 (10.0.0.10)' can't be established.
ED25519 key fingerprint is SHA256:qbZSG0HVnCsXmswZvcCtktsIJx/dDa5am7Kpxze/NTw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.0.10' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Sat Dec 28 18:27:42 UTC 2024

  System load:  0.0                  Processes:          102
  Usage of /:   24.6% of 6.71GB    Users logged in:    0
  Memory usage: 20%                IPv4 address for enX0: 10.0.0.10
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-0-10:~$ █

```

i-0a9cff391a0905f79 (Web_server)

PublicIPs: 13.203.76.114 PrivateIPs: 172.31.2.80

Conclusion on this module:

Now we have successfully established a secure connection from the public instance to the private instance in a different region using the VPC peering connection.

Module 2

Configuring the NAT gateway to private instance and install the required packages and clone the code from GitHub repository

1. Create a public subnet within the VPC that hosts the private instance and update the route table to enable access to the target group through the internet gateway, allowing traffic from any source.

The screenshot shows two main sections of the AWS VPC console:

Subnets (1/4) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
private_subnet	subnet-03d70a7602345fc2b	Available	vpc-04c463f0d67a6044e Priva...	Off	10.0.0.0/28	-
RDS-Pvt-subnet-1	subnet-0a046effcad554fb	Available	vpc-04c463f0d67a6044e Priva...	Off	10.0.128/25	-
RDS-Pvt-subnet-2	subnet-0c6e7265d4495e961	Available	vpc-04c463f0d67a6044e Priva...	Off	10.0.0.32/28	-
Public_subnet	subnet-0b46d114d05fd2ebc	Available	vpc-04c463f0d67a6044e Priva...	Off	10.0.0.16/28	-

subnet-0b46d114d05fd2ebc / Public_subnet

Details

Subnet ID subnet-0b46d114d05fd2ebc	Subnet ARN arn:aws:ec2:ap-southeast-1:590183945701:subnet/subnet-0b46d114d05fd2ebc	State Available	Block Public Access Off
IPv4 CIDR 10.0.0.16/28	Available IPv4 addresses 11	IPv6 CIDR -	IPv6 CIDR association ID -
Availability Zone ap-southeast-1a	Availability Zone ID aps1-az1	Network border group ap-southeast-1	VPC vpc-04c463f0d67a6044e PrivateVPC
Route table rtb-0025935d9f65c4685 Public_route_table	Network ACL acl-05995cb020cb36ce5	Default subnet No	Auto-assign public IPv4 address No
Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -	Outpost ID -
IPv4 CIDR reservations -	IPv6 CIDR reservations -	IPv6-only No	Hostname type IP name
Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled	DNS64 Disabled	Owner 590183945701

Route tables (1/3) Info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
Private_route_table	rtb-0e7aafcf225906bd	subnet-03d70a7602345f...	-	Yes	vpc-04c463f0d67a6044e Priva...	590183945701
RDS-Pvt-rt	rtb-056c8ef16c8a5c50	2 subnets	-	No	vpc-04c463f0d67a6044e Priva...	590183945701
Public_route_table	rtb-0025935d9f65c4685	subnet-0b46d114d05fd2...	-	No	vpc-04c463f0d67a6044e Priva...	590183945701

rtb-0e7aafcf225906bd / Private_route_table

Routes (3)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-036be0331ac397345	Active	No
10.0.0.0/24	local	Active	No
172.31.0.0/16	pxx-04610c5d501a70057	Active	No

2. Set up a NAT Gateway in the public subnet and associate an Elastic IP with it and Wait for a few minutes until the status of the NAT Gateway changes to "Available."

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name optional
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

Connectivity type
Select a connectivity type for the NAT gateway.
 Public
 Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.

Additional settings Info

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="mynatgateway"/> <input type="button" value="Remove"/>

You can add 49 more tags.

nat-036be0331ac397345 / mynatgateway

Actions

Details

NAT gateway ID <input type="text" value="nat-036be0331ac397345"/>	Connectivity type Public	State Available	State message <small>Info</small> -
NAT gateway ARN <input type="text" value="arnaws:ec2:ap-southeast-1:1590183945701:natgateway/nat-036be0331ac397345"/>	Primary public IPv4 address <input type="text" value="122.248.221.92"/>	Primary private IPv4 address <input type="text" value="10.0.0.21"/>	Primary network interface ID <input type="text" value="eni-0efac7aed56e490c4"/>
VPC <input type="text" value="vpc-04c463f0d67a6044e / PrivateVPC"/>	Subnet <input type="text" value="subnet-0b46d114d05fd2ebc / Public_subnet"/>	Created <input type="text" value="Tuesday, December 31, 2024 at 23:03:21 GMT+5:30"/>	Deleted -

Secondary IPv4 addresses

Private IPv4 address	Allocation ID	Association ID	Public IPv4 address	Network
Secondary IPv4 addresses are not available for this nat gateway.				

Elastic IP addresses (1/1)

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP address
<input checked="" type="checkbox"/> -	<input type="text" value="122.248.221.92"/>	Public IP	eipalloc-0d2c91013d4ae69ec	-	-	10.0.0.21

ways
internet

- Now, update the route table of the private of target NAT gateway by giving the destination anyway.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
Private_route_table	rtb-0e7aaafcfc225906bd	subnet-03d70a7602345f...	-	Yes	vpc-04c463f0d67a6044e Priva...	590183945701
RDS-Pvt-rt	rtb-056cbe5f16c8a3c50	2 subnets	-	No	vpc-04c463f0d67a6044e Priva...	590183945701
Public_route_table	rtb-0025935d9f65c4685	subnet-0b46d114d05fd2...	-	No	vpc-04c463f0d67a6044e Priva...	590183945701

rtb-0025935d9f65c4685 / Public_route_table																			
Details	Routes	Subnet associations	Edge associations	Route propagation	Tags														
<table border="1"> <thead> <tr> <th colspan="2">Routes (2)</th> </tr> <tr> <th>Destination</th> <th>Target</th> <th>Status</th> <th>Propagated</th> </tr> </thead> <tbody> <tr> <td>0.0.0.0/0</td> <td>igw-03041911e03b173b8</td> <td>Active</td> <td>No</td> </tr> <tr> <td>10.0.0.0/24</td> <td>local</td> <td>Active</td> <td>No</td> </tr> </tbody> </table>						Routes (2)		Destination	Target	Status	Propagated	0.0.0.0/0	igw-03041911e03b173b8	Active	No	10.0.0.0/24	local	Active	No
Routes (2)																			
Destination	Target	Status	Propagated																
0.0.0.0/0	igw-03041911e03b173b8	Active	No																
10.0.0.0/24	local	Active	No																

Note: For the private instance, configure the security group to allow all traffic (required port) from the public VPC CIDR block and outbound to anyway.

4. Now, on the public instance, install the required packages.
 - sudo apt install upgrade
 - sudo apt install apache2
 - sudo apt install php php-mysqlnd
5. Now, on the private instance, install the required packages.
 - sudo apt install upgrade
 - sudo apt install mariadb-server #For RDS instance configuration.
 - sudo apt install s3fs #For S3 Bucket configuration.
6. clone the code from the GitHub repository using the Git technology:
Git clone <https://github.com/varunsimha-MP/Php CRUD-Basic-.git>
7. On the public instance, set up the web application using Apache2.
 - Transfer the previously cloned code to the /var/www/html directory as shown below.

