Deploying a Static Website on AWS S3

Bucket

[**1. Background and Context 3**](#_w2uq5hvy95kj)

[1.1. Introduction: 3](#_mvxe1i94pwin)

[1.2. Problem Statement: 4](#_13kl9o61lsru)

[1.3. Prerequisites: 4](#_msqsp4359ahi)

[**2. Static Website Configuration: 5**](#_kt48q6lqkyh5)

[a. Set up S3 Bucket on AWS: 5](#_hnvrezos0jb8)

[1. Login to AWS: 5](#_9quia6fxww5s)

[2. Create S3 Bucket: 6](#_lz0ds1d1sjh3)

[3. Disable “Block Public Access” 7](#_tv1d0mtsv74k)

[4. Enable Static Website Hosting: 7](#_mm430hyxrsz3)

[5. Add Bucket Policy: 8](#_vox4b4lxivyc)

[b. Upload Website Files: 9](#_6riqdatjul18)

[1. Upload Website Files: 9](#_egpr2hfp9xiq)

[2. Check Website Accessibility on S3 endpoint. 10](#_oidlb7svbfzw)

[c. Domain Configuration: 10](#_enw3l827n8qw)

[1. Purchase Domain: 10](#_yxctmd642ygr)

[2. Create Hosted Zone in Route 53: 10](#_hspkemfk5ric)

[3. Update DNS Records in your domain provider DNS settings: 11](#_rrax5pdv43xd)

[d. Security and Performance: 13](#_u0gx3z7bkg2a)

[1. Request a SSL certificate. 13](#_pb4yxmb78gbx)

[2. Create a CloudFront distribution. 15](#_3k1fgsckav4l)

[3. Create A Record of CloudFront distribution in Route53. 18](#_yonelpy1aswx)

[**3. Troubleshooting: 18**](#_68paga6ewf8)

[1. Website not accessible after configuring S3 bucket for website hosting: 19](#_rjo7e3xb67a2)

[2. DNS records not propagating or taking a long time to propagate: 19](#_ocu2gwel5jhb)

[3. Website not accessible via the domain/subdomain: 19](#_c7xo3f3twhfl)

[4. SSL certificate errors or HTTPS not working: 19](#_6384mj3bccdf)

[5. CloudFront distribution issues: 20](#_jrvv7092z0ls)

[**4. Conclusion: 20**](#_wy6uvsb9f1v3)

[**5. References: 21**](#_3jnkqljk475p)

# 

# Background and Context

In today's digital world, having a strong online presence is essential for individuals and businesses. A static website is a common and straightforward way to showcase information, promote products or services, or share personal portfolios. The cloud distribution has made it easy to host and deploy static websites.

Amazon Web Services (AWS) is one of the cloud providers that offers services to deploy and host a static websites. AWS service S3 Bucket (Simple Storage Service) provides an easy, scalable and reliable storage solution. While service like CloudFront provides content delivery and caching capabilities. Route53 is another service of AWS that is used to enable mapping of domain names to the website hosted on S3.

## Introduction:

The objective is to deploy a static website on an AWS S3 bucket, utilizing various AWS services for configuration, domain mapping, security, and performance optimization. This document provides a step-by-step guide to accomplish the task, ensuring that the static website is accessible to users and meets best practices in terms of security and performance.

The deployment process involves several key steps. First, you will create an S3 bucket to host the static website and configure it for website hosting. Next, you will upload the website files to the S3 bucket and verify their accessibility. Then, you will configure a domain name or subdomain for the website and set up DNS records to map the domain/subdomain to the S3 bucket. To enhance security, you will enable HTTPS using either AWS Certificate Manager (ACM) or a third-party SSL certificate. Finally, you will implement caching and content delivery mechanisms, such as CloudFront or a CDN, to optimize the website's performance.

Throughout the documentation, each step will be explained in detail, including any assumptions or considerations made during the deployment process. So, let's get started and deploy your static website on AWS S3!

## Problem Statement:

The task at hand is to deploy a static website on an AWS S3 bucket while adhering to best practices for configuration, domain mapping, security, and performance optimization. The objective is to create a production-ready infrastructure that ensures the website is accessible to users, secure, and performs optimally.

The task includes creating an S3 bucket, configuring it for website hosting, uploading website files, and ensuring the appropriate permissions are set for public access. Additionally, mapping a domain name or subdomain to the S3 bucket and setting up DNS records can be complex tasks that require careful configuration. Security considerations involve enabling HTTPS using AWS Certificate Manager (ACM) or a third-party SSL certificate, while performance optimization entails implementing caching and content delivery mechanisms like CloudFront or a CDN.

## Prerequisites:

It is importat to ensure that you have following prerequisite in place before

following this documentation.

1. **AWS Account:** You must have a valid AWS account with appropriate permissions to create and manage AWS services.
2. **Domain Name:** If you intend to use a custom domain name for your static website, you should have already purchased or registered a domain name from a domain registrar like Namecheap, GoDaddy, or AWS Route 53.
3. **DNS Management:** You should have access to the DNS management settings of your domain registrar or DNS provider. This will allow you to configure DNS records necessary for mapping the domain or subdomain to the S3 bucket.
4. **Website Files:** Prepare the static website files including HTML, CSS, JavaScript, images, and any other assets required for your website. Ensure that the files are properly structured and organized.

By fulfilling these prerequisites, you will be well-prepared to proceed with the deployment of your static website on an AWS S3 bucket.

