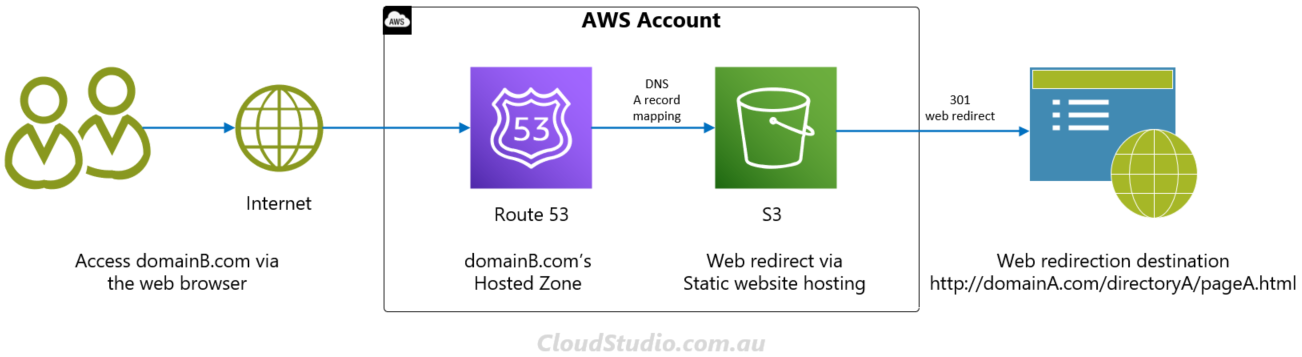
### **Setting Up a Static Website with Amazon S3 and Amazon Route 53**

### **(1)Objective**

The objective is to set up a static website hosted on Amazon S3 and configure DNS routing with Route 53 using a domain name purchased from Hostinger.



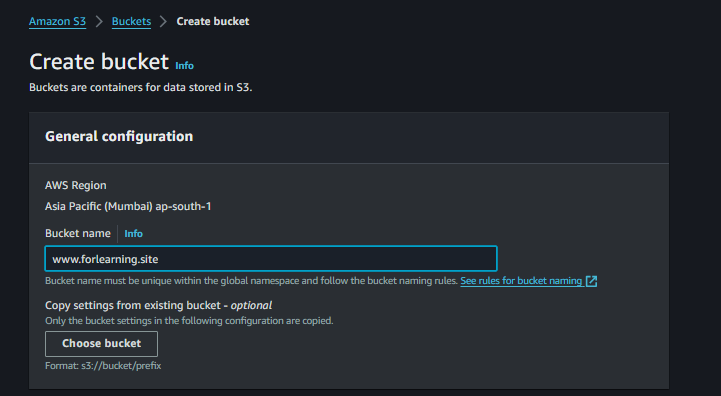
### **Architecture Flow**

1. **Amazon S3 Bucket**: Create an S3 bucket to store and host your static website files.
2. **Website Configuration**: Configure the S3 bucket for static website hosting.
3. **Route 53**: Set up Amazon Route 53 to manage the DNS records for your domain.
4. **Domain Configuration**: Point your Hostinger domain to Route 53 to handle the DNS queries.

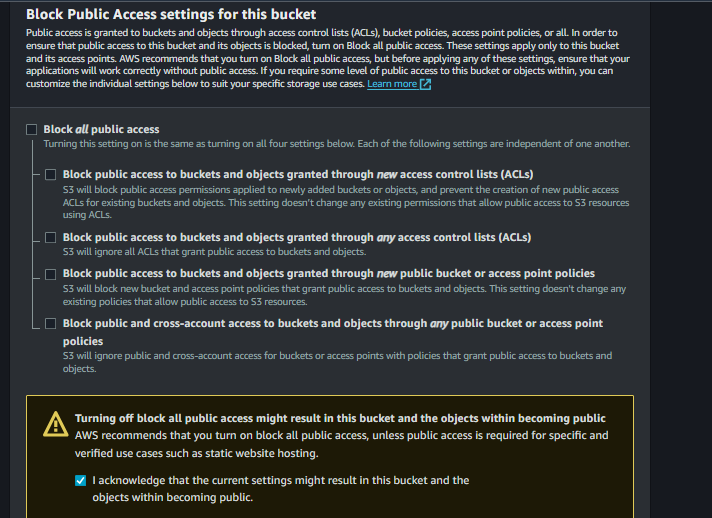
### **Steps**

#### **1. Set Up the Amazon S3 Bucket**

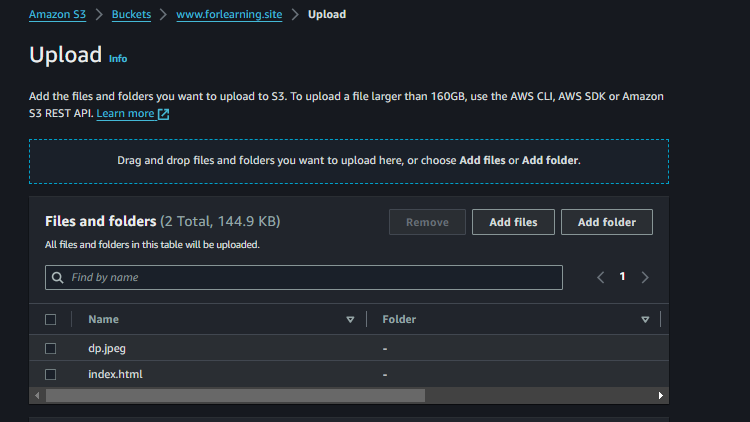
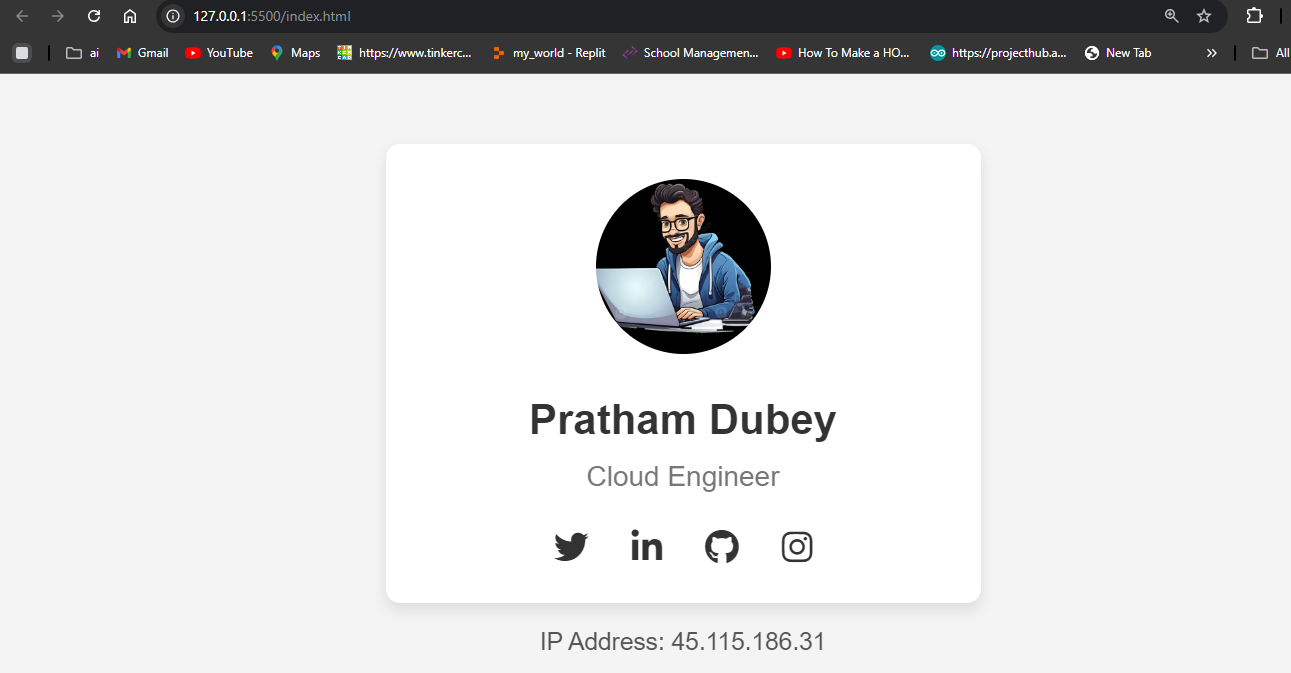
1. **Create an S3 Bucket**:
   * Go to the [Amazon S3 Console](https://console.aws.amazon.com/s3/).
   * Click on **Create bucket**.
   * Enter a unique bucket name that matches your domain name (e.g., www.example.com).



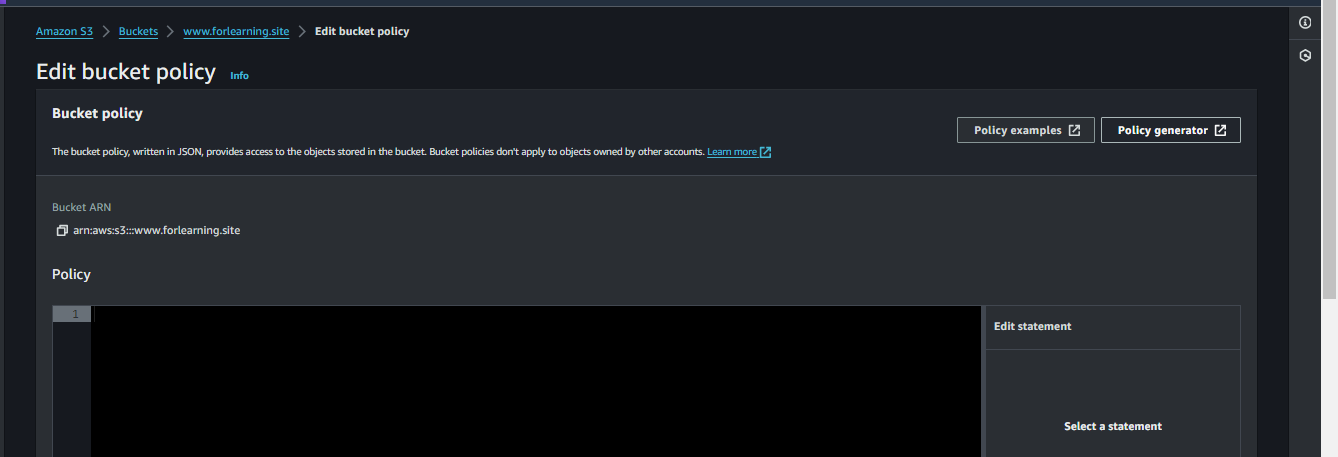
* + Choose a region.
  + Uncheck **Block all public access** (to allow public access to your website).
  + Acknowledge that the bucket will be public, then click **Create bucket**.