```
root@ip-172-31-2-80:~# cp -r Php CRUD-Basic-/php_crud/*.html /var/www/html/
root@ip-172-31-2-80:~# cp -r Php CRUD-Basic-/php_crud/*.sql /var/www/html/
root@ip-172-31-2-80:~# cp -r Php CRUD-Basic-/php_crud/*.php /var/www/html/
root@ip-172-31-2-80:~# cp -r Php CRUD-Basic-/php_crud/*.js /var/www/html/
root@ip-172-31-2-80:~# cd /var/www/html
root@ip-172-31-2-80:/var/www/html# ls
basic.js dbconnect.php delete.php delinfo.php index.html php_crud.sql register.php update.php updateuser.php
root@ip-172-31-2-80:/var/www/html# []
```

i-0a9cff391a0905f79 (Web_server)

PublicIPs: 13.203.76.114 PrivateIPs: 172.31.2.80

8. After transferring the code, check if the static/index file is working by accessing it through the public instance's public IP address as the URL.

The screenshot shows a web browser window with the title bar "HTML Form". The address bar indicates the URL is "Not secure 13.203.76.114". The main content area contains a form titled "STUDENT INFORMATION" with a green border. The form fields include:

- First name: [Text input]
- Last name: [Text input]
- USN: [Text input]
- Branch: CSE ISE EC CV MECH
- Semester: 1st Semester
- Gender: Male Female
- Mail Id: [Text input]
- Phone Number: [Text input] XXX-XXX-XXXX
- Select your birthday: [Text input] mm/dd/yyyy
- Select a Photo: [Text input] Choose File No file chosen

Below the form are two buttons: "submit" and "Reset". A note at the bottom states: "Note: Only valid details are considered". At the very bottom right, it says "Created by VSMP".

Conclusion of this module:

We have successfully configured the NAT gateway to private instance for secure internet and installed all necessary packages, including Apache2 for the web application, MariaDB for RDS, and the S3 package for mounting the S3 bucket on the respective instances. Additionally, we cloned the code from the GitHub repository using Git and tested the accessibility of the static web application on the public instance.

Module - 03

Create an RDS instance, set up the connection to the private instance and configuration accordingly

1. Create a role that provides access to the instance by attaching the RDS full access policy.

The screenshot shows the AWS IAM console with the 'Ec2Instancerole' role created. The 'Permissions' tab is active, displaying two attached policies: 'AmazonRDSDataFullAccess' and 'AmazonS3FullAccess'. The ARN of the role is listed as arn:aws:iam::590183945701:role/Ec2Instancerole.

2. Now, attach the RDS full access role to the private instance.

The screenshot shows the AWS EC2 Instances page with the 'Modify IAM role' dialog open for instance i-03c712fb19be636e7. The 'IAM role' dropdown is set to 'Ec2Instancerole'.

3. Now, create the **RDS database** in the same region as the private instance. Choose the standard method and select "MySQL" as the database engine according to the requirements.

RDS > Create database

Create database Info

Choose a database creation method

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

- Aurora (MySQL Compatible) 
- Aurora (PostgreSQL Compatible) 
- MySQL 
- PostgreSQL 
- MariaDB 
- Oracle 
- Microsoft SQL Server 
- IBM Db2 

4. Always choose the latest engine version or required for your and select the "Free Tier" template as we are doing for learning purposes.

RDS > Create database

Edition
 MySQL Community

Engine version Info
View the engine versions that support the following database features.

Show only versions that support the Multi-AZ DB cluster Info
Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show only versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine version
 MySQL 8.4.3

Enable RDS Extended Support Info
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

Templates
Choose a sample template to meet your use case.

Production
Use defaults for high availability and fast, consistent performance.

Dev/Test
This instance is intended for development use outside of a production environment.

Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

MySQL

MySQL is the most source database in on RDS offers the MySQL community flexibility to easily resources or storage database.

- Supports database TBS.
- Supports General Memory Optimized Performance in.
- Supports automatic point-in-time replication.
- Supports up to per instance, with Region or 5 regions.

Availability and durability

Deployment options Info
The deployment options below are limited to those supported by the engine you selected above.

Single DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a single DB instance with no standby DB instances.

Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

5. Provide the database name and the credentials for logging into the RDS instance. Additionally, configure the CPU, memory, and storage according to the requirements.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed
Create your own password or have RDS create a password that you manage.

Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Password strength Strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ ^ @

Confirm master password [Info](#)

MySQL

MySQL is the source database on RDS offers MySQL compatibility to ease resources or database.

- Supports up to 1TB.
- Supports General Purpose Performance.
- Supports point-in-time replication.
- Supports up to 5 per instance, Region or 5 regions.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)
 Show instance classes that support Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Include previous generation classes

Standard classes (includes m classes)
 Memory optimized classes (includes r and x classes)
 Burstable classes (includes t classes)

db.t4g.micro
2 vCPUs | 1 GiB RAM | Network: Up to 2,085 Mbps

Storage

Storage type [Info](#)
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage [Info](#)
 GiB
Allocated storage value must be 20 GiB to 6,144 GiB

Connectivity [Info](#)

Compute endpoints

MySQL

MySQL is the source database on RDS offers MySQL compatibility to ease resources or database.

- Supports up to 1TB.
- Supports General Purpose Performance.
- Supports point-in-time replication.
- Supports up to 5 per instance, Region or 5 regions.

6. For connectivity, choose to set up the connection now and select the private instance.

Note: The connection to the instance can be established after the RDS creation is complete.

Connectivity Info

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

EC2 instance Info
Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-03c712f819be656e7
App_Server

Some VPC settings can't be changed when a compute resource is added
Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group rds-ec2-X is added to the database and another called ec2-rds-X to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

Network type Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

PrivateVPC (vpc-04c463f0d57a6044c)
2 Subnets, 1 Availability Zone

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

Choose existing
Choose existing DB subnet group

Automatic setup
RDS creates a new subnet group for you or reuses an existing subnet group

7. Now, select the automatic options and leave the settings as default, then click on "Create."

DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

Choose existing
Choose existing DB subnet group

Automatic setup
RDS creates a new subnet group for you or reuses an existing subnet group

DB subnet group name
rds-ec2-db-subnet-group-2

Existing DB subnet group reused.

Public access Info

- Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
- No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Additional VPC security group
Choose one or more options
default

Amazon RDS will add a new VPC security group rds-ec2-4 to allow connectivity with your compute resource.

Availability Zone Info
ap-southeast-1a

Certificate authority - optional Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expires May 22, 2061

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

MySQL
MySQL is the r source databa on RDS offers MySQL comm flexibility to ei resources or st database.

- Supports di TIB.
- Supports G Memory & Performance
- Supports ai point-in-tin
- Supports u per instanc Region or 5 region.

The screenshot shows the 'Create database' wizard in the AWS RDS console. It includes sections for 'Database authentication' (with 'Password authentication' selected), 'Monitoring' (with 'Enable Enhanced Monitoring' unchecked), 'Additional configuration' (with 'Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off'), and 'Estimated monthly costs' (listing free tier benefits). A note at the bottom states: 'You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.'

8. Wait for a few minutes until the RDS status changes to "Available."

The screenshot shows the 'Summary' tab for the database 'my-website-db'. Key details include: DB identifier 'my-website-db', Status 'Available', Role 'Instance', Engine 'MySQL Community', and Region & AZ 'ap-southeast-1a'. The 'Connectivity & security' tab is active, showing the endpoint 'my-website-db.c1wuawkouym0.ap-southeast-1.rds.amazonaws.com' and port '3306'. The 'Networking' section shows the availability zone 'ap-southeast-1a', VPC 'PrivateVPC (vpc-04c463f0d67a6044e)', and subnet group 'rds-ec2-db-subnet-group-2'. The 'Security' section lists VPC security groups 'rds-ec2-5 (sg-007da0a0d978299da)' and 'default (sg-054a9be0b2065a766)', both marked as 'Active'. The 'Publicly accessible' setting is set to 'No'.

9. Now, from the private instance configure MariaDB on the Private Instance:

- Log in to the RDS instance using the RDS endpoint:

Syntax: mysql -h <RDS_ENDPOINT> -u admin -p

```
mysql -h my-website-db.c1wuawkouym0.ap-southeast-1.rds.amazonaws.com -u admin -p
```

- Enter the password when prompted to access the RDS instance.

- After logging in, create a database and grant full access to the admin user for the newly created database.

```
CREATE DATABASE my_db;
```

```
GRANT ALL PRIVILEGES ON my_db.* TO 'admin'@'%';
```

```
FLUSH PRIVILEGES;
```

```
Exit;
```

```
Last login: Sat Dec 28 18:55:45 2024 from 172.31.2.80
ubuntu@ip-10-0-0-10:~$ sudo -i
root@ip-10-0-0-10:~# mysql -h my-website-db.clwuawkouym0.ap-southeast-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.4.3 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> create database my_db;
Query OK, 1 row affected (0.038 sec)

MySQL [(none)]> GRANT ALL PRIVILEGES ON my_db.* TO 'admin'@'%';
Query OK, 0 rows affected (0.029 sec)

MySQL [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.006 sec)

MySQL [(none)]> []
```

i-0a9cff391a0905f79 (Web_server)

PublicIPs: 13.203.76.114 PrivateIPs: 172.31.2.80

- Use the code transferred earlier from the public to the private instance, now import the SQL file into the RDS database:

```
mysql -h <RDS_ENDPOINT> -u admin -p my_db < /tmp/php_crud.sql
```

```
(mysql [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.006 sec)

MySQL [(none)]> exit;
Bye
root@ip-10-0-0-10:~# mysql -h my-website-db.clwuawkouym0.ap-southeast-1.rds.amazonaws.com -u admin -p my_db < /tmp/php_crud.sql
-bash: /tmp/php_crud.sql: No such file or directory
root@ip-10-0-0-10:~# mysql -h my-website-db.clwuawkouym0.ap-southeast-1.rds.amazonaws.com -u admin -p my_db < /tmp/Php CRUD-Basic-/php_crud/php_crud.sql
root@ip-10-0-0-10:~# mysql -h my-website-db.clwuawkouym0.ap-southeast-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.4.3 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> use my_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [my_db]> show tables;
+-----+
| Tables_in_my_db |
+-----+
| personalinfo |
+-----+
1 row in set (0.001 sec)

MySQL [my_db]> []
```

i-0a9cff391a0905f79 (Web_server)

PublicIPs: 13.203.76.114 PrivateIPs: 172.31.2.80

- Now, go to the public instance and update the database connection file.

```
{
```

```

    "db_host": "<RDS_ENDPOINT>",

    "db_user": "admin",

    "db_password": "<Your_Password>",

    "db_name": "my_db"

}

```

12. After updating the database connection file, test the database connectivity on the public instance by running:

```
php dbconnect.php
```

Note: If no error occurs, the connection is working fine.

