

Report: Domain name and AWS Route53

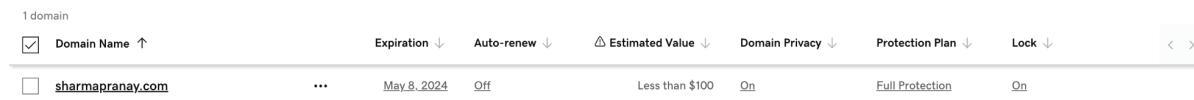
Name: Pranay Sharma

G#: G01393761

Introduction: In this report, I will outline the detailed process of setting up my website using my domain name 'sharmapranay.com' and hosting it on an EC2 instance running Apache2 using AWS Route53 hosted zone.

Step 1: Purchase a domain name

The first step in setting up a website is to purchase a domain name from a domain registrar such as GoDaddy.com. In this case, I purchased the domain name 'sharmapranay.com' from GoDaddy.

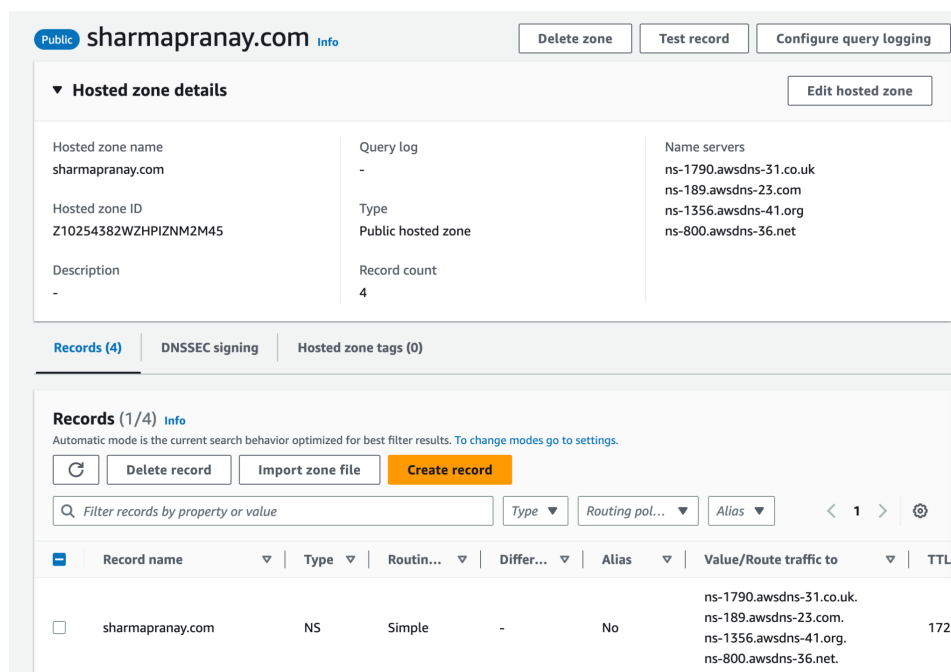


1 domain	Domain Name ↑	Expiration ↓	Auto-renew ↓	Estimated Value ↓	Domain Privacy ↓	Protection Plan ↓	Lock ↓	< >
<input type="checkbox"/>	sharmapranay.com	May 8, 2024	Off	Less than \$100	On	Full Protection	On	

Step 2: Create a hosted zone in AWS Route53

After purchasing the domain name, the next step is to create a hosted zone in AWS Route53. This can be done by following these steps:

1. I logged in to the AWS Learners Lab Console and navigated to the Route53 dashboard.
2. I clicked on "Create Hosted Zone" and entered the domain name that I had purchased from the domain registrar.
3. I clicked on "Create" to create the hosted zone.