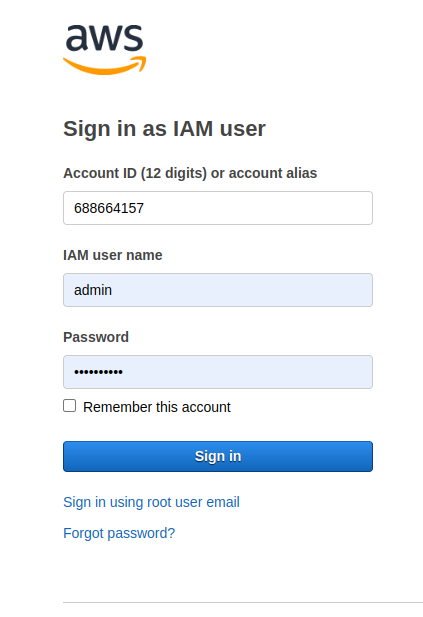
# **Static Website Configuration:**

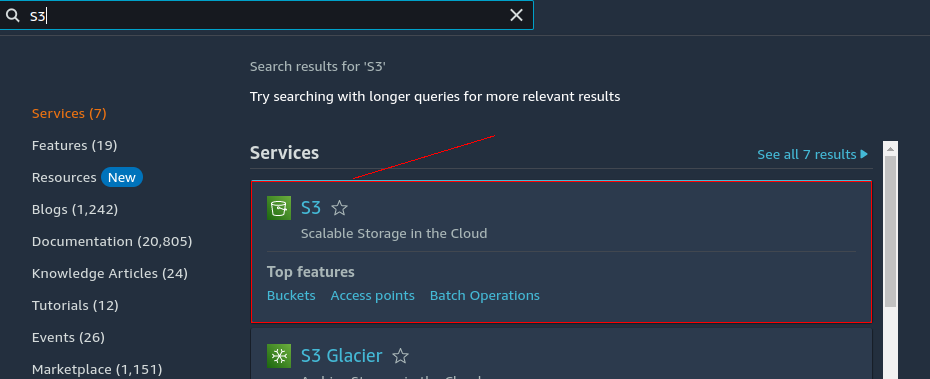
## Set up S3 Bucket on AWS:

Seting up AWS S3 bucket involves several steps to configure it for

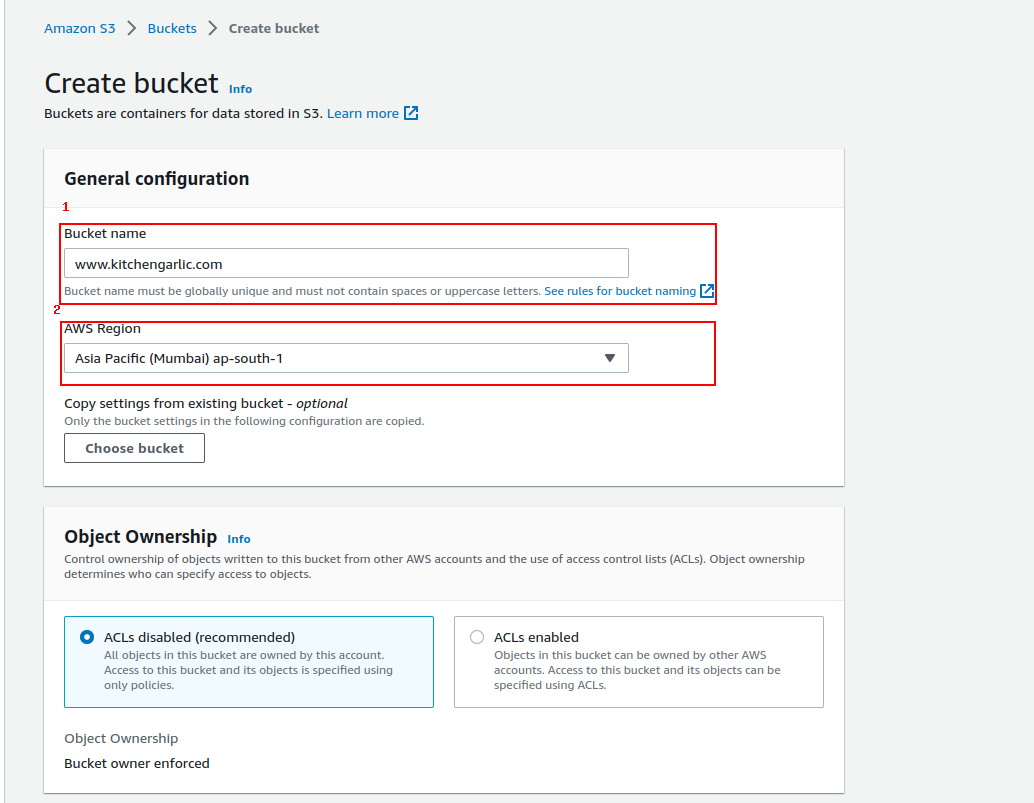
hosting a static website. Here is the step by step guide on how to create S3 bucket.

### Login to AWS:

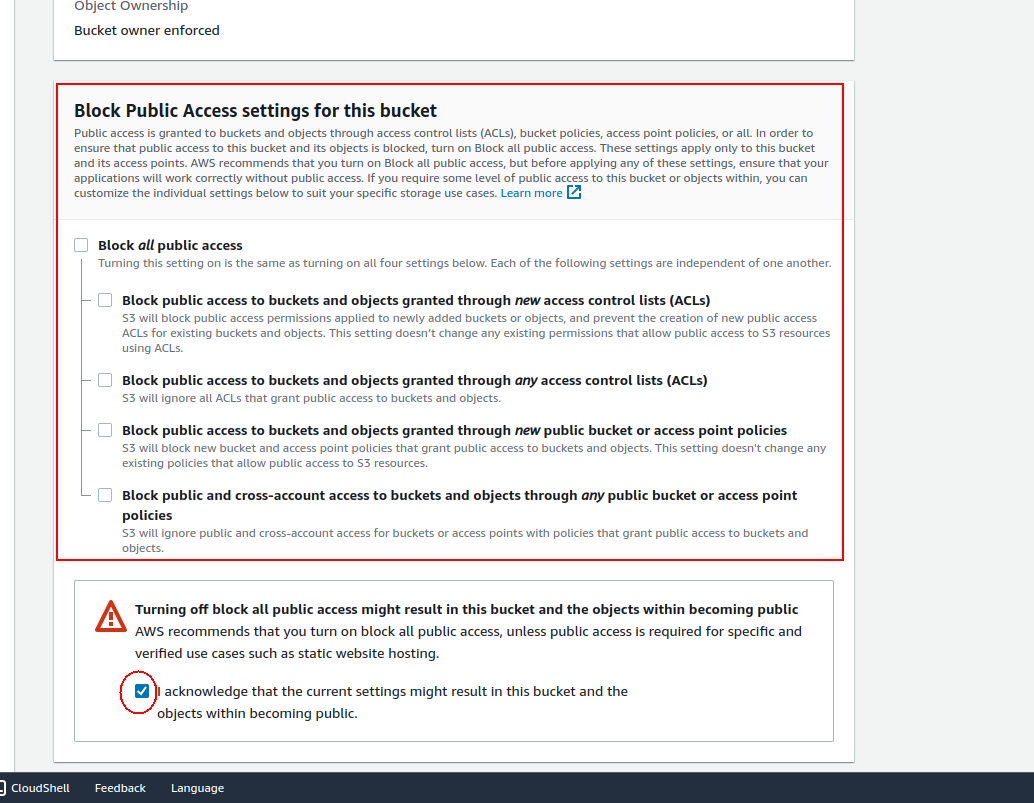
* Login to your AWS account which has permission to create and configure S3 Bucket.
* Navigate to S3 Service.