```

root@ip-172-31-2-80:~# cd /var/www/html/
root@ip-172-31-2-80:/var/www/html# ls
basic.js  dbconnect.php  delete.php  delinfo.php  index.html  php_crud.sql  register.php  update.php  updateuser.php
root@ip-172-31-2-80:/var/www/html# vi dbconnect.php
root@ip-172-31-2-80:/var/www/html# cat dbconnect.php
<?php
$servername = "my-website-db.clwuawkouym0.ap-southeast-1.rds.amazonaws.com";
$username = "admin";
$password = "SimhaVarun";
$dbname = "my_db";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
root@ip-172-31-2-80:/var/www/html# php dbconnect.php
root@ip-172-31-2-80:/var/www/html# []

```

i-0a9cff391a0905f79 (Web_server)

PublicIPs: 13.203.76.114 PrivateIPs: 172.31.2.80

13. To test if the database web application is working, use the public instance's IP address as the URL. Fill out the form and submit it to verify the functionality.

HTML Form

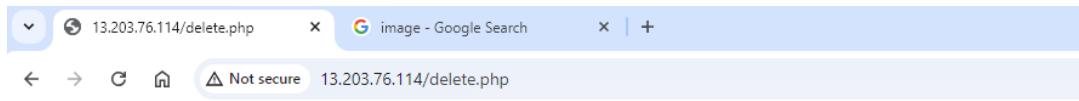
image - Google Search

Not secure 13.203.76.114

STUDENT INFORMATION

First name:	<input type="text" value="varun"/>	Valid
Last name:	<input type="text" value="simha"/>	Valid
USN:	<input type="text" value="4mh19cs113"/>	Valid
Branch:	CSE <input checked="" type="radio"/> ISE <input type="radio"/> EC <input type="radio"/> CV <input type="radio"/> MECH <input type="radio"/>	
Semester:	<input type="button" value="1st Semester"/>	
Gender:	Male: <input checked="" type="radio"/> Female: <input type="radio"/>	
Mail Id:	<input type="text" value="varun@gmail.com"/>	Valid
Phone Number:	<input type="text" value="1234567890"/>	
select your birthday:	<input type="text" value="12/18/2024"/> <input type="button" value=""/>	
Select a Photo:	<input type="file" value="Choose File"/> gratisograph..800x525.jpg	
<input type="button" value="submit"/> <input type="button" value="Reset"/>		

Note: Only valid details are considered



14. Let's check from the instance's end: the changes are reflecting, and everything is working as expected.

```
Last login: Sat Dec 28 19:07:49 2024 from 172.31.2.80
ubuntu@ip-10-0-0-10:~$ mysql -h my-website-db.c1wuawkouym0.ap-southeast-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 31
Server version: 8.4.3 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> use my_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [my_db]> select * from personalinfo;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | firstname | lastname | usn | branch | semester | gender | mail
+----+-----+-----+-----+-----+-----+-----+-----+
| 2 | varun | simha | 4mh19cs113 | CSE | 1st Semester | Male | varun@gmail.com | 1234567890 | 2024-12-18
+----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.002 sec)

MySQL [my_db]> 
```

i-0a9cff391a0905f79 (Web_server)

PublicIPs: 13.203.76.114 PrivateIPs: 172.31.2.80

Conclusion for this module:

We have successfully created an RDS instance and configured it with the private instance according to our code requirements. The SQL file was imported into the RDS, and the connection was established from the public instance by updating the dbconnect.php file. After testing, we confirmed that the public instance's IP address is working as expected, and the changes are reflected in the RDS.

Module - 04

Configure the S3 bucket for the private instance using a VPC endpoint

1. Create a role that grants access to the instance by attaching the "AmazonS3FullAccess" policy.

The screenshot shows the 'Ec2Instancerole' configuration page in the AWS IAM console. The 'Permissions' tab is selected. Under 'Permissions policies', two policies are attached: 'AmazonRDSDataFullAccess' and 'AmazonS3FullAccess'. The 'AmazonS3FullAccess' policy is highlighted with a blue border. Other tabs include 'Trust relationships', 'Tags', 'Last Accessed', and 'Revoke sessions'. The ARN of the role is listed as arn:aws:iam::590183945701:role/Ec2Instancerole, and its instance profile ARN is arn:aws:iam::590183945701:instance-profile/Ec2Instancerole.

2. Now, attach the "S3 Full Access" role to the private instance.

The screenshot shows the 'Modify IAM role' page for the instance i-03c712f819be636e7. The 'IAM role' dropdown is set to 'Ec2Instancerole'. A button to 'Create new IAM role' is visible. The 'Update IAM role' button at the bottom right is highlighted in orange. The top navigation bar shows the instance ID and the region as Singapore.

3. Now, create an S3 bucket and block all public access to it.

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Asia Pacific (Singapore) ap-southeast-1

Bucket name [Info](#)
mybucketmount01

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

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

[Choose bucket](#)

Format: s3://bucket/prefix

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

Block Public Access settings for this bucket

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 this bucket and its 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 this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning the setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLS)**
AWS will ignore new access control lists (ACLS) that grant public access to buckets and objects, and prevent creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLS)**
AWS will ignore all access to buckets and objects that are granted using existing access control lists (ACLS).
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versions is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Enable

Tags - **optional** (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

Default encryption

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable

Enable

- Navigate to the bucket's access points, create a new access point by providing a name, selecting the private VPC, and ensuring that public access is blocked.

Create access point [Info](#)

Amazon S3 Access Points simplify managing data access at scale for shared datasets in S3. Access points are named network endpoints that are attached to buckets that you can use to perform S3 object operations. [Learn more](#)

Properties

AWS Region
Asia Pacific (Singapore) ap-southeast-1

Access point name

Access point names must be unique within the account for this Region and comply with the [rules for access point naming](#).

Bucket name
mybucketmount01

Network origin

- Virtual private cloud (VPC)
No internet access. Requests are made over a specified [Virtual Private Cloud](#).
- Internet

(?) The S3 console doesn't support accessing bucket resources using a virtual private cloud (VPC) access point. To access bucket resources from a VPC access point, use the AWS CLI, AWS SDK, or Amazon S3 REST API. [Learn more](#)

VPC ID

VPC ID must start with vpc-

Block Public Access settings for this Access Point

Public access is granted to buckets and objects through access controls (ACLs), bucket policies, access point policies, or all. These settings apply only to this Access Point. Before applying these settings, ensure that your applications will work correctly without public access. These settings can't be edited after the Access Point is created. [Learn more](#)

Block all public access

5. Create a VPC endpoint with an S3 gateway and attach it to the private instance's route table.

Create endpoint [Info](#)

Create the type of VPC endpoint that supports the service, service network or resource to which you want to connect.

Endpoint settings

Specify a name and select the type of endpoint.

Name tag - optional

S3 configuration

Type

- AWS services
- EC2 Instance Connect Endpoint
- Endpoint services that use NLBs and GWLBs

AWS services

- AWS Lambda, Partner services
- AWS Marketplace services
- Resources - New
- Service networks - New

Services (1/4)

Service Name	Owner	Type	Service Region
com.amazonaws.ap-southeast-1.s3	amazon	Gateway	-
com.amazonaws.ap-southeast-1.s3	amazon	Interface	-
com.amazonaws.ap-southeast-1.s3-endpoint	amazon	Interface	-
com.amazonaws.s3-global.accesspoint	amazon	Interface	-

Network settings

Select the VPC in which to create the endpoint

VPC

Create the VPC endpoint in the VPC in the same AWS Region from which you will access a resource.

vpc-04c463f0d67a6044e (PrivateVPC)

Route tables (1/3)

Name	Route Table ID	Main	Associated Id
Private_route_table	rtb-0e7aefcfc239096d (Private_route_table)	Yes	subnet-03d70a7602345420 (private_subnet)
RDS-Pvt-rt	rtb-096dcf1f15ca3c30 (RDS-Pvt-rt)	No	2 subnets

The screenshot shows the AWS VPC Endpoint service interface. At the top, there's a search bar and navigation links for 'Singapore' and 'varun @ aws'. Below the header, a table lists endpoints. One endpoint, 'S3configuration', is selected and highlighted in blue. The table columns include Name, VPC endpoint ID, Endpoint type, Status, Service name, and Service net. The endpoint details are shown below the table, including its creation time (Sunday, December 29, 2024 at 01:22:46 GMT+5:30), service name (com.amazonaws.ap-southeast-1.s3), and endpoint type (Gateway).

- Wait for a few minutes until the VPC endpoint is automatically updated in the attached route table and the outbound rule of the private instance's security group.

The screenshot shows the AWS Route Tables service interface. At the top, there's a search bar and navigation links for 'Singapore' and 'varun @ aws-sanjay-s'. Below the header, a table lists route tables. One route table, 'Private_route_table', is selected and highlighted in blue. The table columns include Name, Route table ID, Explicit subnet assoc..., Edge associations, Main, VPC, and Owner ID. The route table details are shown below the table, including its creation time (less than a minute ago), service name (vpc-04c463f0d67a6044e | Priva...), and owner ID (590183945701).

The screenshot shows the AWS VPC Security Groups console. The top navigation bar includes options like [Alt+S], Actions, Export security groups to CSV, Create security group, and a user dropdown for varun @ aws-sanjay-s. Below the navigation is a search bar with placeholder text "Find resources by attribute or tag". The main table lists security groups with columns for Name, Security group ID, Security group name, VPC ID, and Description. One row is selected, "Privateec2instanceSG", which has a description: "Security group attached to my-website-db to allow EC2 instances with specific IP addresses to connect to my-database. Modified on 2023-06-20 10:00 UTC by varun". Below the table, a message says "sg-054a9be0b2065a766 - default". At the bottom, tabs for Details, Inbound rules, Outbound rules (which is selected), Sharing - new, VPC associations - new, and Tags are visible.

Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination
sgr-0b95e7fbfeef95524	-	HTTPS	TCP	443	pl-6fa54006 (com.amazonaws.ap-southeast-1.s3)	

- Goto Edit bucket policy give the below code which will Locking down Amazon S3 bucket to VPC access only.

Note: Outside VPC even from the console we can't access it.

The screenshot shows the AWS S3 Bucket Policy editor. The top navigation bar includes a search bar, Actions, Export bucket policy to CSV, Create bucket policy, and a user dropdown for varun @ aws-sanjay-s. The left sidebar shows the bucket structure: on S3 > Buckets > mybucketmount01 > Edit bucket policy. The main area displays a JSON policy document:

```

Bucket ARN: arnaws:s3:::mybucketmount01

Policy:
1  {
2    "Version": "2012-10-17",
3    "Id": "S3BucketPolicyVPCAccessOnly",
4    "Statement": [
5      {
6        "Sid": "DenyIfNotFromAllowedVPC",
7        "Effect": "Deny",
8        "Principal": "*",
9        "Action": [
10          "s3:GetObject",
11          "s3>ListBucket",
12          "s3:PutObject"
13        ],
14        "Resource": [
15          "arnaws:s3:::mybucketmount01",
16          "arnaws:s3:::mybucketmount01/*"
17        ],
18        "Condition": {
19          "StringEquals": {
20            "aws:SourceVpc": "vpc-04c463f0d67a6044e"
21          }
22        }
23      }
24    ]
25  }
  
```

Below the JSON code, there are buttons for "+ Add new statement" and "JSON Ln 25, Col 1". To the right, there are sections for "Edit statement" and "Select a statement". The "Select a statement" section contains the text "Select an existing statement in the policy or add a new statement." and a button "+ Add new statement".

- Now, from the private instance, create a mount point directory:
- sudo mkdir -p /mnt/s3bucket

- Mounting the S3 bucket Using IAM Role:

Note: Earlier, we have installed the s3fs package on the private instance.

- Mount the S3 bucket using `s3fs`:

```

Syntax: s3fs <bucket-name> /mnt/s3bucket -o iam_role=auto -o
url=https://s3.<region>.amazonaws.com -o use_path_request_style

s3fs mybucketmount01 /mnt/s3bucket -o iam_role=auto -o url=https://s3.ap-
southeast-1.amazonaws.com -o use_path_request_style

```

Replace:

- `<bucket-name>` with your S3 bucket name.
- `<region>` with your AWS region (e.g., `us-east-1`).

Options Explained:

- `-o iam_role=auto` : Automatically retrieves credentials from the attached IAM role.
- `-o url` : Directs requests to the S3 service in the specified region.
- `-o use_path_request_style` : Enables path-style access, necessary for certain configurations.

10. Test the mount point before making it persistent.

- Check the file system - `df -hT`
- Create a file on the instance and check if it is reflected in the S3 bucket.

```

root@ip-10-0-0-10:~# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/root       ext4     6.8G  2.3G  4.5G  33% /
tmpfs          tmpfs    479M   0  479M  0% /dev/shm
tmpfs          tmpfs    192M  884K  191M  1% /run
tmpfs          tmpfs    5.0M   0  5.0M  0% /run/lock
/dev/xvda16     ext4     881M   76M  744M  10% /boot
/dev/xvda15     vfat     105M   6.1M  99M   6% /boot/efi
tmpfs          tmpfs    96M   12K   96M   1% /run/user/1000
s3fs           fuse.s3fs 4.0G   0  4.0G  0% /mnt/s3bucket
root@ip-10-0-0-10:~# cd /mnt/s3bucket/
root@ip-10-0-0-10:/mnt/s3bucket# mkdir varun
root@ip-10-0-0-10:/mnt/s3bucket# ls
varun
root@ip-10-0-0-10:/mnt/s3bucket# 

```

i-0a9cff391a0905f79 (Web_server)

Public IPs: 13.203.76.114 Private IPs: 172.31.2.80

The screenshot shows the Amazon S3 console interface. At the top, there's a navigation bar with 'Amazon S3 > Buckets > mybucketmount01'. Below the navigation is a header with tabs: 'Objects' (which is active), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Under the 'Objects' tab, there's a sub-header 'Objects (1) Info' with a 'Copy S3 URI' button. A note below says 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)'. There's a search bar 'Find objects by prefix' and a 'Show versions' toggle. The main table lists one object: 'varun/' which is a 'Folder'. The table has columns for Name, Type, Last modified, Size, and Storage class. At the bottom right of the table area, there are navigation arrows and a gear icon.

11. Persist the mount in the “/etc/fstab” file:

- Edit the file: `sudo vi /etc/fstab`
- Add the following entry:

```
s3fs#<bucket-name> /mnt/s3bucket fuse  
_netdev,iam_role=auto,allow_other,use_path_request_style,url=https://s3.<region>.amazonaws.com 0 0
```

Options Explained:

- ` _netdev` : Ensures the mount is dependent on network services.
- ` allow_other` : Allows access to users other than the owner.

12. Save the file and test:

- `sudo mount -a`

Conclusion of this module:

We have successfully configured the S3 bucket on the private instance using the VPC endpoint, ensuring that data is transferred securely without exposure to the public internet.

Module – 05

Configure automatic power off and on for an EC2 instance using AWS Lambda and Event Bridge

1. Create a role that grants access to the Lambda service by attaching the "EC2 Full Access" policy.

The screenshot shows the AWS IAM Roles page. On the left, the navigation menu includes 'Identity and Access Management (IAM)', 'Access management', 'Access reports', and 'Logs'. The main content area displays the 'Lamdaroldforec2instance' role. The 'Summary' tab shows the creation date as December 22, 2024, 02:19 (UTC+05:30), last activity 3 days ago, ARN as arn:aws:iam::590183945701:role/Lamdaroldforec2instance, and maximum session duration as 1 hour. The 'Permissions' tab is selected, showing one managed policy: 'AmazonEC2FullAccess'. A 'Filter by Type' dropdown is set to 'All types'.

2. Now, create a Lambda function by selecting the runtime (type of language) and architecture, and then assign the "EC2 Full Access" role to it.

The screenshot shows the 'Create function' wizard. The first step, 'Create function', asks for the function name ('autopowerofandon') and runtime ('Python 3.13'). The second step, 'Basic information', shows the selected architecture as 'x86_64'. The third step, 'Permissions', shows the 'Change default execution role' section where the 'Lamdaroldforec2instance' role is selected. The final step, 'Additional Configurations', is partially visible at the bottom.

3. After creation, provide the following Python code to execute:

Note: The code below will toggle the instance power off/on based on the event-driven time.

```
import boto3
import logging

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)

INSTANCE_DETAILS = [
    {"instance_id": "i-0ddde86ee005cbab5", "region": "ap-southeast-1"}, # Private
    instance
    {"instance_id": "i-0e82be1a040318011", "region": "ap-south-1"} # Public instance
]

def toggle_instance_state(instance_id, region):
    ec2 = boto3.client('ec2', region_name=region)
    try:
        # Get the current state of the instance
        response = ec2.describe_instances(InstanceIds=[instance_id])
        state = response['Reservations'][0]['Instances'][0]['State']['Name']
        logger.info(f"Instance {instance_id} in region {region} is currently {state}")

        # Toggle the state
        if state == "stopped":
            ec2.start_instances(InstanceIds=[instance_id])
            logger.info(f"Started instance {instance_id} in region {region}")
    except Exception as e:
        logger.error(f"Error occurred while toggling instance {instance_id}: {str(e)}")
```

```

        elif state == "running":
            ec2.stop_instances(InstanceIds=[instance_id])
            logger.info(f"Stopped instance {instance_id} in region {region}")

        else:
            logger.warning(f"Instance {instance_id} in region {region} is in an unsupported
state: {state}")

        except Exception as e:
            logger.error(f"Failed to toggle state for instance {instance_id} in region {region}: {e}")

def lambda_handler(event, context):
    try:
        for instance in INSTANCE_DETAILS:
            toggle_instance_state(instance["instance_id"], instance["region"])

    return {"statusCode": 200, "body": "Toggled instance states successfully."}

    except Exception as e:
        logger.error(f"Error: {e}")

        return {"statusCode": 500, "body": str(e)}

```

Note: Edit the region and instance ID in the code according to the specific requirements.

-
4. Now, click on **Deploy** to deploy the code. Then, change the configuration time to 10 seconds for the Lambda function and test the code.