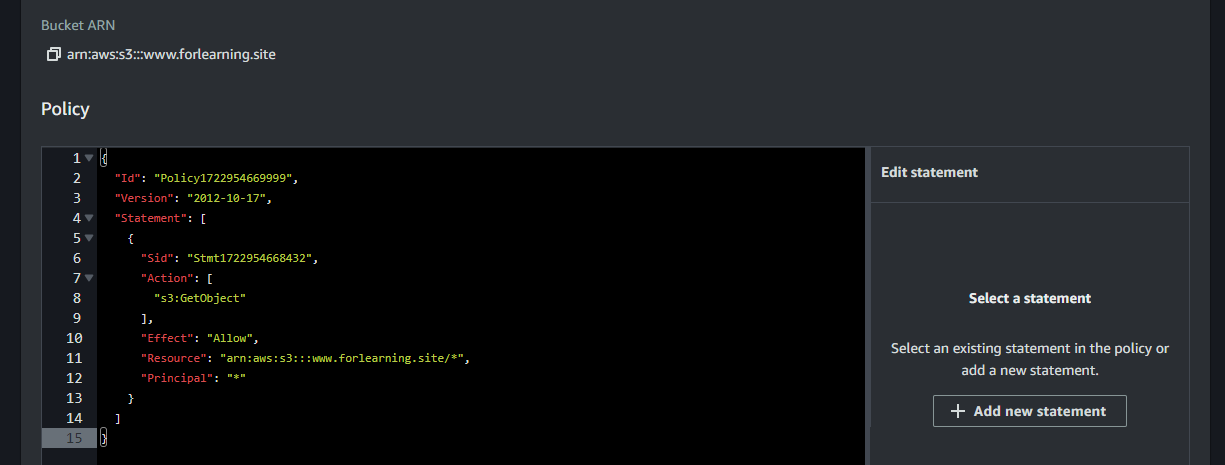
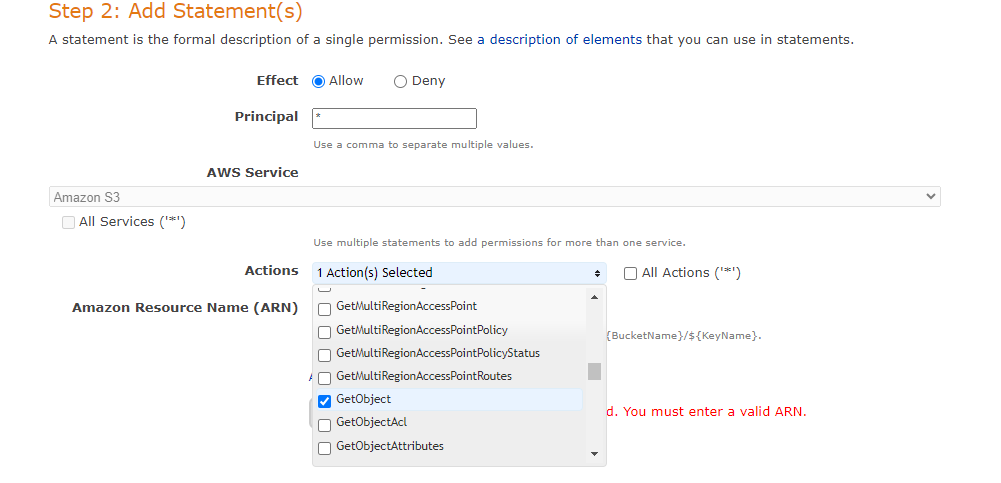


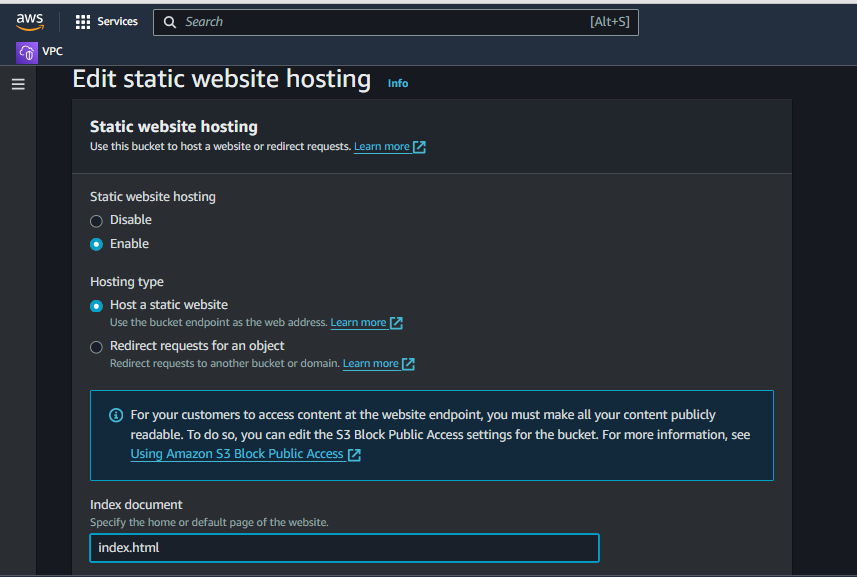
1. **Upload Website Files**:
   * Click on the newly created bucket.
   * Click **Upload** and add your website files (HTML, CSS, JavaScript, images, etc.)

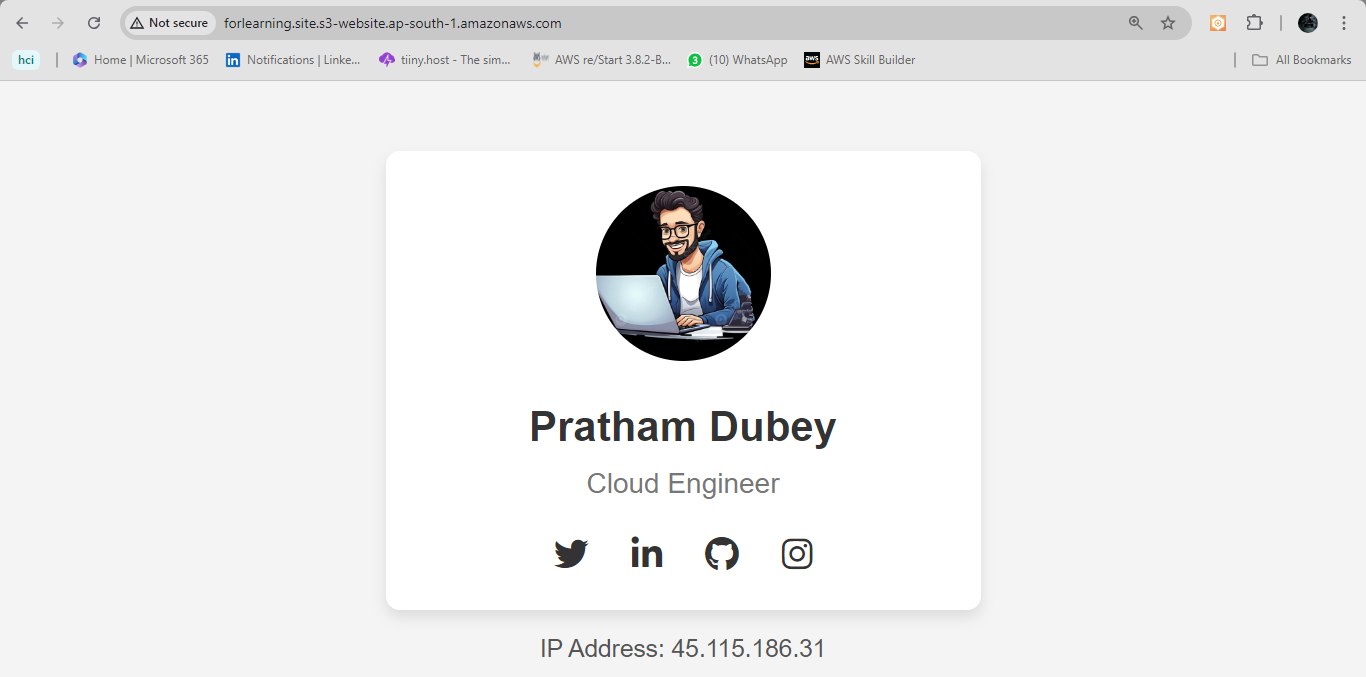
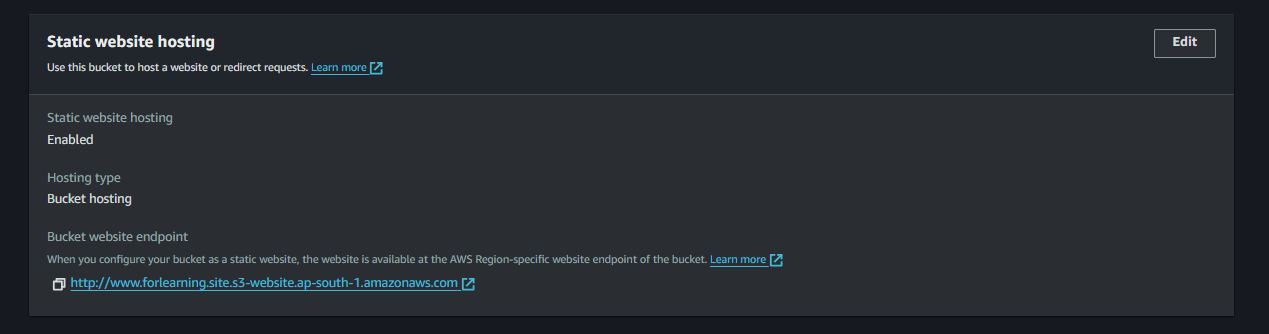


1. **Enable Static Website Hosting**:
   * Go to the **Properties** tab of the bucket.
   * Scroll down to **Static website hosting**.
   * Select **Enable** and choose **Host a static website**.
   * Specify the **index document** (e.g., index.html).
   * Note the **Bucket website endpoint**, as you'll use it later.



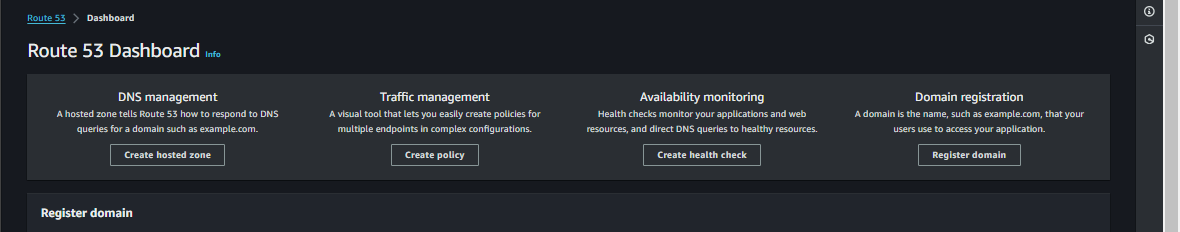




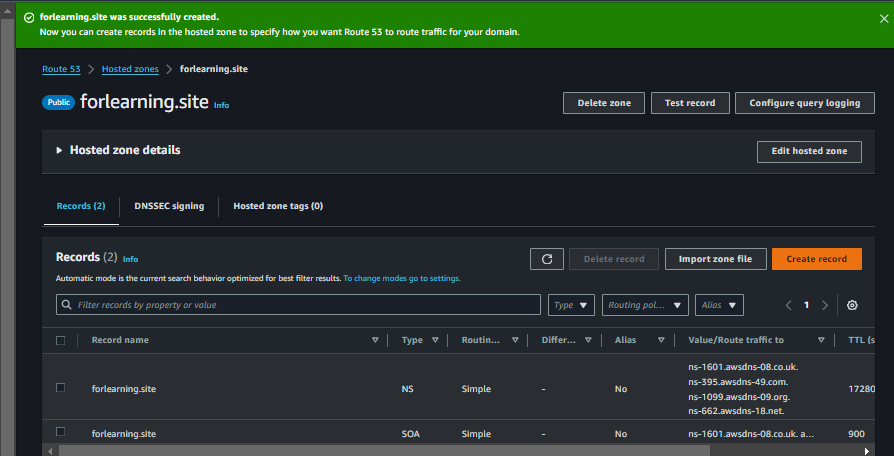
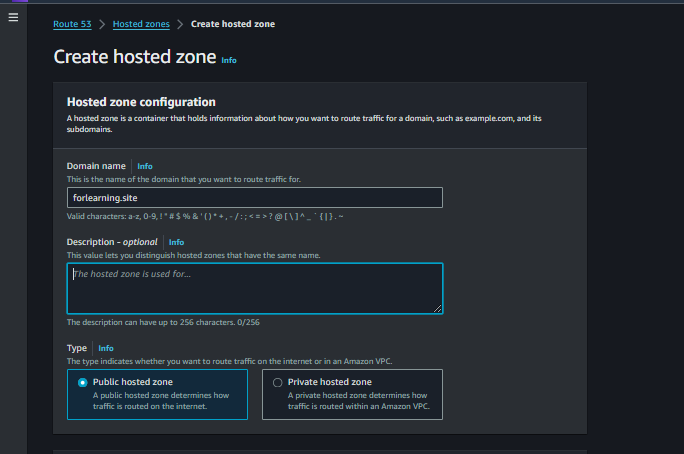