### Create S3 Bucket:

* Click on create bucket this will open up the create bucket configurations enter a unique and descriptive bucket name. ( Use same name as your purchased domain name)
* Choose the region where you want to create your bucket. 

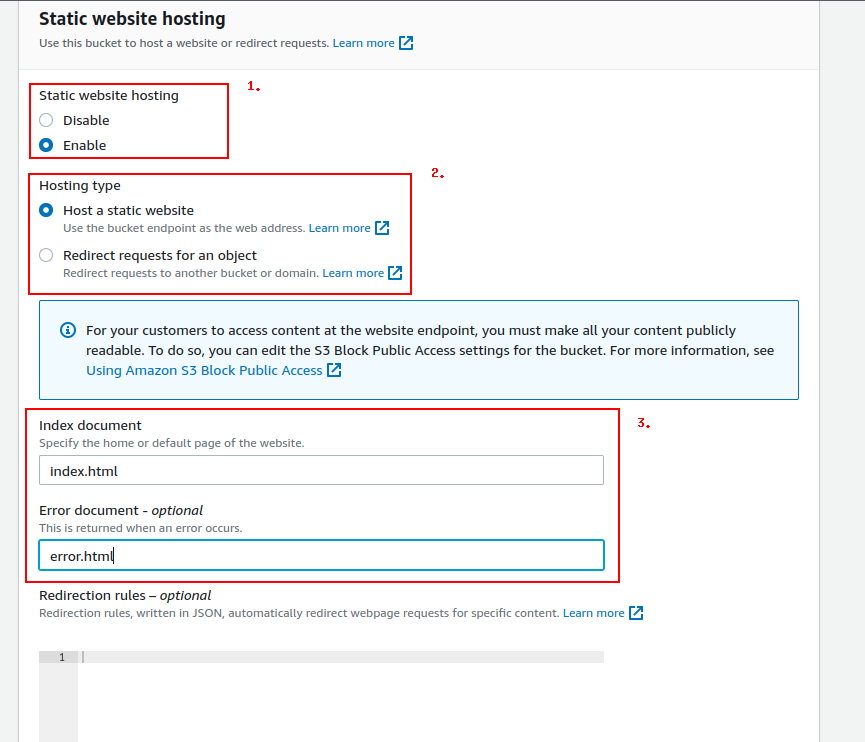
### Disable “Block Public Access”

* By disabling "Block Public Access" in S3, you can enable public access to your bucket when necessary, such as hosting a static website, while still maintaining granular control over who can access your data and how it is shared.

### Enable Static Website Hosting:

After creating the bucket go to the created s3 bucket and enable Static Website Hosting by

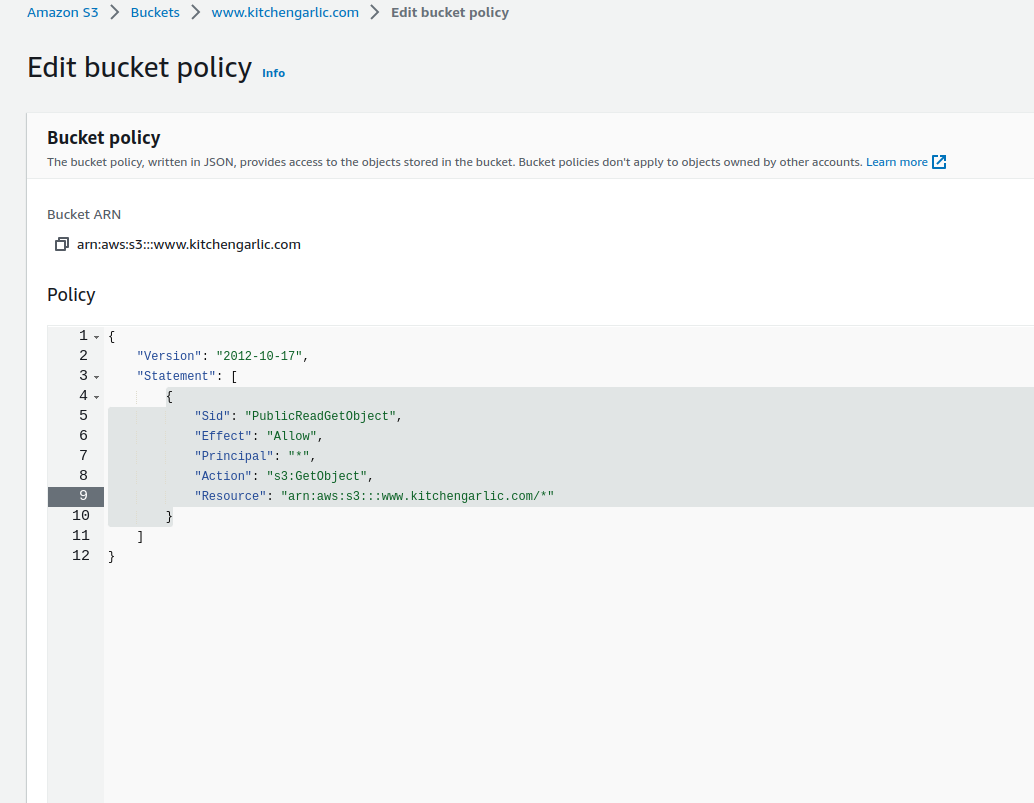
* Go to the "Properties" tab and scroll down to the "Static website hosting" section.
* Click on the "Edit" button.
* Select the "Enable" option for the "Static website hosting" feature.
* Specify the index document (e.g., index.html) and, if needed, the error document (e.g., error.html).
* Click on save button to save the setteings



### Add Bucket Policy:

Edit bucket policy to allow the public access to read only.