```

lambda_function.py
...
def toggle_instance_state(instance_id, region):
    ec2 = boto3.client('ec2', region_name=region)
    try:
        # Get the current state of the instance
        response = ec2.describe_instances(InstanceIds=[instance_id])
        state = response['Reservations'][0]['Instances'][0]['State']['Name']
    except:
        return {
            "statusCode": 500,
            "body": "An error occurred while trying to get the instance state."
        }
    if state == 'running':
        ec2.stop_instances(InstanceIds=[instance_id])
        state = 'stopped'
    else:
        ec2.start_instances(InstanceIds=[instance_id])
        state = 'running'
    response = ec2.describe_instances(InstanceIds=[instance_id])
    return {
        "statusCode": 200,
        "body": f"Toggled instance states successfully. {state}"
    }
}

```

Status: Succeeded
Test Event Name: (unsaved) test event
Response:
{
 "statusCode": 200,
 "body": "Toggled instance states successfully."
}
Function Logs:
START RequestId: 4366e0e-a7c2-45cf-b56e-ffe6428ee6cf Version: \$LATEST
[INFO] 2024-12-28T20:31:53.243Z 4366e0e-a7c2-45cf-b56e-ffe6428ee6cf Found credentials in environment variables.
[INFO] 2024-12-28T20:31:56.043Z 4366e0e-a7c2-45cf-b56e-ffe6428ee6cf Instance 1-03c712f819be636e7 in region ap-southeast-1 is currently running
[INFO] 2024-12-28T20:31:57.347Z 4366e0e-a7c2-45cf-b56e-ffe6428ee6cf Stopped instance 1-03c712f819be636e7 in region ap-southeast-1
[INFO] 2024-12-28T20:31:58.402Z 4366e0e-a7c2-45cf-b56e-ffe6428ee6cf Instance 1-0a9cff391a005f79 in region ap-southeast-1 is currently running

5. Create an Event Bridge Scheduler with the following configuration:

- Set the schedule to recur daily.
- Set the time zone.
- Use the cron expression: (0 6,18 ? * * *).
- Turn off the flexible time window.

Note: The cron expression will trigger daily at 6 AM and 6 PM.

Specify schedule detail

Schedule name and description

Schedule name: (Required)

Description (optional):

Schedule group: (Optional)

Occurrences: Recurring schedule

Time zone: (Optional)

Schedule type: Cron-based schedule

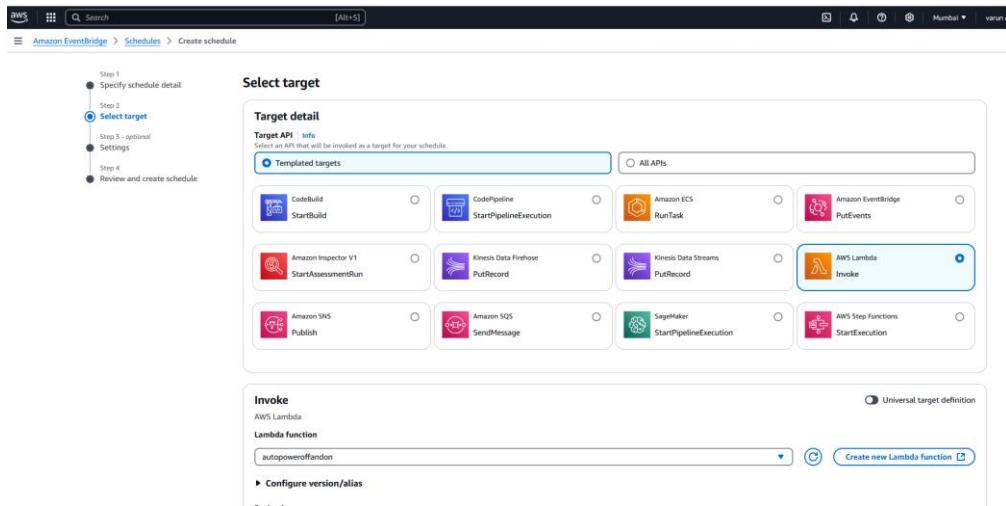
Cron expression: (Required)

Next 10 trigger dates:

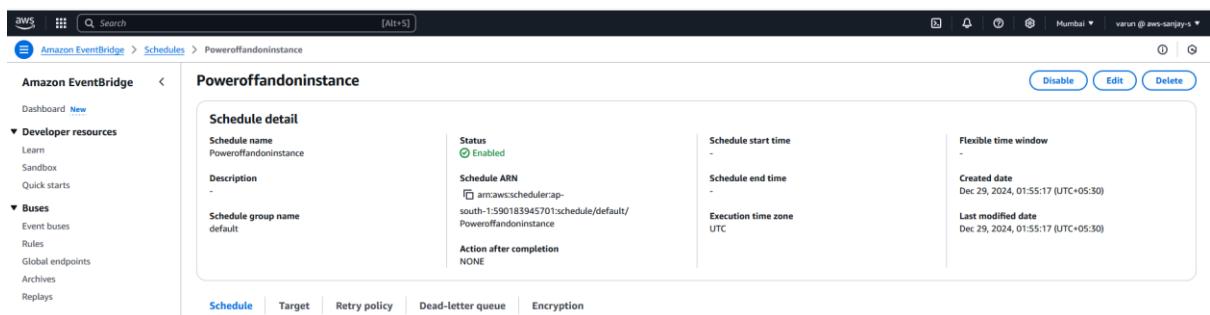
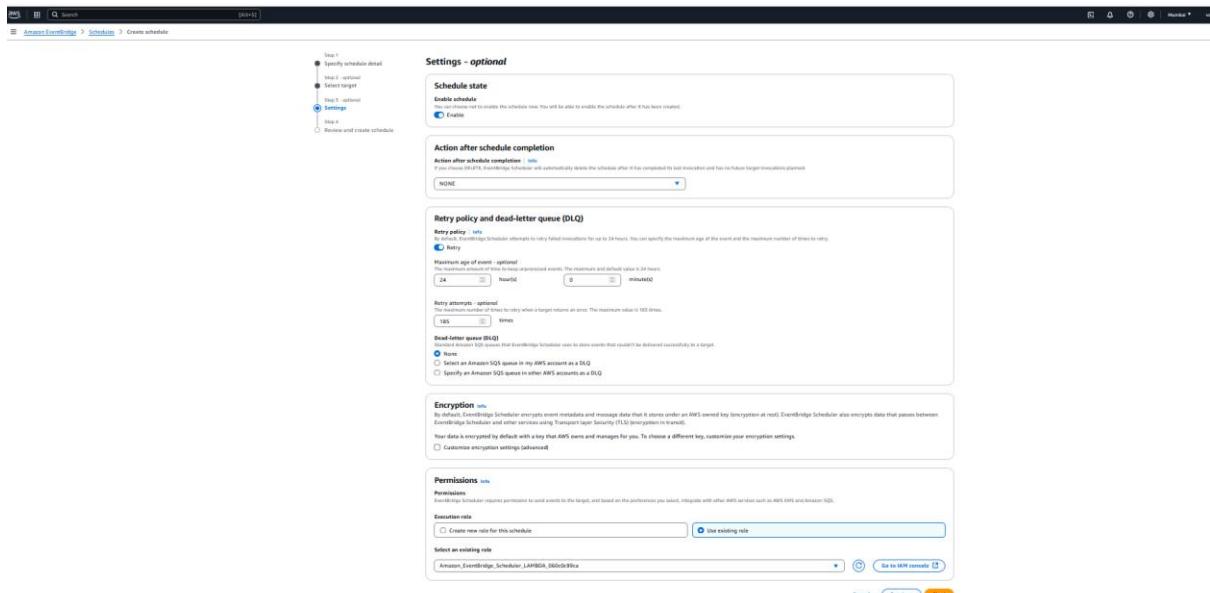
Flexible time window:

Timeframe:

6. After specifying the schedule, select the target as **Lambda Function**.



7. Then, configure the basic settings and click on **Create**.



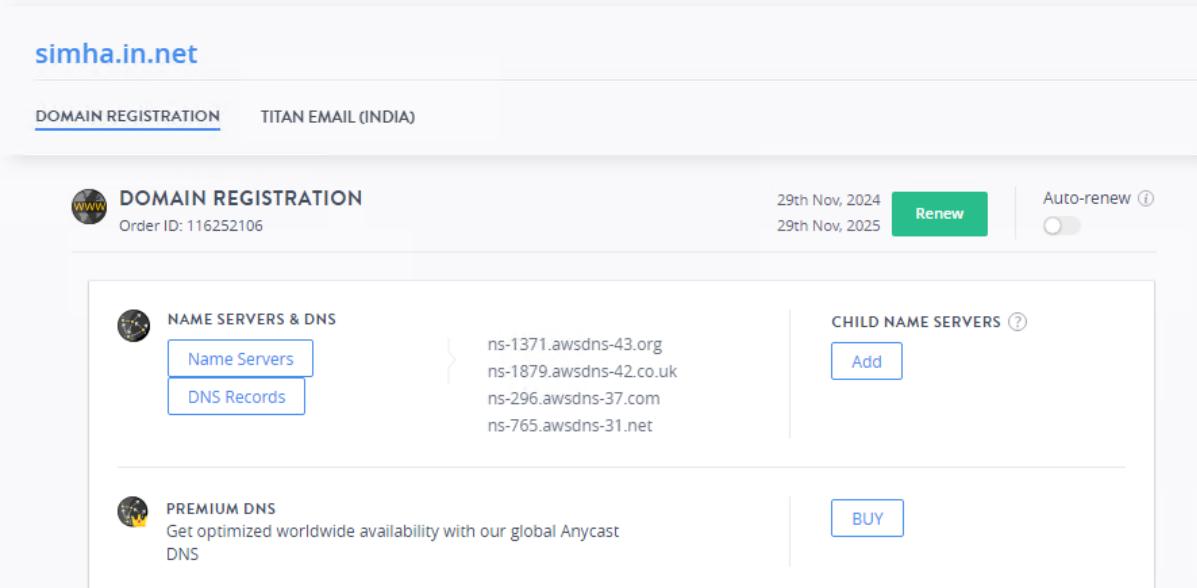
Conclusion of this module:

We have successfully set up automation that will power off and power on the instance at 6 AM and 6 PM UTC daily, triggered by the Lambda function through the Event Bridge scheduler, helping to reduce resource costs.

Module – 06

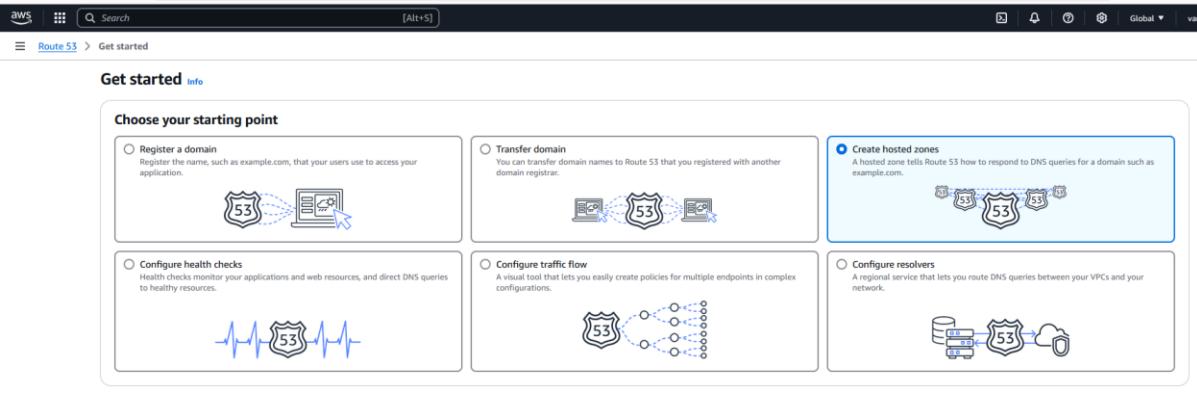
Setting up a secure web application using Route 53, a Load Balancer, and ACM

1. Purchased or registered a domain (in this case, it was purchased through BigRock, a third-party provider).



The screenshot shows the BigRock domain registration page for the domain `simha.in.net`. At the top, there are tabs for **DOMAIN REGISTRATION** and **TITAN EMAIL (INDIA)**. The **DOMAIN REGISTRATION** tab is active, showing the domain name `simha.in.net`, Order ID `116252106`, and expiration dates `29th Nov, 2024` and `29th Nov, 2025`. A green **Renew** button and an **Auto-renew** toggle switch are present. Below this, the **NAME SERVERS & DNS** section lists four AWS Route 53 NS records: `ns-1371.awsdns-43.org`, `ns-1879.awsdns-42.co.uk`, `ns-296.awsdns-37.com`, and `ns-765.awsdns-31.net`. There is also a link to **PREMIUM DNS** with a **BUY** button. On the right, there is a section for **CHILD NAME SERVERS** with a **Add** button.

2. Create a hosted zone on Route 53 by providing the domain name you purchased and selecting the type as **Public Hosted Zone**.



The screenshot shows the **Get started** page for AWS Route 53. The top navigation bar includes the AWS logo, search bar, and global region selector. The main content area is titled **Get started** with a **Info** link. It features a grid of six cards under the heading **Choose your starting point**:

- Register a domain**: Register the name, such as `example.com`, that your users use to access your application. (Icon: shield with 53 and document)
- Transfer domain**: You can transfer domain names to Route 53 that you registered with another domain registrar. (Icon: shield with 53 and three documents)
- Create hosted zones**: A hosted zone tells Route 53 how to respond to DNS queries for a domain such as `example.com`. (Icon: shield with 53 and four small shields)
- Configure health checks**: Health checks monitor your applications and web resources, and direct DNS queries to healthy resources. (Icon: shield with 53 and heart rate monitor)
- Configure traffic flow**: A visual tool that lets you easily create policies for multiple endpoints in complex configurations. (Icon: shield with 53 and network diagram)
- Configure resolvers**: A regional service that lets you route DNS queries between your VPCs and your network. (Icon: shield with 53 and network components)

At the bottom right, there are **Cancel** and **Get started** buttons.

Screenshot of the AWS Route 53 'Create hosted zone' configuration page.

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name [Info](#)
This is the name of the domain that you want to route traffic for.

Description - optional [Info](#)
This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 0/256

Type [Info](#)
The type indicates whether you want to route traffic on the internet or in an Amazon VPC.
 Public hosted zone
A public hosted zone determines how traffic is routed on the internet.
 Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags [Info](#)
Apply tags to hosted zones to help organize and identify them.
No tags associated with the resource.
[Add tag](#)
You can add up to 50 more tags.

[Cancel](#) [Create hosted zone](#)

- After creating the hosted zone, copy the **NS (Name Server)** record values and paste them into the **Name Server** settings where you purchased the domain.

Screenshot of the AWS Route 53 'Hosted zone details' page for simha.in.net.

Records (2) [Info](#)

The following table lists the existing records in simha.in.net. You can't delete the SOA record or the NS record named simha.in.net.

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (seconds)
simha.in.net	NS	Simple	-	ns-1509.awsdns-60.org. ns-446.awsdns-55.com. ns-1779.awsdns-30.co.uk. ns-887.awsdns-46.net.	172800
simha.in.net	SOA	Simple	-	ns-1509.awsdns-60.org. aws...	900

Record details

[Edit record](#)

Record name: simha.in.net
Record type: NS
Value:
 ns-1509.awsdns-60.org.
 ns-446.awsdns-55.com.
 ns-1779.awsdns-30.co.uk.
 ns-887.awsdns-46.net.