#### **2. Configure Amazon Route 53**

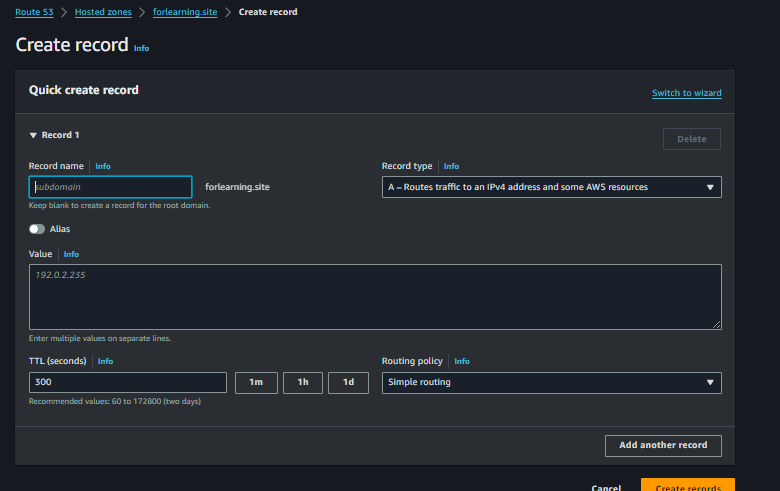
1. **Access Route 53**:
   * Go to the [Route 53 Console](https://console.aws.amazon.com/route53/).



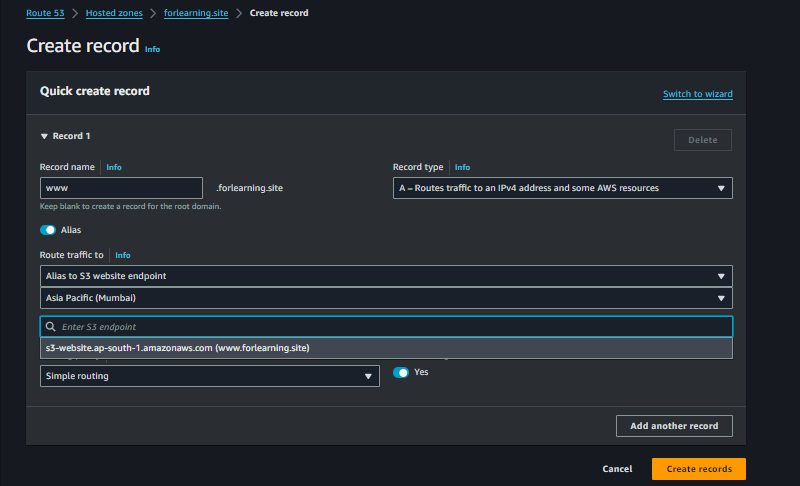
1. **Create a Hosted Zone**:
   * Click on **Create hosted zone**.
   * Enter your domain name (e.g., example.com).
   * Select **Public hosted zone**.
   * Click **Create hosted zone**.



1. **Create Record Sets**:
   * Click on the hosted zone you just created

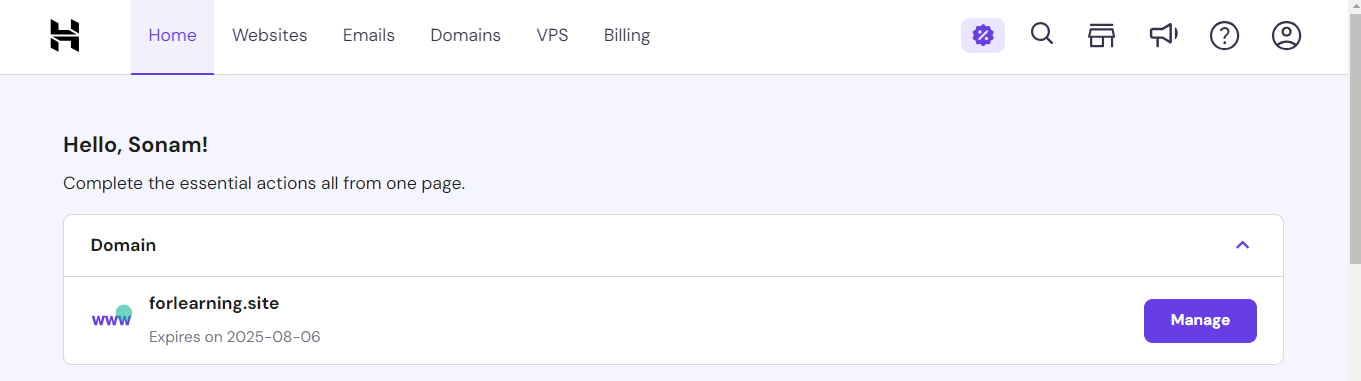
.

* + Click **Create record** and choose the **A record** type for www.example.com.
  + Select **Alias** and then **Alias to S3 website endpoint**.
  + Select the S3 bucket endpoint from the dropdown list.
  + Click **Create records**.

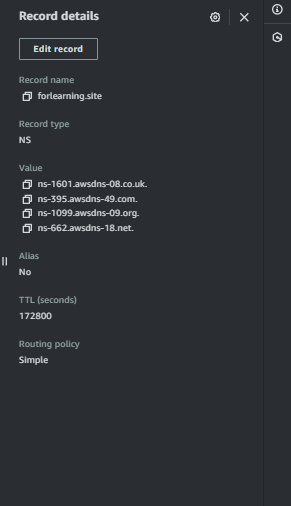


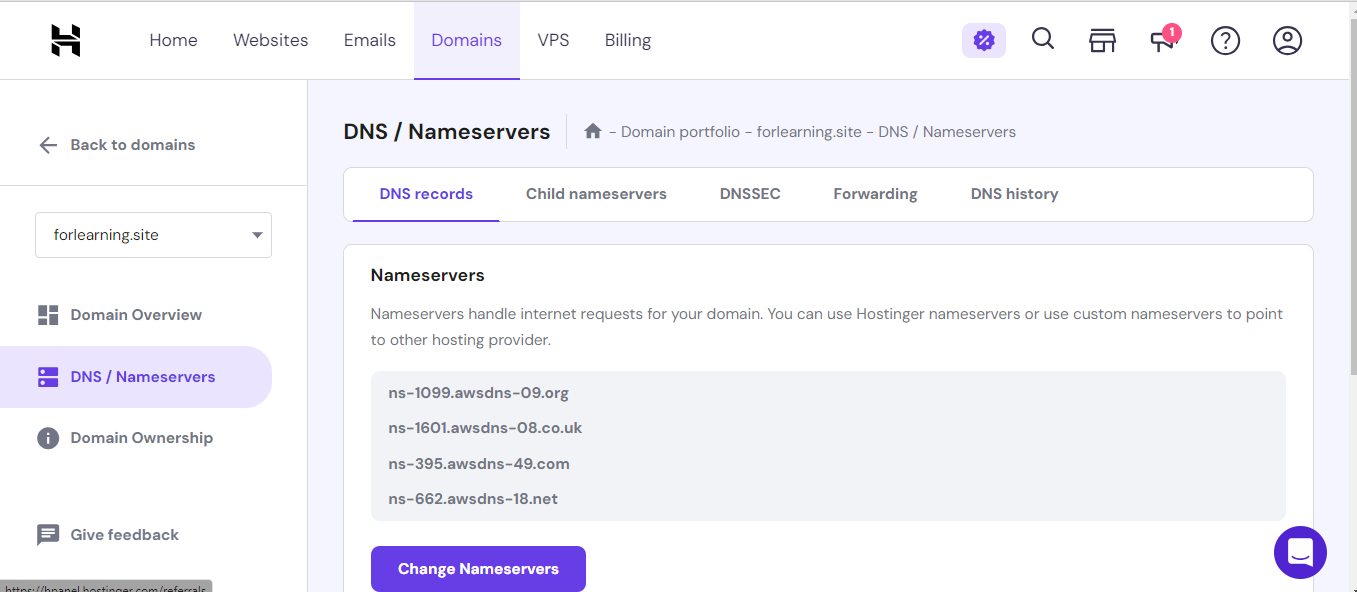
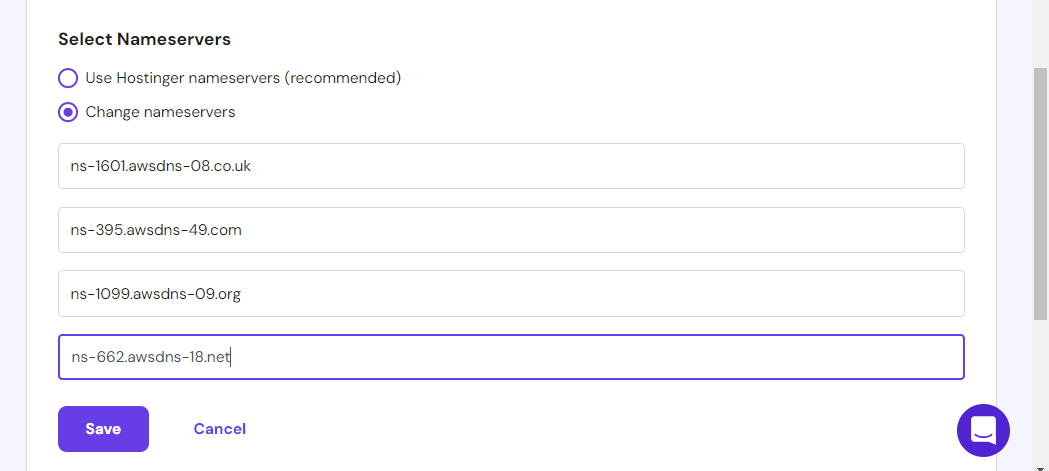
#### **3. Configure Hostinger Domain**

1. **Access Hostinger Dashboard**:
   * Log in to your Hostinger account and go to **Domains**.