Public sharmapranay.com [Info](#) [Delete zone](#) [Test record](#) [Configure query logging](#)

Hosted zone details [Edit hosted zone](#)

Hosted zone name sharmapranay.com	Query log -	Name servers ns-1790.awsdns-31.co.uk ns-189.awsdns-23.com ns-1356.awsdns-41.org ns-800.awsdns-36.net
Hosted zone ID Z10254382WZHP1ZNM2M45	Type Public hosted zone	
Description -	Record count 4	

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

Records (1/4) [Info](#)
Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

[Refresh](#) [Delete record](#) [Import zone file](#) [Create record](#)

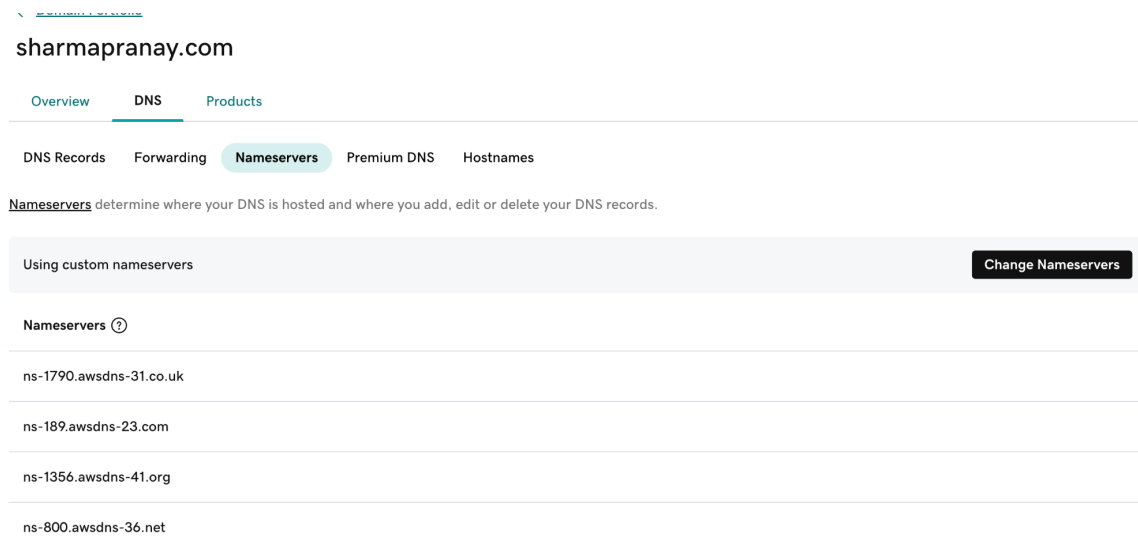
[Type](#) [Routing pol...](#) [Alias](#) [< 1 >](#) [Settings](#)

<input type="checkbox"/>	Record name	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL
<input type="checkbox"/>	sharmapranay.com	NS	Simple	-	No	ns-1790.awsdns-31.co.uk. ns-189.awsdns-23.com. ns-1356.awsdns-41.org. ns-800.awsdns-36.net.	172

Step 3: Update the nameservers at the domain registrar

Once the hosted zone was created, I updated the nameservers in the domain registrar to point to the AWS Route53 hosted zone. This can be done by following these steps:


1. I logged in to my GoDaddy account.
2. I found the domain name that I had purchased and clicked on "Manage DNS".
3. I updated the nameservers by adding the nameservers of the AWS Route53 hosted zone.
4. I saved the changes.



Step 4: Launch an EC2 instance with Ubuntu server and install Apache2

To host a website, an EC2 instance running Apache2 must be launched. In this case, I launched an EC2 instance with Ubuntu server, and Apache2 was installed by following these steps:

1. I logged in to the AWS Management Console and navigated to the EC2 dashboard.
2. I clicked on "Launch Instance" and selected Ubuntu server.
3. I followed the prompts to configure the instance.
4. I connected to the instance via SSH and installed Apache2 by running the command "sudo apt-get install apache2".