* In the bucket properties, go to the "Permissions" tab.
* Click on the "Bucket Policy" button.
* Add the follwing bucket policy to allow read only permissions and click save changes button

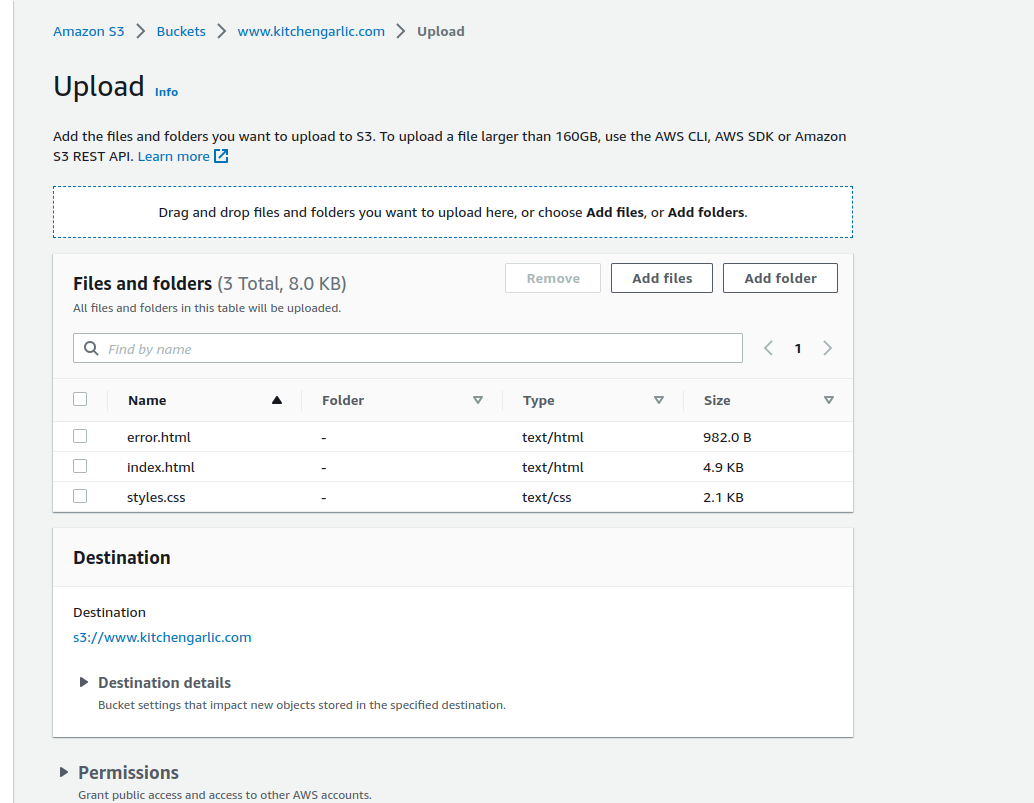


## Upload Website Files:

### Upload Website Files:

Now that our bucket is ready and configured to host static website it’s time to upload our website files to the s3 bucket for this follow the following steps.

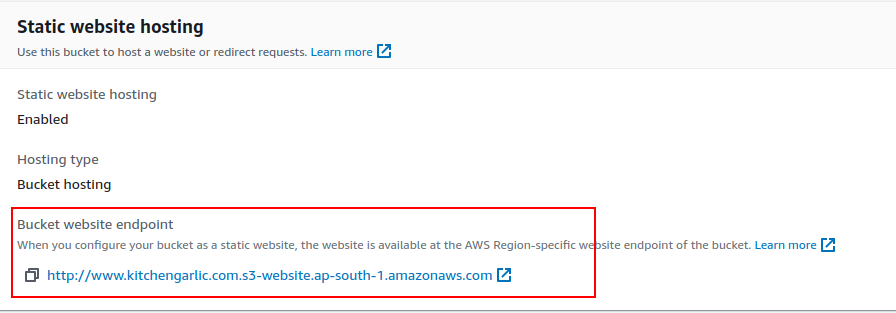
* Go back to the "Overview" tab of your S3 bucket.
* Click on the "Upload" button.
* Select the website files (HTML, CSS, JavaScript, images, etc.) from your local machine and upload them to the bucket.
* Ensure that the index document (e.g., index.html) is included in the uploaded files.



### Check Website Accessibility on S3 endpoint.

After uploading files it’s time to check if the website is accessible at provided s3 bucket endpoint for this follow these steps.

* Open your created bucket, click on its name to access the bucket properties.
* In the bucket properties, you will find the S3 bucket endpoint listed. It will be displayed as the "Endpoint" or "Website Endpoint" depending on the bucket configuration.
* Note down the S3 bucket endpoint URL. It should be in the format bucket-name.s3.amazonaws.com.
* Open in a web browser to see if your website is accessible.