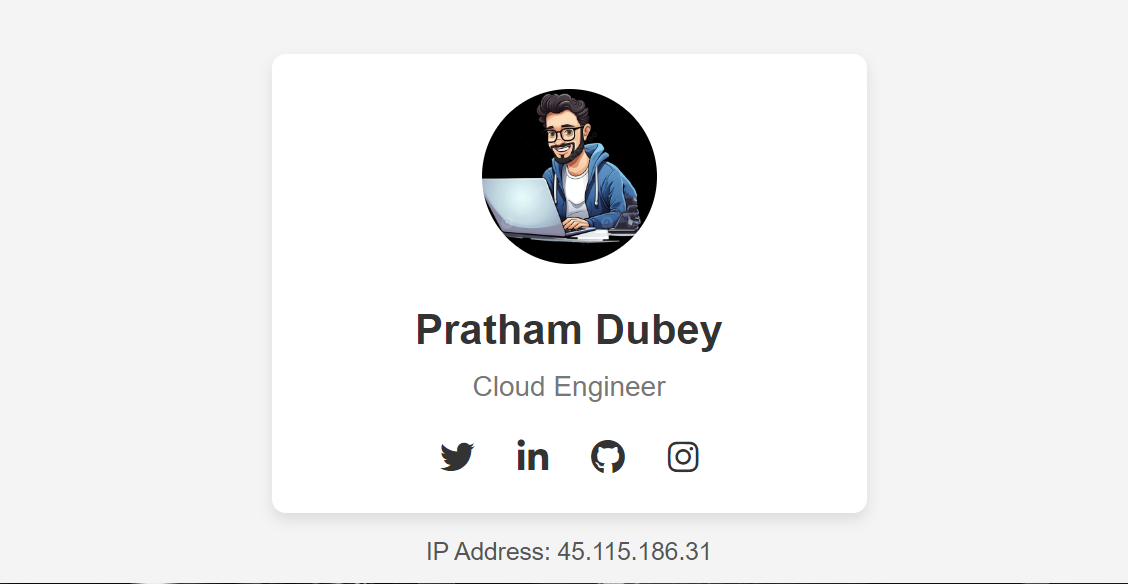


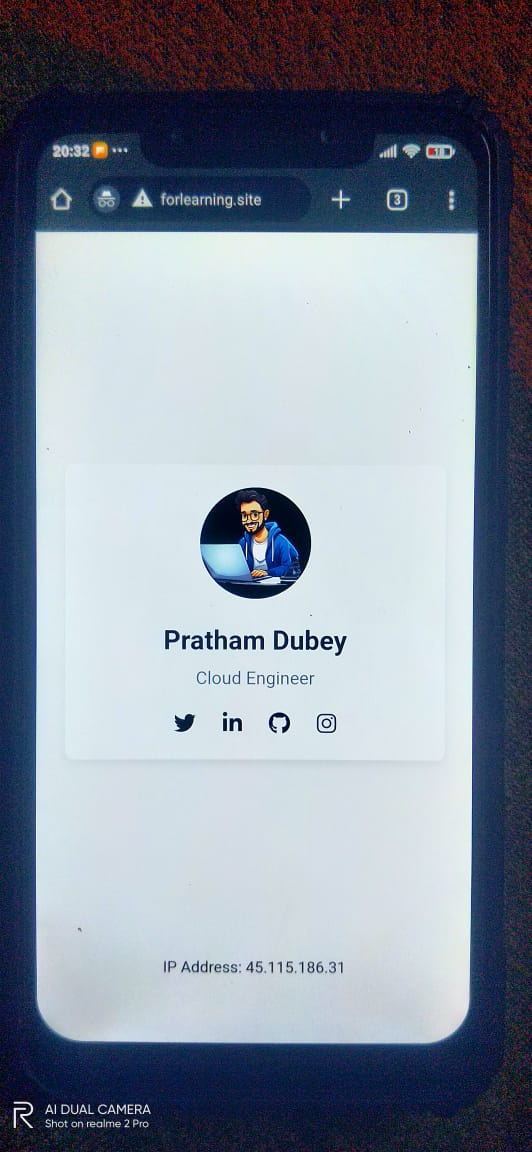
1. **Update Nameservers**:
   * Select your domain and navigate to **DNS/Nameservers** settings.
   * Update the nameservers to point to the Route 53 nameservers provided in your hosted zone:

****

****

1. **Verify DNS Propagation**:
   * Use a DNS propagation checker to ensure that the DNS changes have propagated.





### **Conclusion**

By following these steps, you've successfully set up a static website using Amazon S3 for hosting and Amazon Route 53 for DNS management with a domain name registered with Hostinger. Your website is now accessible through your domain name, and Route 53 efficiently handles DNS queries, providing a scalable and reliable solution for your static website hosting needs.

**Code for this**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Portfolio Card</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

<style>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

font-family: Arial, sans-serif;

background-color: #f4f4f4;

}

.portfolio-card {

background-color: #fff;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

overflow: hidden;

width: 300px;

text-align: center;

padding: 20px;

position: relative;

}

.profile-img img {

width: 100px;

height: 100px;

border-radius: 50%;

margin-bottom: 20px;

}

.profile-info h2 {

margin: 0;

font-size: 24px;

color: #333;

}

.profile-info p {

color: #777;

margin: 10px 0;

}

.social-media {

margin-top: 20px;

}

.social-media a {

color: #333;

font-size: 20px;

margin: 0 10px;

text-decoration: none;

transition: color 0.3s;

}

.social-media a:hover {

color: #0073b1; /\* Change this color to match the brand color of the social media icon \*/

}

.ip-address {

position: absolute;

bottom: 10px;

left: 50%;

transform: translateX(-50%);

font-size: 14px;

color: #555;

}

</style>

</head>

<body>

<div class="ip-address">

website ip is $(hostname)

</div>

<div class="portfolio-card">

<div class="profile-img">

<img src="https://s3.ap-south-1.amazonaws.com/www.forlearning.site/dp.jpeg" alt="Profile Image">

</div>

<div class="profile-info">

<h2>Pratham Dubey</h2>

<p>Cloud Engineer</p>

<div class="social-media">

<a href="https://twitter.com" target="\_blank"><i class="fab fa-twitter"></i></a>

<a href="https://linkedin.com" target="\_blank"><i class="fab fa-linkedin-in"></i></a>

<a href="https://github.com" target="\_blank"><i class="fab fa-github"></i></a>

<a href="https://instagram.com" target="\_blank"><i class="fab fa-instagram"></i></a>

</div>

</div>

</div>

</body>

</html>

### **(2)Objective**

Set up a static website hosted on two EC2 instances with an Application Load Balancer for traffic management and high availability. Use Route 53 to configure DNS routing with a Hostinger domain.

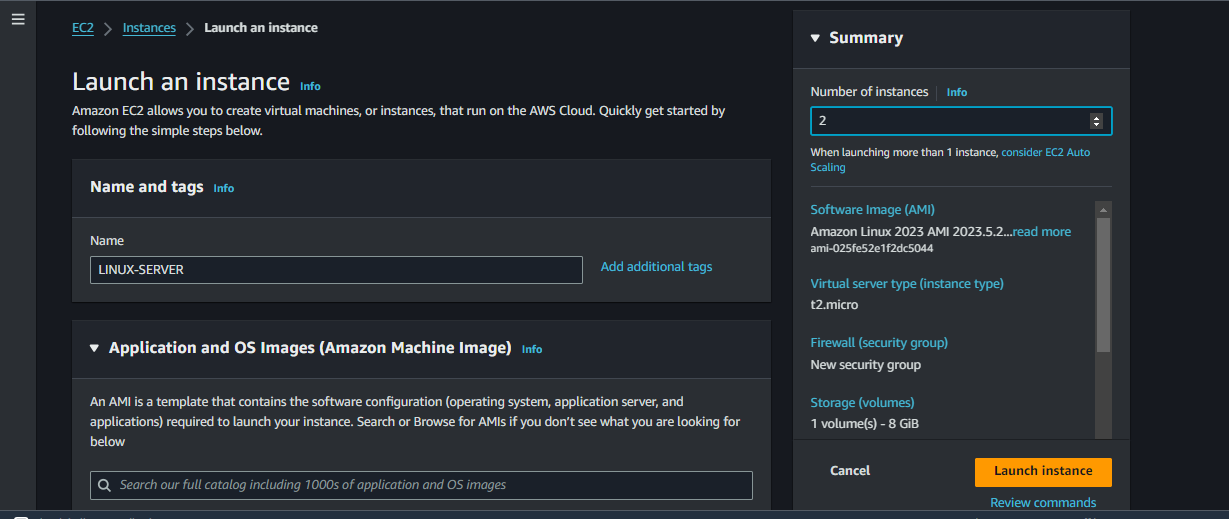
### **Architecture Flow**

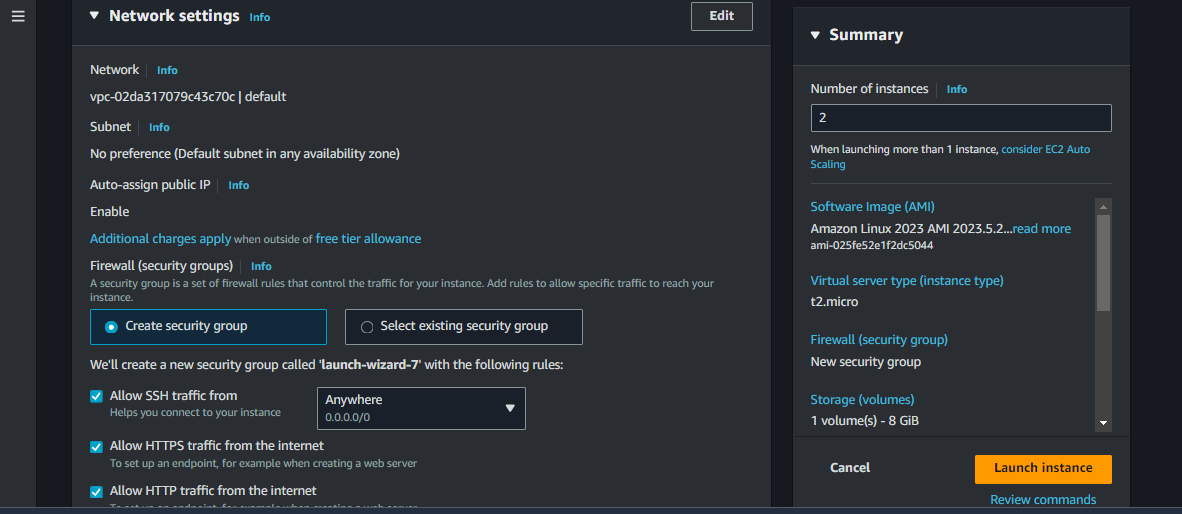
1. **Amazon EC2 Instances**: Launch two EC2 instances to host your static website files.
2. **Application Load Balancer (ALB)**: Distribute incoming traffic across the EC2 instances for high availability and fault tolerance.
3. **Route 53**: Manage DNS records for your domain and route traffic to the ALB.
4. **Domain Configuration**: Point your Hostinger domain to Route 53.