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

Step 5: Transfer website files to the EC2 instance

Once the EC2 instance was up and running with Apache2 installed, I transferred the website files to the EC2 instance. This can be done by following these steps:

1. I connected to the instance via SSH.
2. I transferred the website files to the `/var/www/html/` directory on the instance using the command `"scp"`.

```

ubuntu@ip-172-31-16-179:~$ cd /var/www/html/
ubuntu@ip-172-31-16-179:/var/www/html$ ls -lrth
total 28K
-rw-r--r-- 1 root root 11K May  9 00:55 index-backup.html
-rw-r--r-- 1 root root 5.5K May  9 01:13 index.html
drwxr-xr-x 2 root root 4.0K May  9 01:17 css
drwxr-xr-x 2 root root 4.0K May  9 01:18 img
ubuntu@ip-172-31-16-179:/var/www/html$ cd css/
ubuntu@ip-172-31-16-179:/var/www/html/css$ ls -lrth
total 8.0K
-rw-r--r-- 1 ubuntu ubuntu 1.6K May  9 01:16 home.css
-rw-r--r-- 1 ubuntu ubuntu 3.4K May  9 01:16 main.css
ubuntu@ip-172-31-16-179:/var/www/html/css$ cd ../img/
ubuntu@ip-172-31-16-179:/var/www/html/img$ ls -lrth
total 3.1M
-rw-r--r-- 1 ubuntu ubuntu 2.2M May  9 01:18 form_snapshot.png
-rw-r--r-- 1 ubuntu ubuntu 93K May  9 01:18 greet.png
-rw-r--r-- 1 ubuntu ubuntu 815K May  9 01:18 pink_panther.gif
ubuntu@ip-172-31-16-179:/var/www/html/img$

```

Step 6.1: Create a subdomain A record in the AWS Route53 hosted zone

To point the website to the EC2 instance using the **public IP address**, a subdomain record must be created in the AWS Route53 hosted zone. This can be done by following these steps:

1. I navigated to the AWS Route53 dashboard and clicked on the hosted zone that I had created earlier.
2. I clicked on "Create Record Set" and entered the subdomain name, in this case, "home.sharmapranay.com".
3. I selected "A - IPv4 address" for the record type.
4. I entered the IP address of the EC2 instance in the "Value" field.
5. I clicked on "Create" to create the subdomain record.

The screenshot shows the 'Create record' page in the AWS Route53 console. At the top, there's a 'Quick create record' section with a 'Switch to wizard' link. Below this, 'Record 1' is expanded, showing a 'Record name' field with 'hello' and a domain '.sharmapranay.com'. The 'Record type' is set to 'A - Routes traffic to an IPv4 address and some AWS resources'. There's an 'Alias' toggle which is currently off. The 'Value' field contains the IP address '54.234.210.200'. At the bottom, the 'TTL (seconds)' is set to '300' with buttons for '1m', '1h', and '1d'. The 'Routing policy' is set to 'Simple routing'. A note at the bottom states 'Recommended values: 60 to 172800 (two days)'.

Step 6.2: Create a subdomain Alias record in the AWS Route53 hosted zone

To point the website to the EC2 instance using the **Public IPv4 DNS**, a subdomain record must be created in the AWS Route53 hosted zone. This can be done by following these steps:

1. I navigated to the AWS Route53 dashboard and clicked on the hosted zone that I had created earlier.
2. I clicked on "Create Record Set" and entered the subdomain name, in this case, "hi.sharmapranay.com".
3. I selected "CNAME" for the record type.
4. I entered the public DNS for the EC2 machine.
5. I clicked on "Create" to create the subdomain record.

Create record [Info](#)

Quick create record

[Switch to wizard](#)

▼ Record 1

[Delete](#)Record name [Info](#) .sharmapranay.com

Keep blank to create a record for the root domain.

Record type [Info](#)

CNAME – Routes traffic to another domain name and to some AWS reso... ▼

☒ AliasValue [Info](#)

Enter multiple values on separate lines.