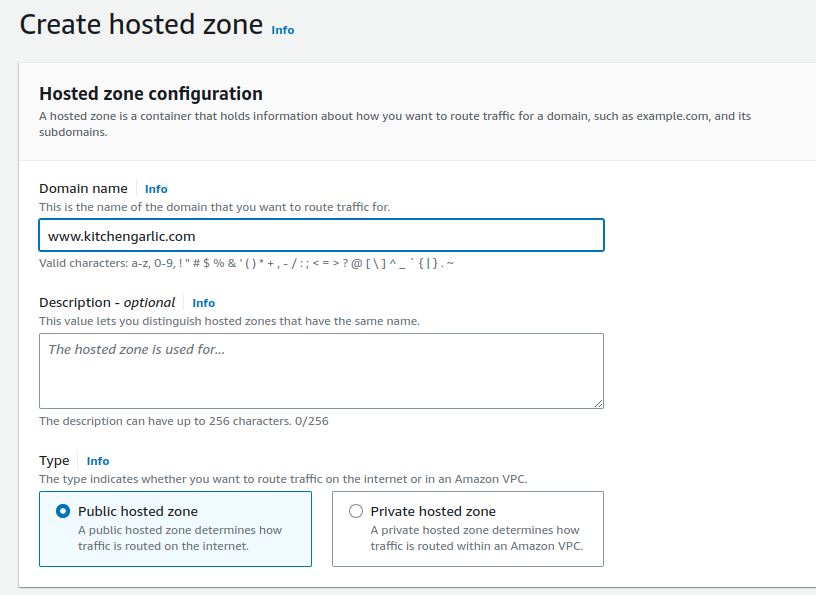
## Domain Configuration:

### Purchase Domain:

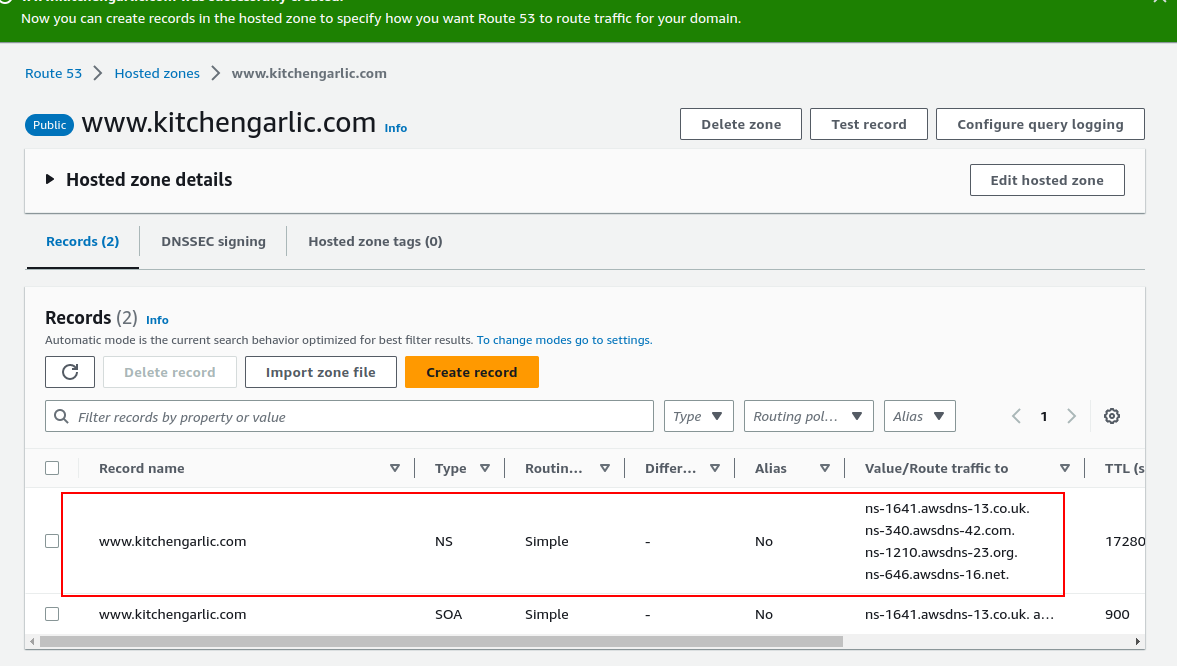
After S3 configuration next step is to configure DNS for this first step is to have a domain ready you can purchase domain for any domain provider such as Godady, NameCheap e.t.c for this example we will follow the domain purchased from NameCheap [www.kitchengarlic.com](http://www.kitchengarlic.com).

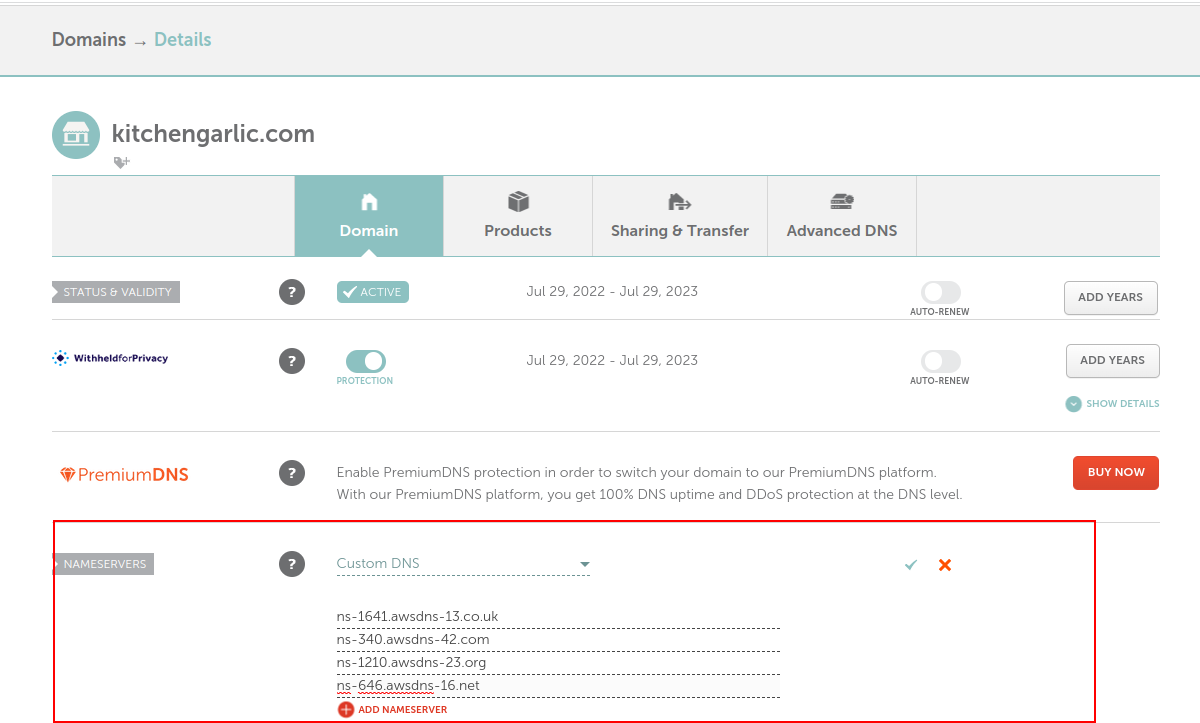
### Create Hosted Zone in Route 53:

* Search for Route53 in AWS and choose hosted zone.
* Click on create hosted zone and enter the domain you purchased.
* Add any description you want and choose public or private hosted zone based on your requirement.
* Click save to create hosted zone.



### Update DNS Records in your domain provider DNS settings:

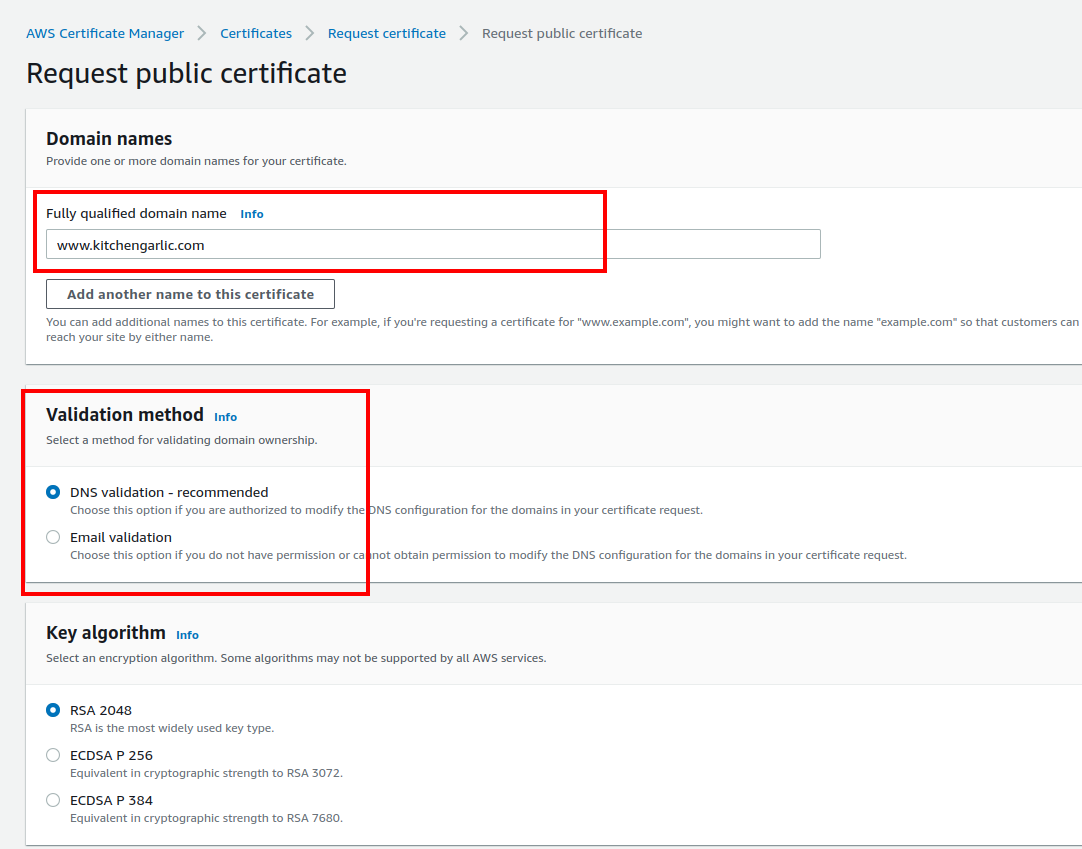
* For NameCheap follow below steps.
* Open your newly created hosted zone and note down entries in the records against the NS type.
* Add the DNS entries in your domain provider settings for NameCheap go to your domain settings find NAMESERVERS and configure Custom DNS.
* DNS changes may take some time to propagate. Wait for the changes to propagate across the DNS network, which can typically take a few minutes to a few hours.