Alias: No
TTL (seconds): 172800
Routing policy: Simple

simha.in.net

DOMAIN REGISTRATION TITAN EMAIL (INDIA)

NAME SERVERS & DNS
For: simha.in.net

Provide at least 2 name servers

ns-1509.awsdns-60.org *

ns-446.awsdns-55.com *

ns-1779.awsdns-30.co.uk

ns-887.awsdns-46.net

+Add Name Servers

UPDATE NAME SERVERS Cancel Changes

- Now, create a certificate in ACM for the region where the public instance is located. Choose the certificate type as **Public** and leave the remaining settings as default.

aws [Alt+S]

AWS Certificate Manager > Certificates > Request certificate

Request certificate

Certificate type info
ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

Request a private certificate
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit [AWS Private Certificate Authority](#).

Cancel **Next**

Domain names
Provide one or more domain names for your certificate.
Fully qualified domain name [info]
simha.in.net

Add another name to this certificate
You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

Validation method [info]
Select a method for validating domain ownership.
 DNS validation - recommended
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

Email validation
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

Key algorithm [info]
Select an encryption algorithm. Some algorithms may not be supported by all AWS services.
 RSA 2048
RSA is the most widely used key type.

ECCDSA P 256
Equivalent in cryptographic strength to RSA 3072.

ECCDSA P 384
Equivalent in cryptographic strength to RSA 7680.

Tags [info]
No tags associated with the resource.
Add new tag
You can add up to 50 tags.

Cancel Previous Request

5. After the certificate is created, copy the **CNAME** record and then add it to Route 53.

aeed623f-41e4-43a9-86d8-ca7309ab8891

Certificate status

Identifier aeed623f-41e4-43a9-86d8-ca7309ab8891	Status Pending validation [info]
ARN arn:aws:acm:ap-south-1:590183945701:certificate/aeed623f-41e4-43a9-86d8-ca7309ab8891	
Type Amazon Issued	

Domains (1)

Domain	Status	Renewal status	Type	CNAME name
simha.in.net	Pending validation	-	CNAME	2187974508a90af77e2ac5bf272aa343.simha.in.net

Details

In use No	Serial number N/A	Requested at December 29, 2024, 02:13:26 (UTC+05:30)	Renewal eligibility Ineligible
Domain name simha.in.net	Public key info RSA 2048	Issued at N/A	
Number of additional names	Signature algorithm	Not before	

Create DNS records in Amazon Route 53 (1/1)

Search domains [Alt+S] 1 match

Validation status = Pending validation X Validation status = Failed X Is domain in Route 53? = Yes X Clear filters

<input checked="" type="checkbox"/> Domain	Validation status	Is domain in Route 53?
simha.in.net	Pending validation	Yes

Cancel Create records

Records (3) | DNSSEC signing | Hosted zone tags (0)

Records (3) info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Record name	Type	Routing policy	Differentiator	Alias	Value/Route traffic to	TTL (s)	Health status	Evaluation context
simha.in.net	NS	Simple	-	No	ns-1509.awsdns-60.org. ns-446.awsdns-55.com. ns-1779.awsdns-30.co.uk. ns-887.awsdns-46.net.	172800	-	-
simha.in.net	SOA	Simple	-	No	ns-1509.awsdns-60.org. awsdns-hostmaster.amazon.com.	900	-	-
_fc..._2187974508a90af77e2ac5bf272aa343.simha.in.net	CNAME	Simple	-	No	_fc...eb3dc1a4b582207fdff514cd3a2.zfyfvmc...	300	-	-

- Wait for a few minutes for the validation to complete, and the status will change to **Issued** in the ACM service.

Certificates (1)

Certificate ID	Domain name	Type	Status	In use	Renewal eligibility	Key algorithm
aeed623f-41e4-43a9-86d8-ca7309ab8891	simha.in.net	Amazon Issued	Issued	No	Ineligible	RSA 2048

- Now, create the **Target Group** for the Load Balancer, selecting the target type as **Instance**, and leave the remaining settings at their default values.

Step 1: Specify group details

Step 2: Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

- Instances
 - Supports load balancing to instances within a specific VPC.
 - Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.
- IP addresses
 - Supports load balancing to VPC and on-premises resources.
 - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
 - Offers flexibility with microservice based architectures, simplifying inter-application communication.
 - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- Lambda function
 - Facilitates routing to a single Lambda function.
 - Accessible to Application Load Balancers only.
- Application Load Balancer
 - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
 - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol & Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. You can change either after creation.

Protocol: HTTP

Port: 80

IP address type: Only targets with the indicated IP address type can be registered in this target group.

IPv4

An instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is one that will be assigned to the target.

IPv6

An instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC: Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

Protocol version:

HTTP1

Send requests to targets using HTTP1.1. Supported when the request protocol is HTTP1.1 or HTTP2.

HTTP2

Send requests to targets using HTTP2. Supported when the request protocol is HTTP2 or gRPC, but gRPC-specific features are not available.

gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks: The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol: HTTP

Health check path: Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred. /

Use 1-1024 characters allowed.

Advanced health check settings

Attributes: Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Tags - optional: Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Next

8. Here, selects the instance that needs to be registered as the target for the target group.

Step 1: Select group details

Step 2: Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (1)

Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID	Last modified
i-0a9cf391a0905f79	Web_server	Running	default	ap-south-1b	172.31.2.80	subnet-0c77452e16c19ded6	Dec 29, 2024

Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include pending below

1 selection is now pending below. Include more or register targets when ready.

Review targets

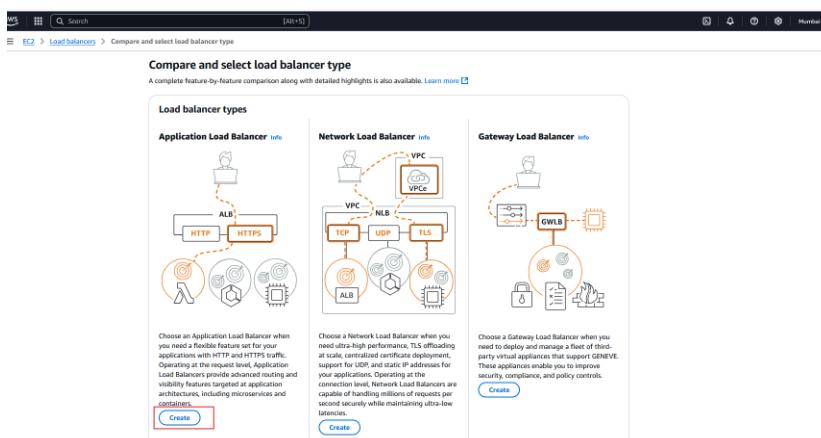
Targets (1)

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-0a9cf391a0905f79	Web_server	80	Running	default	ap-south-1b	172.31.2.80	subnet-0c77452e16c19ded6	December 29, 2024, 02:03 (UTC+05:30)

1 pending

Create target group

9. After creating the target group, create the **Application Load Balancer** to allow secure communication to the instance.



10. Configure the internet-facing and choose the VPC and its minimum two availability zone where instance lies.