TTL (seconds) [Info](#)

1m

1h

1d

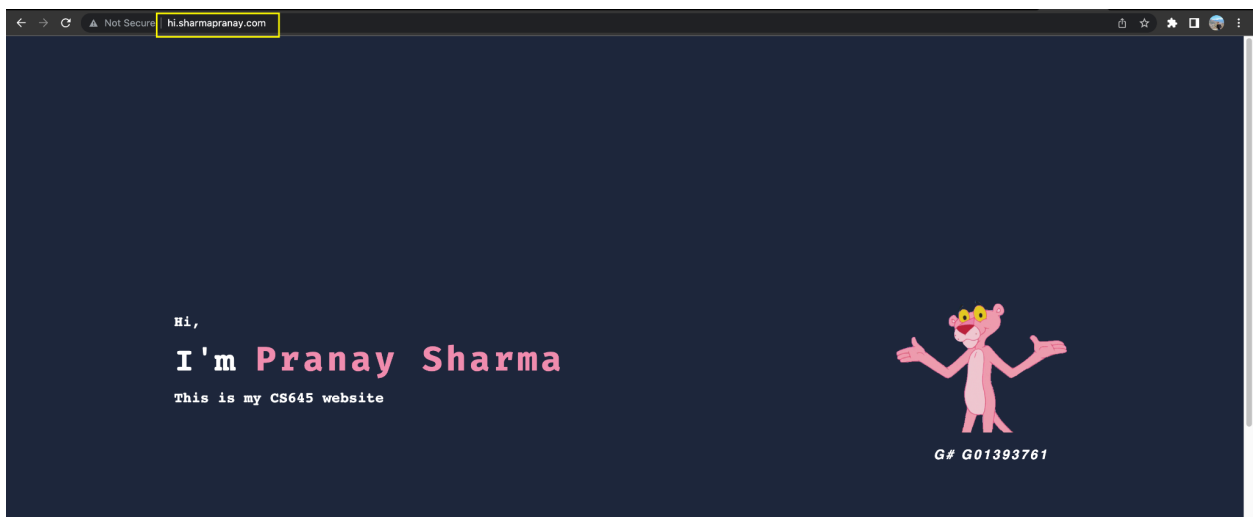
Routing policy [Info](#)

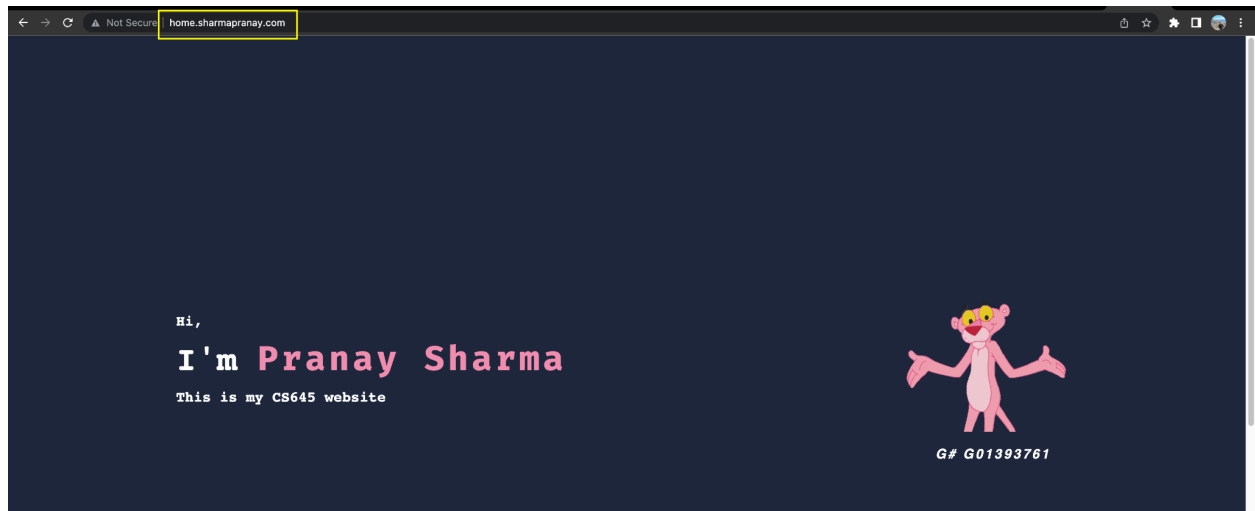
Simple routing ▼

Recommended values: 60 to 172800 (two days)

Step 7: Verify website is live

To confirm that the website is live, I visited the subdomain in a web browser. In this case, visiting "hi.sharmapranay.com" and "hello.sharmapranay.com" displayed the website that I had transferred to the EC2 instance.