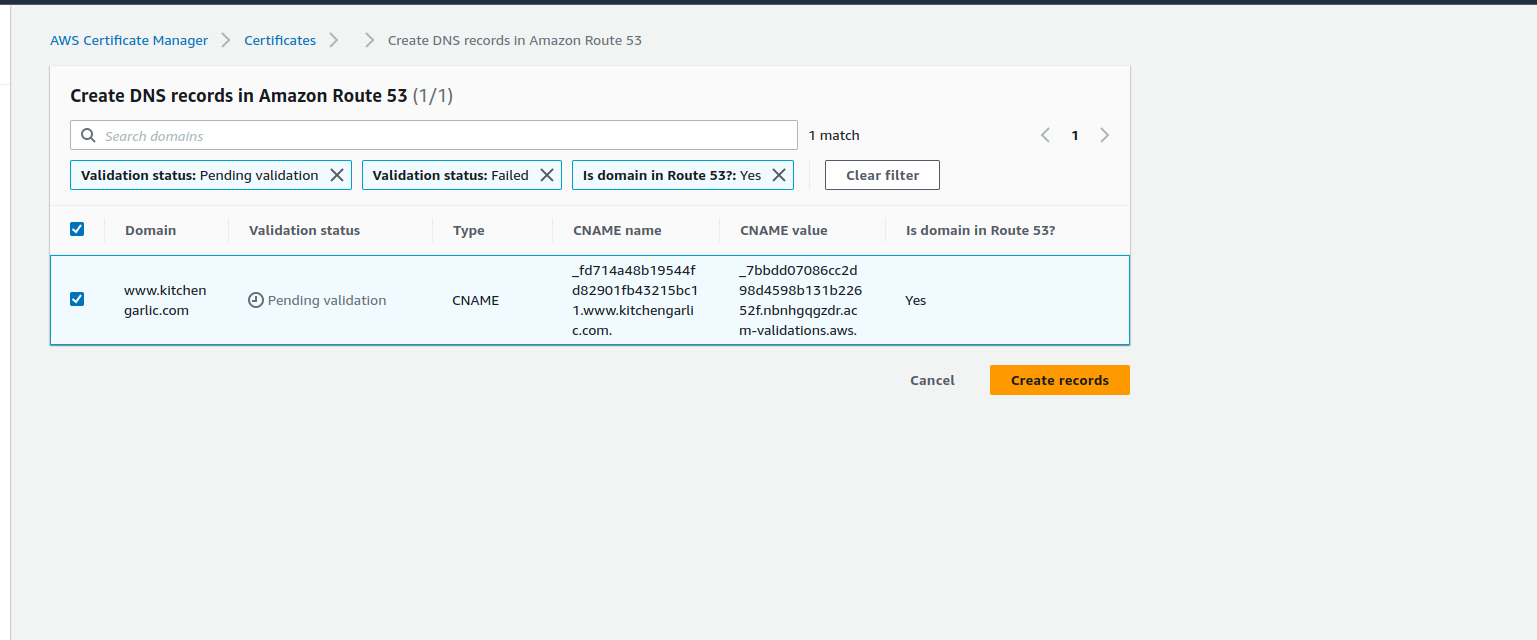


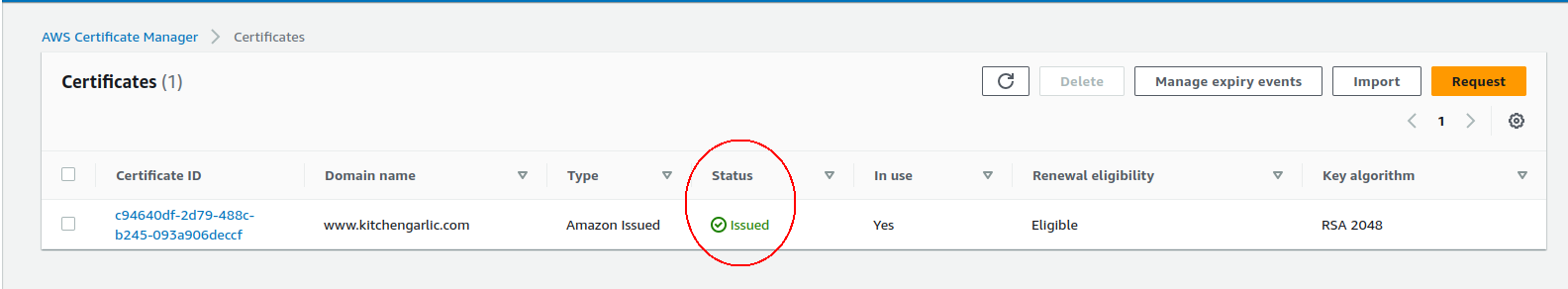
## Security and Performance:

### Request a SSL certificate.

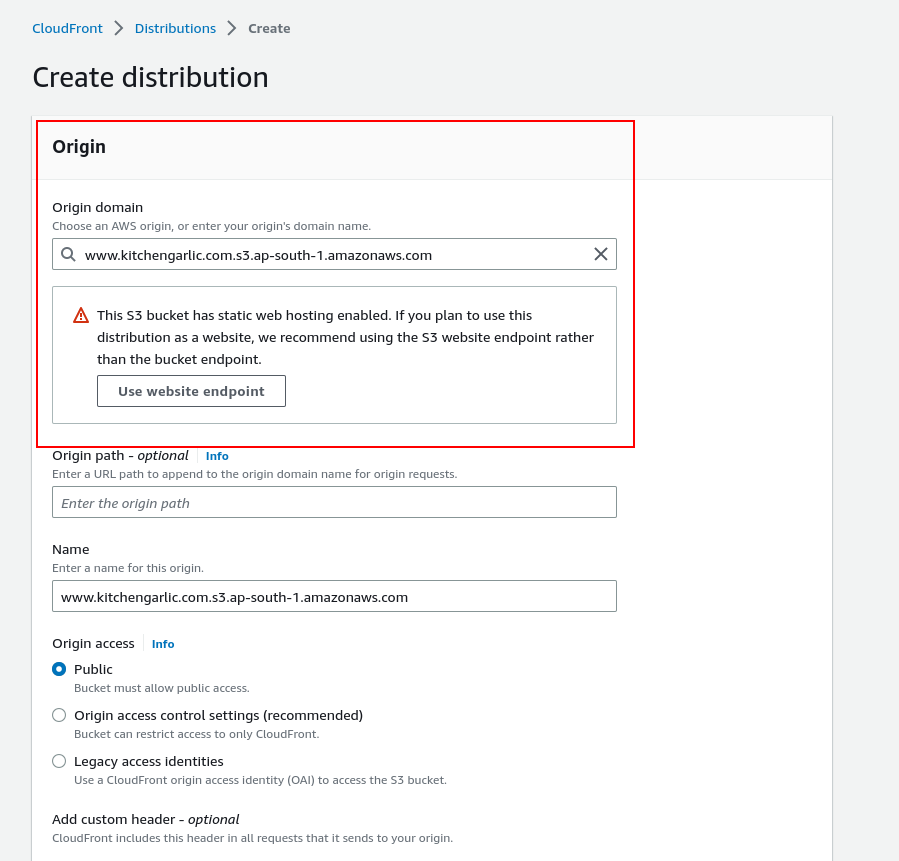
* Navigate to the ACM Service In the AWS Management Console, search for and select the "ACM" service.
* Request a Certificate In the ACM dashboard, click on the "Request a certificate" button.
* Choose Request a public certificate select the option "Request a public certificate" to request a certificate for a publicly accessible domain.
* In the "Domain name" field, enter the domain names for which you want to request the SSL certificate.
* Select Validation Method (DNS Validation) ACM will provide you with DNS records that you need to add to your DNS configuration for domain ownership validation.
* Click on request certificate