Create Application Load Balancer info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme info
Scheme can't be changed after the load balancer is created.

Internet-facing
• Severs internet-facing traffic.
• Has public IP addresses.
• Only traffic is publicly resolvable.
• Requires a public subnet.

Internal
• Serves internal traffic.
• Has private IP addresses.
• Only traffic is publicly resolvable.
• Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with internet-facing load balancers only.

Network mapping info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups. For a new VPC, create a VPC.

PublicVPC
vpc-02d14d237f
(IPv4 CIDR: 172.31.0.0/16)

Mappings info
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

Availability Zones

- ap-south-1a (ap-s1-az1)
Subnet: subnet-058181e2532654e9 (IPv4 subnet CIDR: 172.31.32.0/20)
- ap-south-1b (ap-s1-az3)
Subnet: subnet-0c77452e16c19ded5 (IPv4 subnet CIDR: 172.31.0.0/20)
- ap-south-1c (ap-s1-az2)

Security groups info
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.

Security groups

11. On the **Listeners and Routing** section, add both **HTTP** and **HTTPS** listeners. For both listeners, set the default action to "forward to target group".

Security groups info
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.

Security groups

Select up to 5 security groups

sg-014032f6d216c1ca1 VPC: vpc-0ab0222cd31d0227f

Listeners and routing info
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP:80

Protocol	Port	Default action	Remove
HTTP	80	LBTarget Target type: Instance, IPv4	
Create target group			

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag
You can add up to 50 more tags.

Listener HTTPS:443

Protocol	Port	Default action	Remove
HTTPS	443	LBTarget Target type: Instance, IPv4	
Create target group			

12. Since we have added the **HTTPS listener**, choose the **default security policy**. For the certificate, select **from ACM** and choose the certificate that was created earlier. Then, click on Create.

Secure listener settings

Security policy: ELBSecurityPolicy-TLS13-1-2-2021-06 (recommended)

Default SSL/TLS server certificate: From ACM (selected)

Certificate source: From ACM (selected)

Client certificate handling

Mutual authentication (mTLS): Mutual TLS (Transport Layer Security) authentication offers two-way peer authentication. It adds a layer of security over TLS and allows your services to verify the client that's making the connection.

Summary

Basic configuration: WebLB, Internet-facing, IPv4

Security groups: Default (sg-014b32feff266c6a1)

Network mapping: VPC (vpc-0ab922cd1d02271), PublicSubnet

Listeners and routing:

- HTTP:80 defaults to LBTTarget
- HTTPS:443 defaults to LBTTarget

Secure listener settings:

- ELBSecurityPolicy-TLS13-1-2-2021-06
- simha.in.net (From ACM)

Create workflow and status

Server-side tasks and status: After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Create lead balancer

13. This will take a few minutes to set up. Once the process is complete, you can edit the **HTTP listener** settings.

Listeners and rules (1/2)

Protocol: HTTP:80

Forward to target group:

- LBTTarget (100%)
- Target group stickiness: Off

Forward to target group:

- LBTTarget (100%)
- Target group stickiness: Off

Details

Load balancer type: Application

Status: Provisioning

VPC: vpc-0ab922cd1d02271

Hosted zone: ZP97RAFLXLYNZK

Availability Zones: subnets-0x77452e16c19d6d, ap-south-1b (ap-s1-a23), subnets-053819e2532bc54e9, ap-south-1a (ap-s1-a21)

DNS name info: WebLB.122315792.ap-south-1.elb.amazonaws.com (A Record)

14. While editing the HTTP listener, change the default action to "**Redirect to URL**" and set the URL port to **HTTPS – 443**. This ensures that whenever a user tries to access the application using HTTP, they will be automatically redirected to HTTPS for a secure connection.

15. After the Load Balancer is active, the target group will begin to register the instances, and the status of the instances should show as **healthy**. This indicates that the instances are properly receiving traffic and responding as expected through the Load Balancer.

The screenshot shows the AWS Elastic Load Balancing Target Groups interface. On the left, there's a sidebar with navigation links like 'Search', 'Target groups', 'LBTarget', 'Types', 'templates', 'tests', 'Logs', 'Instances', 'Hosts', 'Reservations', 'Log Store', 'Manager', 'Groups', 'IT Groups', 'Interfaces', and 'Lancing'. The main content area has a title 'LBTarget' and an 'Actions' dropdown. Below it, the 'Details' section shows the target type as 'Instance', protocol as 'HTTP: 80', and version as 'HTTP1'. It also lists the VPC and subnet information. A summary table shows 1 total target, 1 healthy, 0 unhealthy, 0 unused, 0 initial, and 0 draining. A section titled 'Distribution of targets by Availability Zone (AZ)' follows. At the bottom, tabs for 'Targets' (selected), 'Monitoring', 'Health checks', 'Attributes', and 'Tags' are visible. The 'Registered targets' section shows one target: 'i-0ab922cd31d02273' (Web_server, port 80, healthy, zone ap-south-1b). An info icon and a 'Register targets' button are present.

16. Now, in Route 53, create a record for HTTPS by setting the traffic routing value to the application Load Balancer and specifying its region, along with the Load Balancer's DNS name.

The screenshot shows the 'Create record' wizard in the AWS Route 53 console. The top navigation bar includes the search bar 'ACM-PCA', tabs for 'Route 53', 'Hosted zones', 'simha.in.net', and 'Create record', and icons for 'Global' and 'varun'. The main title is 'Create record' with a 'Info' link. A 'Switch to wizard' button is located in the top right corner.

Record name (Info): `subdomain` (highlighted), `simha.in.net`. A note says 'Keep blank to create a record for the root domain.'

Record type (Info): `A` – Routes traffic to an IPv4 address and some AWS resources.

Alias: A radio button is selected. Below it, the 'Route traffic to' section shows 'Alias to Application and Classic Load Balancer' and 'Asia Pacific (Mumbai)' as options. The selected option is `dualstack.WebLB-122315792.ap-south-1.elb.amazonaws.com`.

Routing policy (Info): `Simple routing`. To the right, there's an 'Evaluate target health' section with a radio button set to 'Yes'.

At the bottom right are buttons for 'Add another record' (blue outline) and 'Create records' (orange).

simha.in.net

simha.in.net [Info](#)

[Delete zone](#) [Test record](#)

Hosted zone details

[Records \(4\)](#) [DNSSEC signing](#) [Hosted zone tags \(0\)](#)

Records (4) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

<input type="checkbox"/> Record name	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health ...
simha.in.net	A	Simple	-	Yes	dualstack.weblb-122315792.ap-south-...	-	-