Step 8: Create an Application Load Balancer:

To handle increased traffic to our website, we created another EC2 instance using the same process as in Step 4 and transferred the website files using Step 5. We then created an Application Load Balancer using the following steps:

1. We navigated to the Load Balancers section in the EC2 dashboard.
2. We clicked on "Create Load Balancer" and selected "Application Load Balancer".
3. We selected the VPC that our EC2 instances were running in and created a new security group for the load balancer.
4. We configured the load balancer to listen on port 80, which is the default port for HTTP traffic.
5. We created a target group containing the two EC2 instances that we had created and added them as targets.
6. We completed the load balancer creation process and verified that it was successfully created.

Target Group:

EC2 > Target groups > ec2-swe645

ec2-swe645

Actions

Details

am:aws:elasticloadbalancing:us-east-1:861892820384:targetgroup/ec2-swe645/8db7b03cdee4ec08

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0ff4f66362057bd41
IP address type	Load balancer		
IPv4	swe645-alb		

Total targets

Healthy

Unhealthy

Unused

Initial

Draining

2

2

0

0

0

0

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (2)

Filter resources by property or value

Deregister

Register targets

< 1 >

Instance ID	Name	Port	Zone	Health status	Health status details
i-051b95cecff4bb7d4	website2	80	us-east-1a	healthy	
i-0dc6409e82ea9db79	Website	80	us-east-1a	healthy	

Load Balancer:

EC2 > Load balancers > swe645-alb

swe645-alb

Actions

Details

am:aws:elasticloadbalancing:us-east-1:861892820384:loadbalancer/app/swe645-alb/8247ec59180f4b2c

Load balancer type	DNS name	Status	VPC
Application	swe645-alb-588356983.us-east-1.elb.amazonaws.com (A Record)	Active	vpc-0ff4f66362057bd41
IP address type	Scheme	Availability Zones	Hosted zone
IPv4	Internet-facing	subnet-083b2823c8197385d us-east-1a (use1-az4) subnet-07d2be9f69a6c6588 us-east-1b (use1-az6)	Z355XDOTRQ7X7K
Date created			
May 9, 2023, 20:30 (UTC-04:00)			

Listeners

Network mapping

Security

Monitoring

Integrations

Attributes

Tags

Listeners (1)

A listener checks for connection requests on its port and protocol. Traffic received by the listener is routed according to its rules.

Search

< 1 >

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL cert	Tags
HTTP:80	<div>Forward to target group</div> <ul style="list-style-type: none">ec2-swe645: 1 (100%)Group-level stickiness: Off	1 rule	ARN	Not applicable	Not applicable	0 tags

Step 9: Create a record with an alias to the Application Load Balancer:

To make the load balancer accessible via a subdomain, we created a record in our AWS Route53 hosted zone with an alias to the Application Load Balancer. The following steps were taken:

1. We navigated to the AWS Route53 dashboard and selected the hosted zone that we had created earlier.
2. We clicked on "Create Record Set" and entered "645lb" as the subdomain name.
3. We selected "A - IPv4 address" for the type of record.
4. We selected "Yes" for "Alias" and chose the load balancer that we had created in Step 8.
5. We completed the record creation process and verified that it was successfully created.

The screenshot shows the 'Create record' wizard in the AWS Route53 console. The 'Quick create record' tab is active, with a 'Switch to wizard' link in the top right. Below the tab, there's a 'Record 1' section with a 'Delete' button. The 'Record name' field contains '645lb' and the 'Record type' dropdown is set to 'A - Routes traffic to an IPv4 address and some AWS resources'. A note below the name field says 'Keep blank to create a record for the root domain.' The 'Alias' radio button is selected. The 'Route traffic to' dropdown is set to 'Alias to Application and Classic Load Balancer'. The 'US East (N. Virginia) [us-east-1]' region is selected. The 'Target' field contains the load balancer ARN: 'dualstack.swe645-alb-588356983.us-east-1.elb.amazonaws.com'. The 'Alias hosted zone ID' is 'Z35SXDOTRQ7X7K'. The 'Routing policy' is set to 'Simple routing'. The 'Evaluate target health' checkbox is checked. At the bottom right, there are 'Cancel' and 'Create records' buttons.