### **Steps**

#### **1. Launch EC2 Instances**

1. **Access EC2 Console**:
   * Go to the [Amazon EC2 Console](https://console.aws.amazon.com/ec2/).
2. **Launch EC2 Instances**:
   * Click on **Launch Instance**.
   * Choose an Amazon Machine Image (AMI), such as Amazon Linux 2.
   * Select an instance type (e.g., t2.micro for free tier).
   * Configure the instance details:
     + Ensure both instances are in the same VPC and region.
     + Assign a security group that allows HTTP (port 80) access.
   * Launch two instances.





1. **Install Web Server and Upload Files**:
   * Connect to each EC2 instance using SSH.

Install a web server (e.g., Apache) on each instance:

#!/bin/bash

sudo su

yum install httpd -y

systemctl start httpd

systemctl enable httpd

echo "<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Portfolio Card</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

<style>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

font-family: Arial, sans-serif;

background-color: #f4f4f4;

}

.portfolio-card {

background-color: #fff;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

overflow: hidden;

width: 300px;

text-align: center;

padding: 20px;

position: relative;

}

.profile-img img {

width: 100px;

height: 100px;

border-radius: 50%;

margin-bottom: 20px;

}

.profile-info h2 {

margin: 0;

font-size: 24px;

color: #333;

}

.profile-info p {

color: #777;

margin: 10px 0;

}

.social-media {

margin-top: 20px;

}

.social-media a {

color: #333;

font-size: 20px;

margin: 0 10px;

text-decoration: none;

transition: color 0.3s;

}

.social-media a:hover {

color: #0073b1; /\* Change this color to match the brand color of the social media icon \*/

}

.ip-address {

position: absolute;

bottom: 10px;

left: 50%;

transform: translateX(-50%);

font-size: 14px;

color: #555;

}

</style>

</head>

<body>

<div class="ip-address">

website ip is $(hostname)

</div>

<div class="portfolio-card">

<div class="profile-img">

<img src="https://s3.ap-south-1.amazonaws.com/www.forlearning.site/dp.jpeg" alt="Profile Image">

</div>

<div class="profile-info">

<h2>Pratham Dubey</h2>

<p>Cloud Engineer</p>

<div class="social-media">

<a href="https://twitter.com" target="\_blank"><i class="fab fa-twitter"></i></a>

<a href="https://linkedin.com" target="\_blank"><i class="fab fa-linkedin-in"></i></a>

<a href="https://github.com" target="\_blank"><i class="fab fa-github"></i></a>

<a href="https://instagram.com" target="\_blank"><i class="fab fa-instagram"></i></a>

</div>

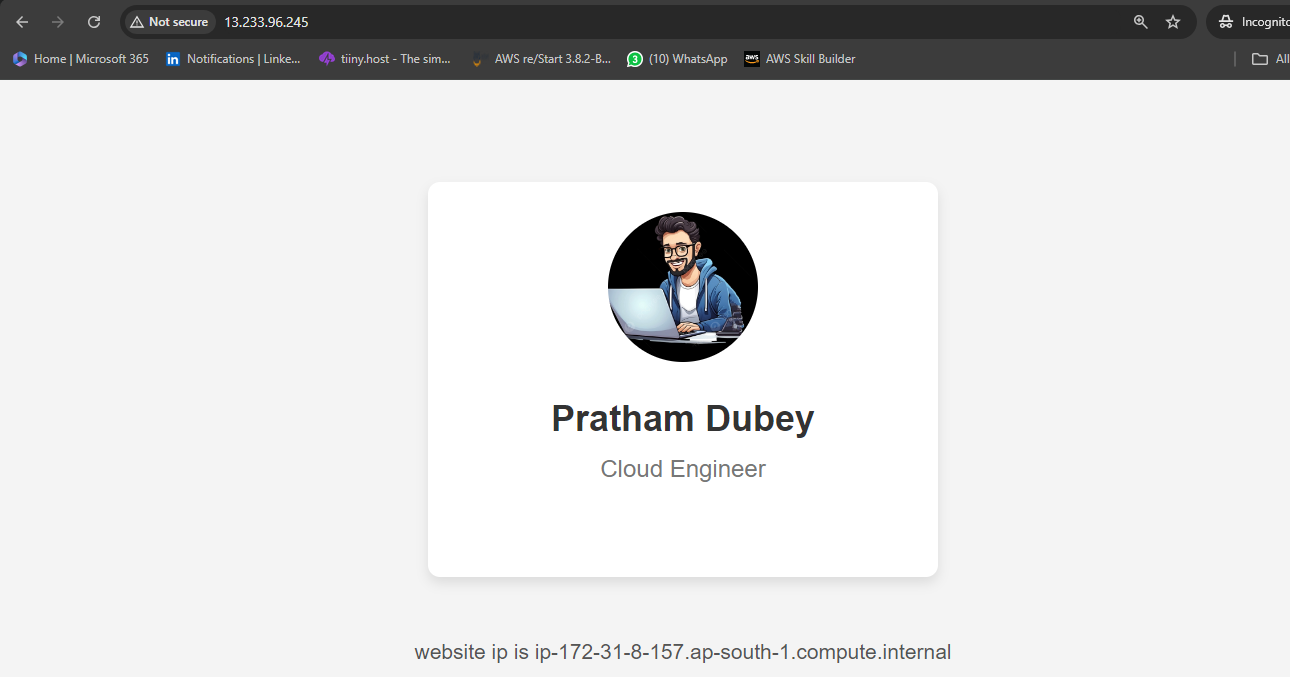
</div>

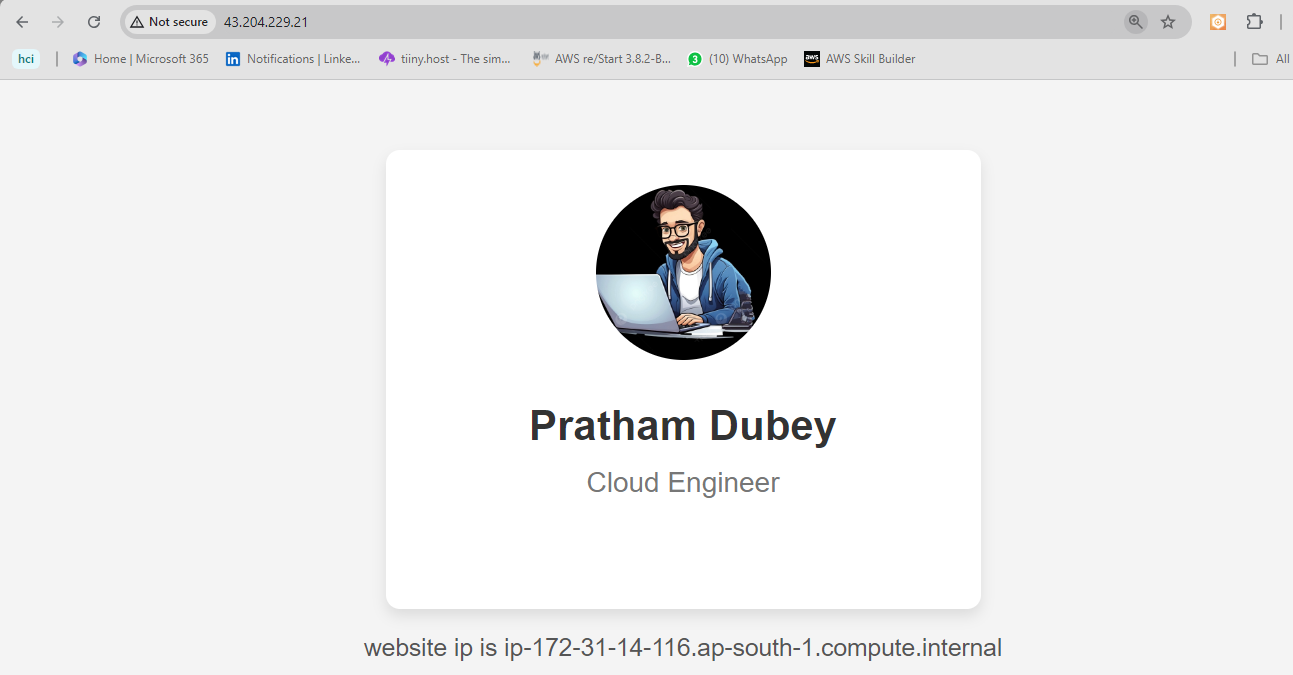
</div>

</body>

</html>

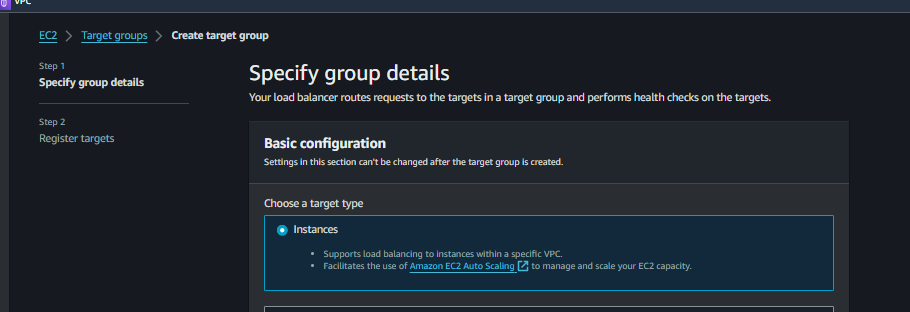
" > /var/www/html/index.html



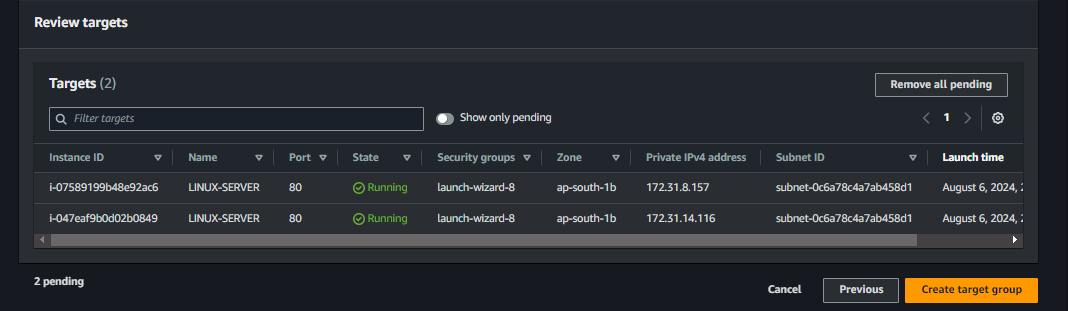
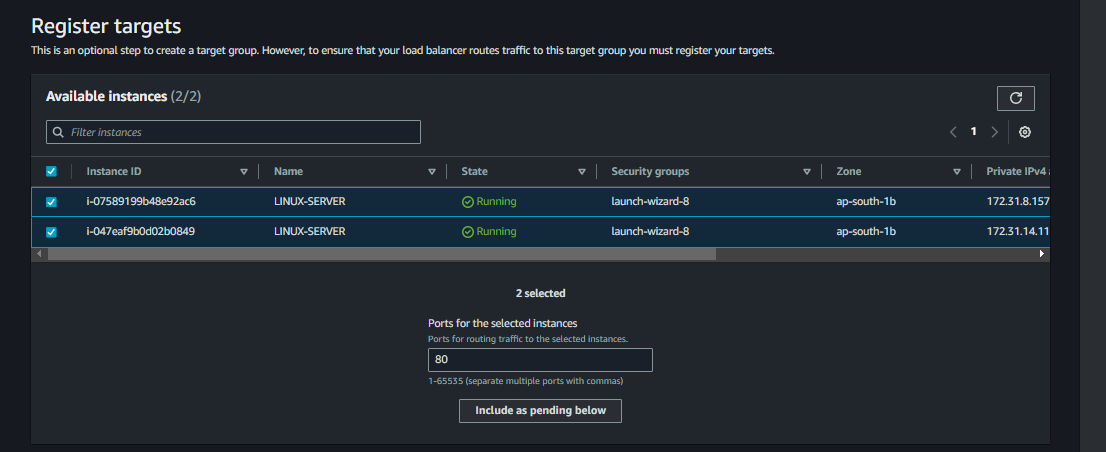


**2. Set Up an Application Load Balancer**

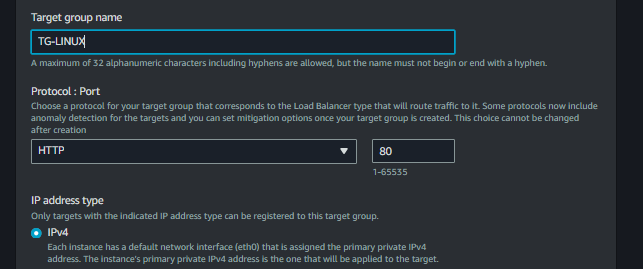
1. **Create a Target Group**:
   * Go to the [EC2 Console](https://console.aws.amazon.com/ec2/).
   * Under **Load Balancing**, select **Target Groups**.
   * Click **Create target group**.