simha.in.net	NS	Simple	-	No	ns-1509.awsdns-60.org. ns-446.awsdns-55.com. ns-1779.awsdns-30.co.uk. ns-887.awsdns-46.net.	172800	-
simha.in.net	SOA	Simple	-	No	ns-1509.awsdns-60.org. awsdns-hostm...	900	-
_2187974508a90af77e2ac5bf272aa343.simha.in.net	CNAME	Simple	-	No	_fc6b0eb3dc1a4b582207fdf514cd3a2....	300	-

Note: We can check the DNS from all the location where it is accessible using the “whatsmydns” website:

whatsmydns.net Global DNS Propagation Checker

simha.in.net A Search

San Jose CA, United States Corporate West	3.110.238.161 35.154.105.152	✓
Kansas City, United States WholeSale Internet	3.110.238.161 35.154.105.152	✓
Dothan AL, United States Comodo		✗
Jamaica NY, United States Level 3 Communications	3.110.238.161 35.154.105.152	✓
Providence RI, United States Verizon	3.110.238.161 35.154.105.152	✓
London ON, Canada Golden Triangle	3.110.238.161 35.154.105.152	✓
Mexico City, Mexico Total Play	3.110.238.161 35.154.105.152	✓
Santa Cruz do Sul, Brazil Claro	3.110.238.161 35.154.105.152	✓
Paterna de Rivera, Spain ServiHosting	3.110.238.161 35.154.105.152	✓
Manchester, United Kingdom Acar B	3.110.238.161 35.154.105.152	✓
Lille, France Completel SAS	3.110.238.161 35.154.105.152	✓
Diemen, Netherlands Tele2 Nederland	3.110.238.161 35.154.105.152	✓
Dortmund, Germany Verizon	3.110.238.161 35.154.105.152	✓
Sassuolo, Italy Telecom Italia	3.110.238.161 35.154.105.152	✓
Cullinan, South Africa Liquid	3.110.238.161 35.154.105.152	✓
Vladivostok, Russia Rostelecom	3.110.238.161 35.154.105.152	✓
Rawalpindi, Pakistan CMPk	3.110.238.161 35.154.105.152	✓
Ariyalur, India Railwire	3.110.238.161 35.154.105.152	✓
Ranokk, Thailand	3.110.238.161	

DNS Propagation Checker
whatsmydns.net lets you instantly perform a DNS lookup to check a domain name's current IP address and DNS record information against multiple nameservers located in different parts of the world.

CRYPTO.COM SPORTS

80% off the Total 5G Unlimited plan. One line, \$25/mo. When you sign up for Auto Pay and bring your own phone. First month \$30. [Switch now](#) [total](#)

17. Wait for a few minutes for the secure connection to be established. After that, search for "simha.in.net" and ensure it appears as secure.

Note: The time for this process may vary; it could take only a few minutes, or sometimes longer.

STUDENT INFORMATION

First name:

Last name:

USN:

Branch: CSE ISE EC CV MECH

Semester: 1st Semester

Gender: Male Female

Mail Id:

Phone Number: XXX-XXX-XXXX

Select your birthday: mm/dd/yyyy

Select a Photo: Choose File No file chosen

Note: Only valid details are considered

Created by VSMP

Conclusion on this module:

We have successfully configured a secure HTTPS web application using the Load Balancer, Route 53 for DNS, and ACM for the SSL certificate.

Overall, of this Project:

Here I like to conclude that, I have designed and implemented a **secure multi-tier cloud architecture** leveraging a range of AWS services. This project showcases a deep focus on **security, cost optimization, and effective resource utilization** while remaining adaptable for future scalability.

Core Components and Configuration

Public Instance (Web Layer):

- Deployed in a **public subnet**, serving as the entry point for users.
- A dynamic website is hosted on this instance using **Git technology** to pull the codebase directly from a **GitHub repository**.
- Configured with HTTPS using **AWS Certificate Manager (ACM)** to ensure secure communication.

- This instance connects securely to a private database hosted in the private subnet.

Private Instance (Application and Data Layer):

- Configured in a **private subnet** in a different AWS region for enhanced security.
- **NAT Gateway** is deployed in the public subnet to provide **controlled internet access** to the private instances.
- **Amazon RDS (Relational Database Service)** hosts the backend database, providing automated backups, high availability, and reliability.
- An **S3 bucket with a VPC endpoint** is integrated into this instance for secure file storage, eliminating the need for internet exposure.

Key Features

✓ VPC Peering Connection:

- Established between the public and private VPCs to enable seamless, secure communication.
- Eliminates the need for direct internet access, ensuring a robust, private, and secure data flow.

✓ AWS Lambda and Event Bridge:

- Automated **powering on/off of instances** during non-business hours, significantly reducing operational costs.
- Event-driven architecture ensures efficient management of resources without manual intervention.

✓ Domain and SSL Configuration:

- The domain name was purchased through **Big Rock** and seamlessly configured in **Amazon Route 53** for DNS management.
- **AWS ACM (Certificate Manager)** provides SSL certificates, ensuring secure HTTPS communication for the website.

✓ Elastic Load Balancer (ELB):

- Used to distribute incoming traffic securely and efficiently.

- Enables high availability and prepares the architecture for potential future scalability, such as auto-scaling.

Why This Architecture Stands Out

Security-First Approach:

- Private resources in the private subnet have **no direct internet access**, ensuring data and resource isolation.
- SSL encryption safeguards all communication between the website and users.

Cost Optimization:

- **Lambda-driven automation** for powering resources on/off during non-peak hours reduces operational costs.
- Efficient use of VPC endpoints and multi-region VPC peering minimizes data transfer costs.

Future-Ready Design:

- The architecture is built to scale with growing workloads by enabling auto-scaling and other cloud-native features like **WAF (Web Application Firewall)** for enhanced security.
-

Business Applications and Use Case

This multi-tier architecture is ideal for businesses that need secure, reliable, and cost-effective web application hosting in the cloud. Benefits include:

- **High Security:** Private and public layers ensure data isolation and controlled communication.
 - **Cost Efficiency:** Automated management minimizes wasteful spending.
 - **Scalability:** Built to scale with business growth.
-

Personal Insights and Future Enhancements

This project demonstrates how AWS services can be creatively combined to deliver secure, efficient, and scalable solutions. Looking ahead, I plan to:

- Enable **Auto Scaling Groups** to handle fluctuating workloads.

- Integrate **AWS WAF** to improve application security against evolving threats.
- Incorporate **CloudWatch monitoring** for better insight into system performance.

What are your thoughts on this architecture?

Have you worked on similar projects? I'd love to hear your feedback and ideas on how we can enhance this design even further!

#AWS #CloudArchitecture #VPC #RDS #Lambda #S3 #Route53 #Automation #SSL
#Scalability