Create record [Info](#)

Quick create record [Switch to wizard](#)

▼ Record 1 [Delete](#)

Record name [Info](#) 645lb .sharma pranay.com

Record type [Info](#) A - Routes traffic to an IPv4 address and some AWS resources

Keep blank to create a record for the root domain.

☒ Alias

Route traffic to [Info](#) Alias to Application and Classic Load Balancer

US East (N. Virginia) [us-east-1]

Q dualstack.swe645-alb-588356983.us-east-1.elb.amazonaws.com X

Alias hosted zone ID: Z35SXDOTRQ7X7K

Routing policy [Info](#) Simple routing

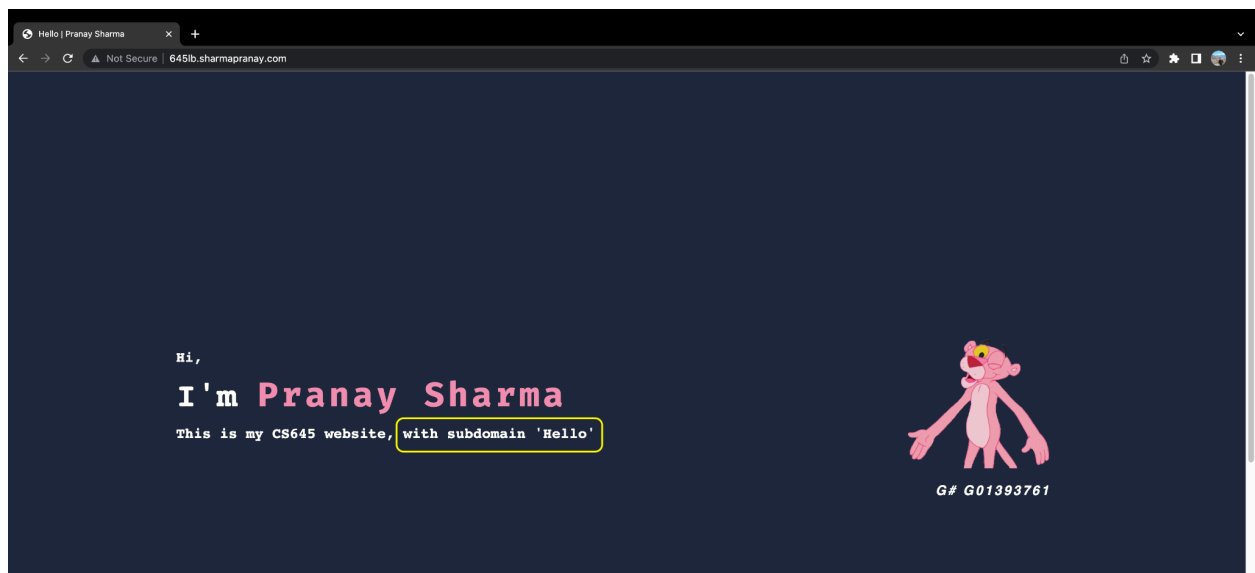
Evaluate target health ☒ Yes

[Add another record](#)

[Cancel](#) [Create records](#)

Step 10: Verify the subdomain points to both instances:

To confirm that the subdomain "645lb.sharma pranay.com" points to both instances, we updated the index.html file on each instance to include a unique identifier. We then visited the subdomain in a web browser and confirmed that the unique identifier was displayed from both instances, indicating that the load balancer was successfully distributing traffic to both instances.



Conclusion:

By following these steps, we have successfully set up a website hosted on an EC2 instance, which is accessible through a subdomain. Additionally, we have created an Application Load Balancer in AWS EC2 which allows for load balancing between multiple instances of the website. Overall, this setup provides a scalable and highly available solution for hosting a website on AWS.