* + Choose **Instances** as the target type.

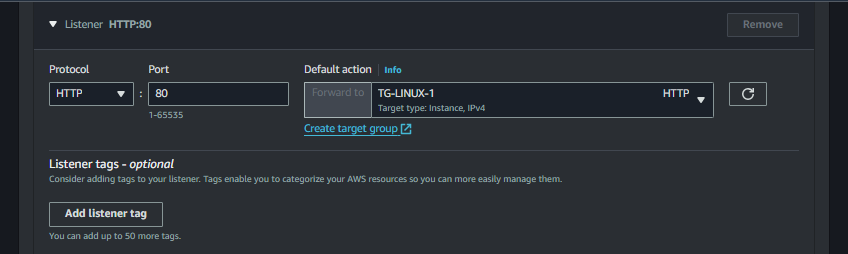


* + Specify a name and protocol (HTTP).

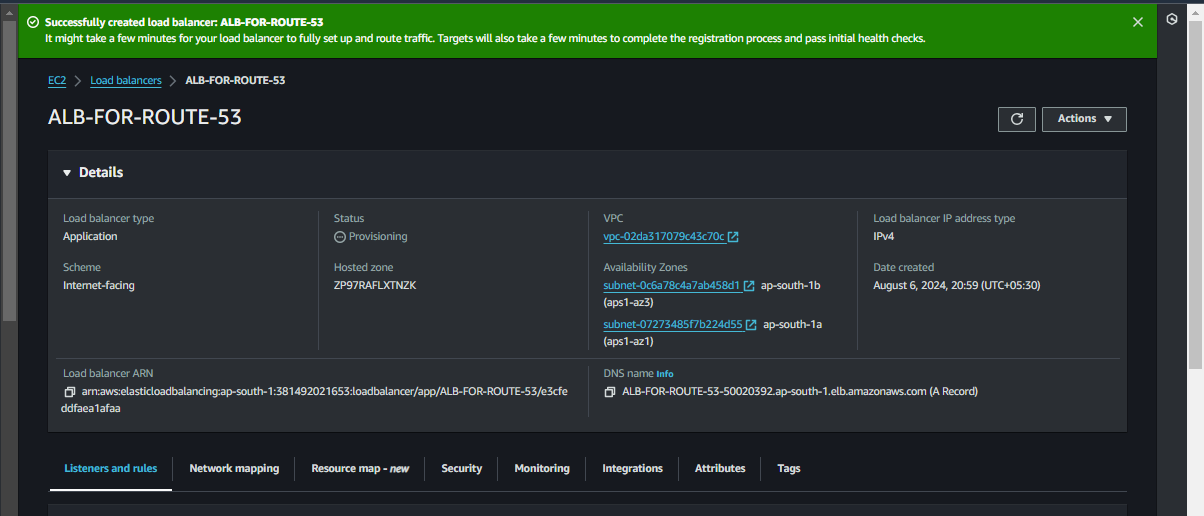
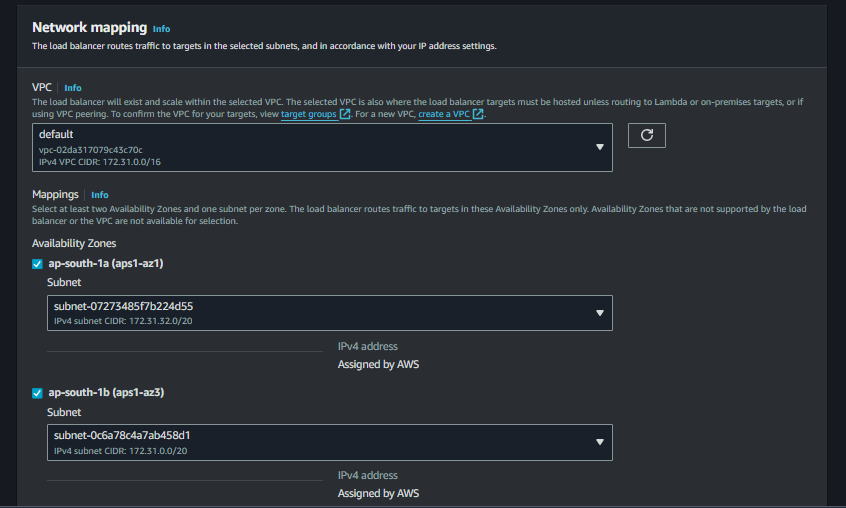
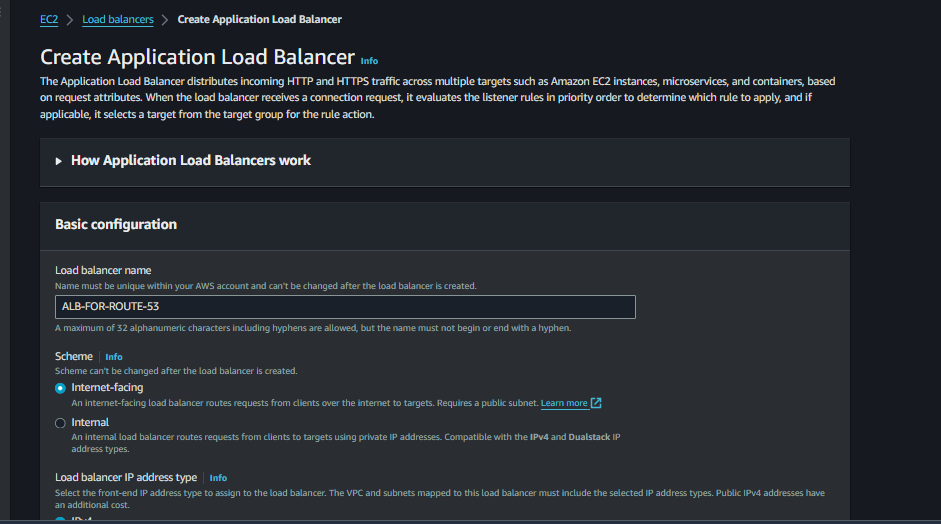


* + Choose the VPC where your instances are located.
  + Click **Next** and register both EC2 instances.

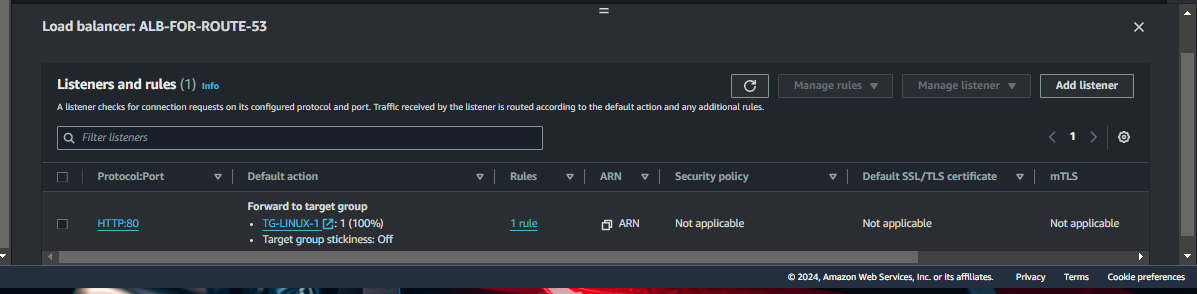
1. **Create an Application Load Balancer**:
   * Go to **Load Balancers** under the EC2 Console.
   * Click **Create Load Balancer** and select **Application Load Balancer**.
   * Specify a name, scheme (internet-facing), and listener (HTTP).
   * Choose subnets in different availability zones for high availability.
   * Configure a security group to allow HTTP (port 80) traffic.
   * Select the target group you created earlier.



* + Click **Create Load Balancer**.

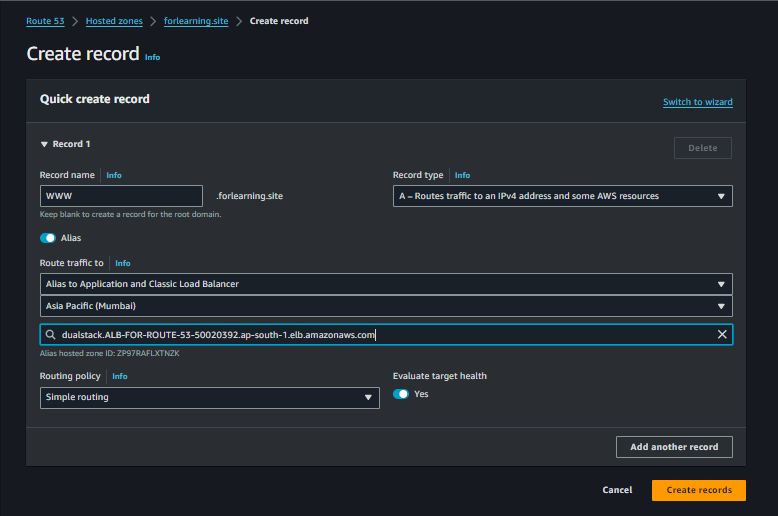


1. **Configure Listeners and Routing**:
   * After creating the ALB, go to the **Listeners** tab.
   * Ensure that the listener forwards requests to the target group.



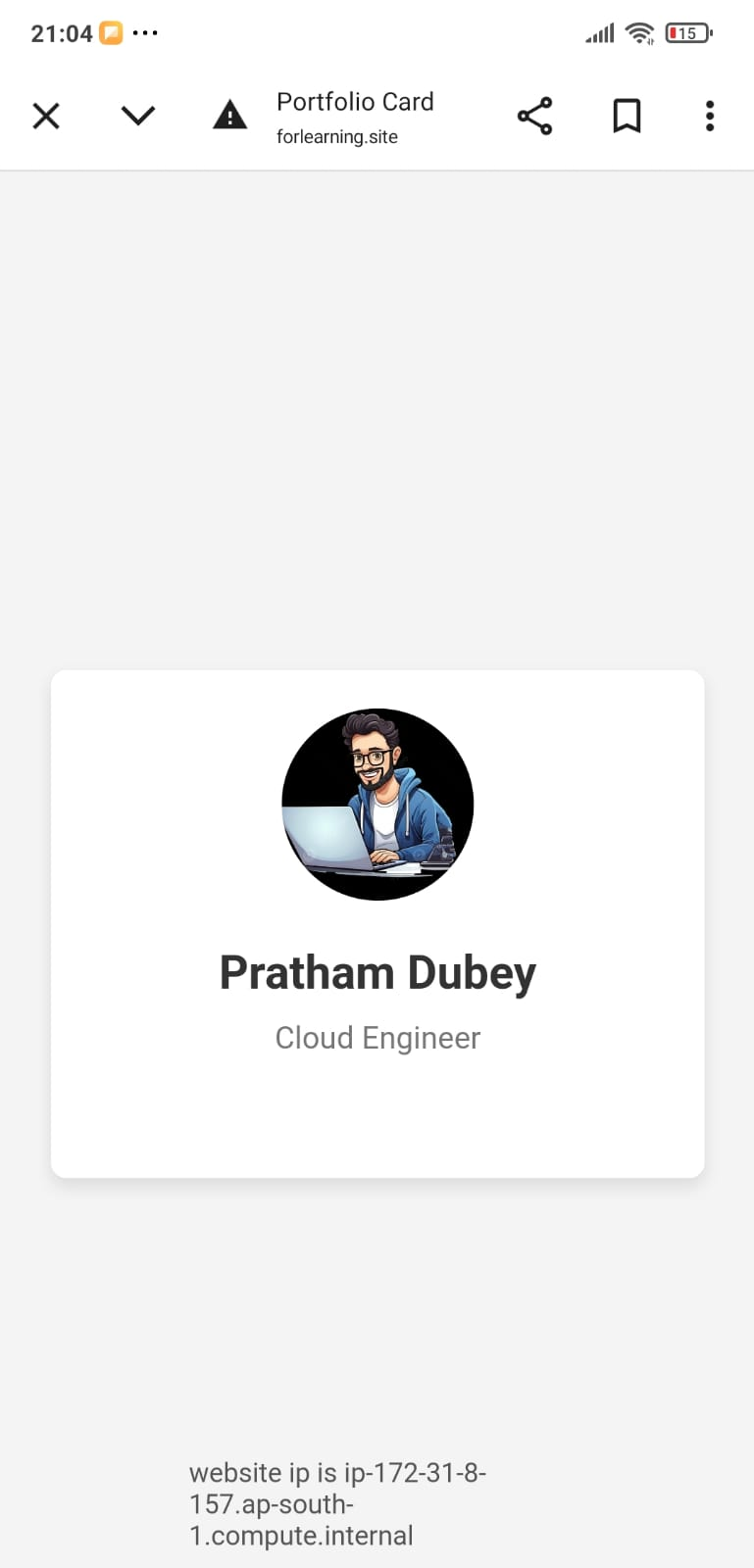
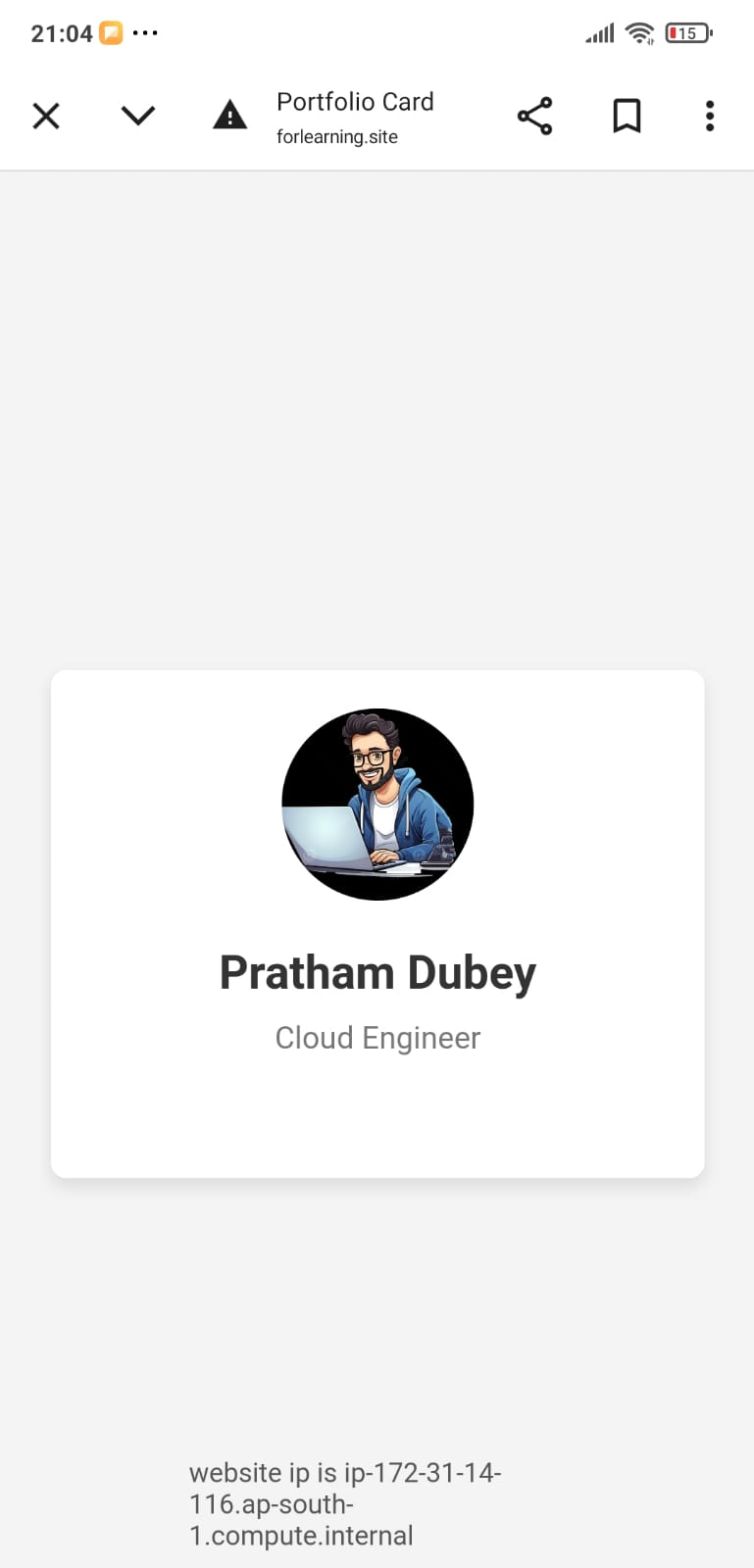
#### **3. Configure Amazon Route 53**

1. **Access Route 53**:
   * Go to the [Route 53 Console](https://console.aws.amazon.com/route53/).
2. **Create a Hosted Zone**:
   * Click on **Create hosted zone**.
   * Enter your domain name (e.g., example.com).
   * Select **Public hosted zone**.
   * Click **Create hosted zone**.
3. **Create Record Sets**:
   * Click on the hosted zone you just created.
   * Click **Create record** and choose the **A record** type for www.example.com.
   * Select **Alias** and then **Alias to Application Load Balancer**.
   * Select the ALB DNS name from the dropdown list.
   * Click **Create records**.



#### **4. Configure Hostinger Domain**

1. **Access Hostinger Dashboard**:
   * Log in to your Hostinger account and go to **Domains**.
2. **Update Nameservers**:
   * Select your domain and navigate to **DNS/Nameservers** settings.
   * Update the nameservers to point to the Route 53 nameservers provided in your hosted zone.
3. **Verify DNS Propagation**:
   * Use a DNS propagation checker to ensure that the DNS changes have propagated.



### **Conclusion**

By following these steps, you've successfully set up a static website using two EC2 instances behind an Application Load Balancer. Amazon Route 53 manages DNS queries, and your website is now accessible via your domain name, providing high availability and fault tolerance. This architecture ensures that traffic is distributed evenly across the instances, enhancing performance and reliability.

## **(3)Objective**

The primary goal of this architecture is to design a scalable and high-performance content delivery system utilizing Amazon Web Services (AWS). The system aims to efficiently serve static assets, such as images and videos, to users globally with minimal latency and high reliability.

## **Architecture Flow**

### **1. Overview**

The architecture leverages three key AWS services: Amazon S3 for storage, Amazon CloudFront for content delivery, and Amazon Route 53 for DNS management.

* **Amazon S3:** Stores static content such as images, videos, and documents.
* **Amazon CloudFront:** Distributes and caches content globally, reducing latency and improving load times for end users.
* **Amazon Route 53:** Manages DNS routing and directs user requests to the appropriate CloudFront distribution.

**Diagram:**

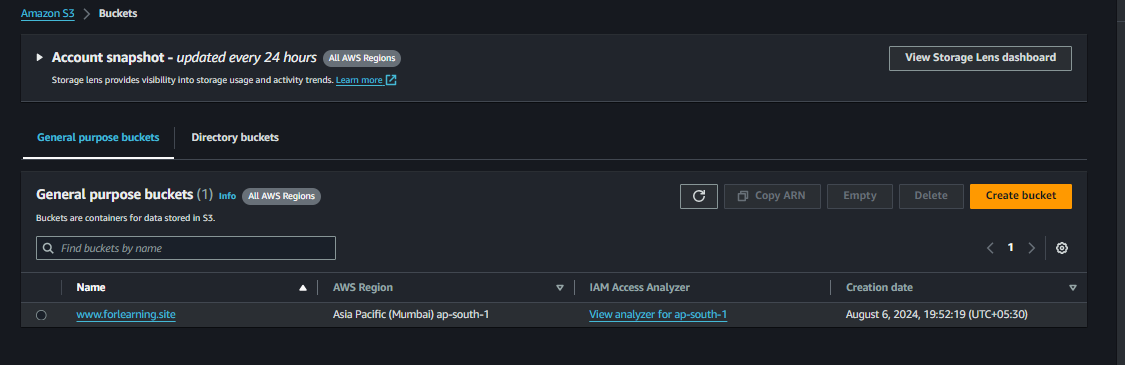
### **2. Components**

* **Amazon S3:**
  + *Role:* Acts as the origin for static content storage.
  + *Configuration:* A bucket is created with public access permissions to enable static website hosting.
* **Amazon CloudFront:**
  + *Role:* Functions as a Content Delivery Network (CDN) to cache and serve content from edge locations around the world.
  + *Configuration:* A CloudFront distribution is set up with the S3 bucket as the origin. HTTPS is enabled, and custom domain names can be configured.
* **Amazon Route 53:**
  + *Role:* Handles DNS management to ensure that user requests are routed to the CloudFront distribution.
  + *Configuration:* A hosted zone is created, and DNS records are set up to point to the CloudFront distribution.

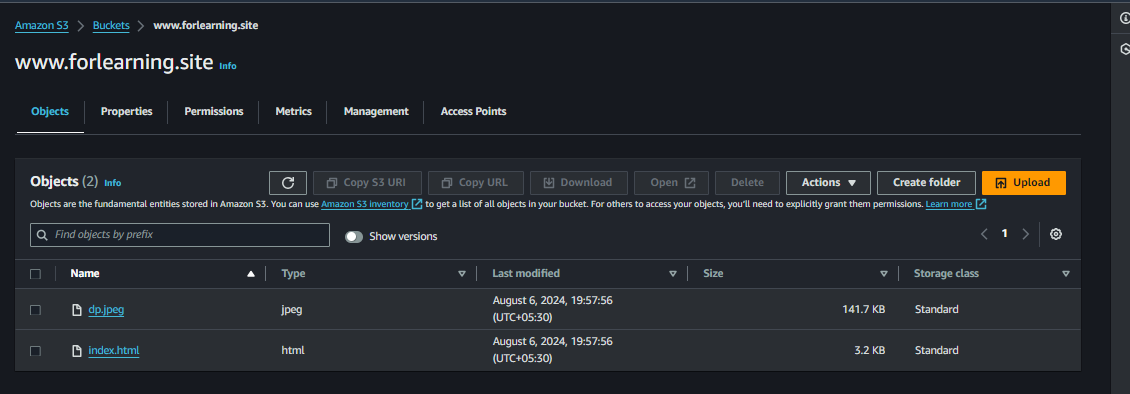
## **Steps**

### **1. Set Up S3 Bucket**

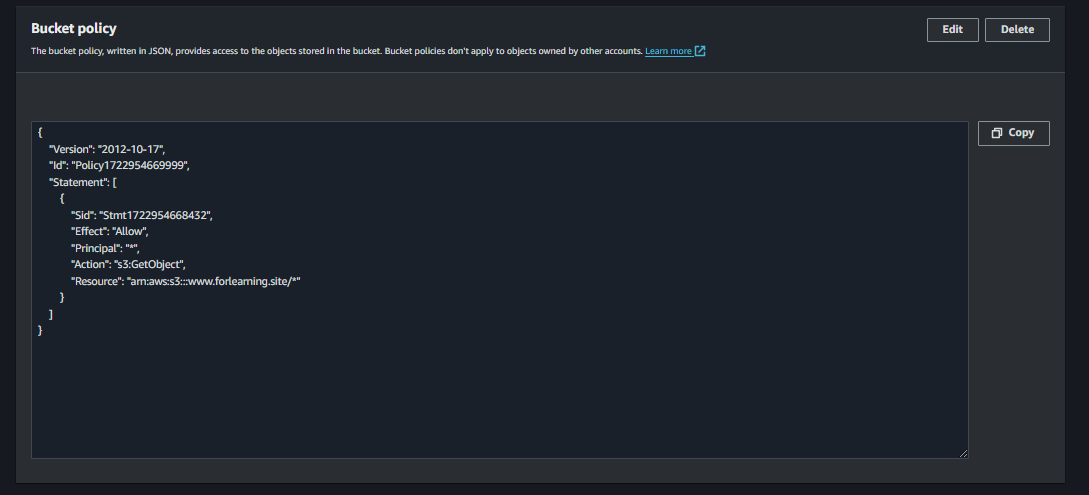
1. **Create the Bucket:**
   * Navigate to the Amazon S3 console.
   * Click “Create bucket.”
   * Enter a unique bucket name and select a region.
   * Click “Create bucket” after configuring additional settings.



1. **Upload Content:**
   * Go to the newly created bucket.
   * Click “Upload” to add files to the bucket.

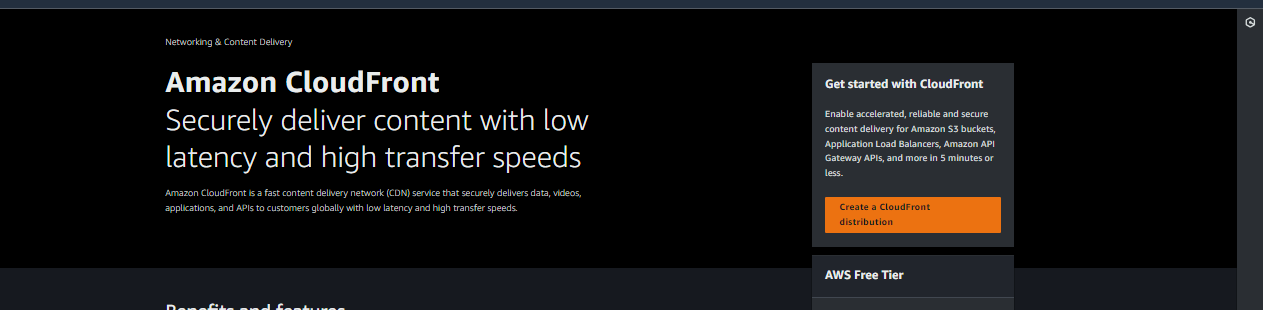


1. **Set Permissions:**
   * Access the bucket’s properties.
   * Go to the “Permissions” tab.
   * Configure the bucket policy to allow public access if required.

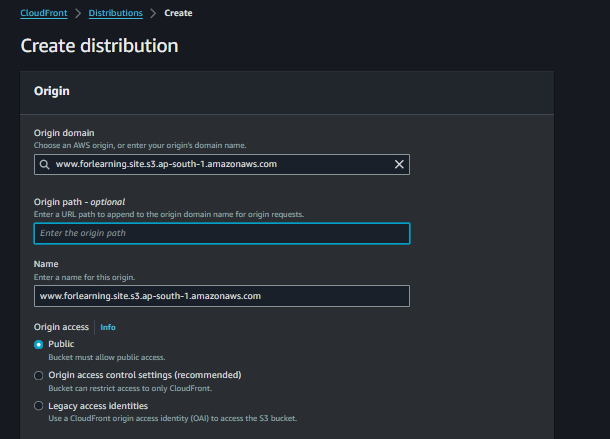


### **2. Configure CloudFront Distribution**

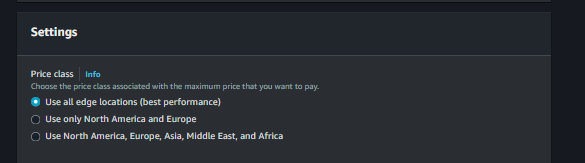
1. **Create Distribution:**
   * Open the CloudFront console.
   * Click “Create Distribution.”
   * Choose the “Web” delivery method.



1. **Configure Origin Settings:**
   * Set “Origin Domain Name” to the S3 bucket URL.
   * Configure additional origin settings as needed.

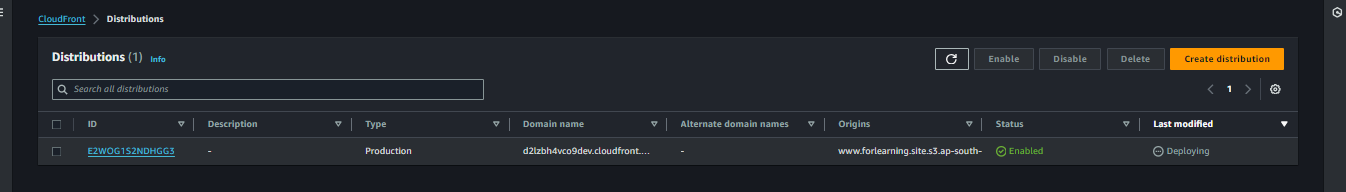
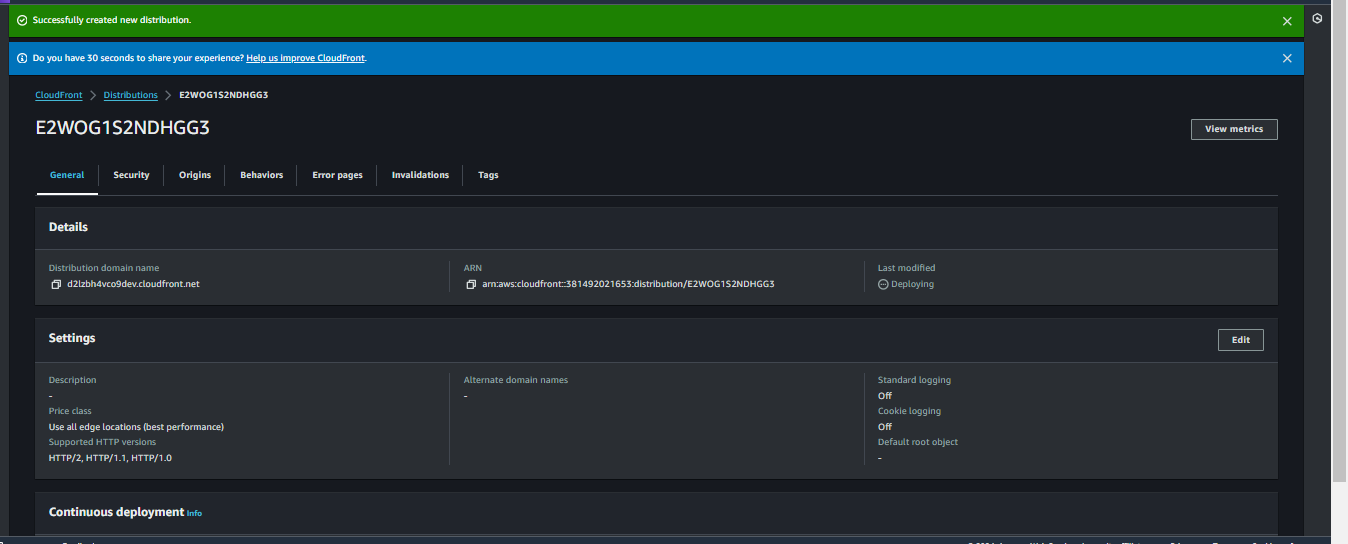


1. **Configure Cache Behavior:**
   * Set the “Viewer Protocol Policy” to “Redirect HTTP to HTTPS” or “HTTPS Only.”
   * Define allowed HTTP methods and caching policies.
2. **Configure Distribution Settings:**
   * Choose a “Price Class” based on expected traffic.



**Create Distribution:**

* + Click “Create Distribution” to finalize the setup.



### **3. Configure Route 53**

1. **Create Hosted Zone:**
   * Open the Route 53 console.
   * Click “Create Hosted Zone.”
   * Enter the domain name and select “Public Hosted Zone.”
2. **Create DNS Record Set:**
   * In the hosted zone, click “Create Record Set.”
   * Set the record type to “A – IPv4 address.”
   * Choose “Alias” and set the alias target to the CloudFront distribution domain name.
   * Configure TTL (Time to Live) as needed.
3. **Update DNS Settings:**
   * For domains registered outside Route 53, update the domain registrar’s DNS settings to use Route 53’s name servers.
   * If using Route 53 as the registrar, DNS settings are automatically updated.

### **4. Validate and Test**

1. **Check CloudFront Status:**
   * Ensure that the CloudFront distribution status is “Deployed” in the CloudFront console.
2. **Test Access:**
   * Access content through the CloudFront distribution URL or the configured custom domain.
   * Verify that content is being served correctly and that caching is functioning as expected.

## **Conclusion**

The implementation of this architecture successfully established a scalable and high-performance content delivery system. Amazon S3 provided reliable storage, CloudFront enhanced global content delivery with reduced latency, and Route 53 efficiently managed DNS routing. The system meets the objective of delivering static content quickly and reliably to users worldwide.