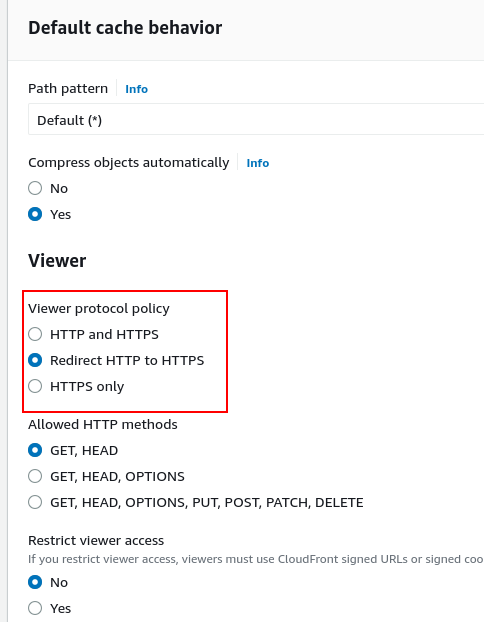


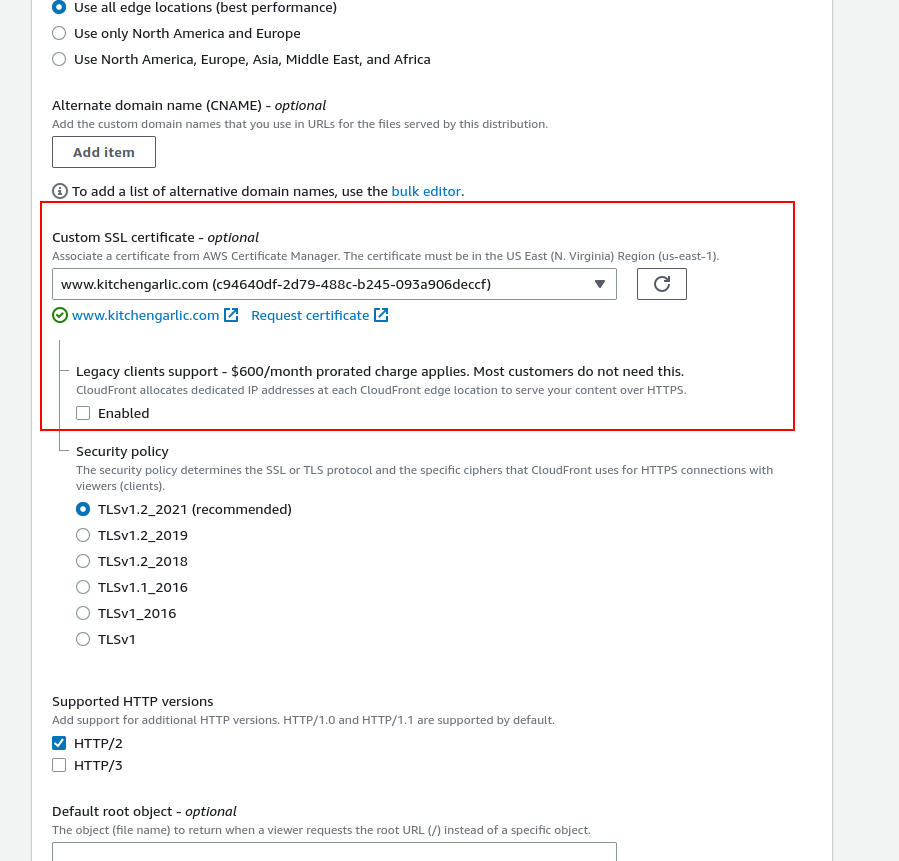
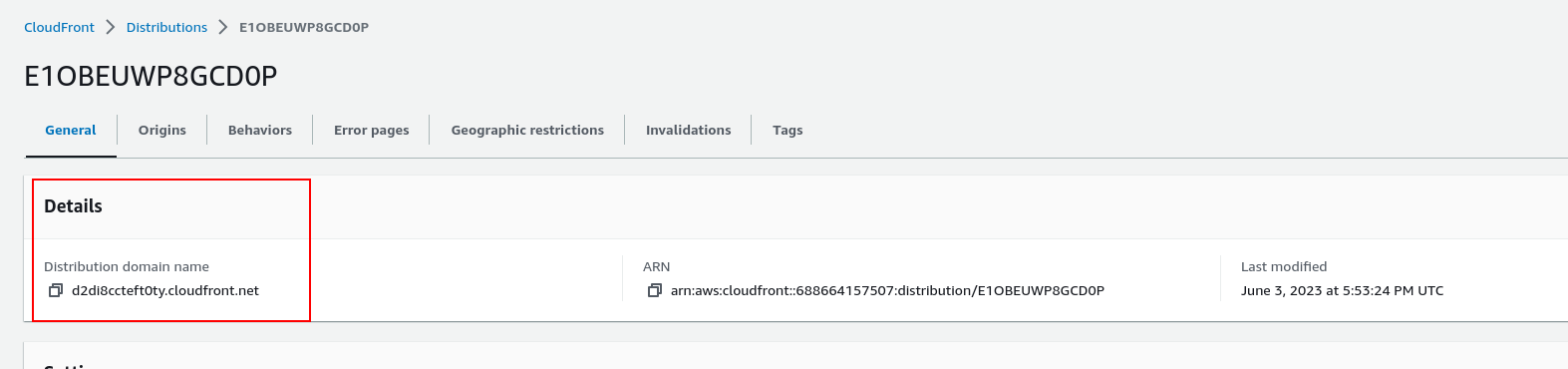
* Complete the Validation Process for DNS validation, you will need to add the provided DNS records to your DNS configuration.
* Open your requested SSL Certificate
* Click on add Record to Route53
* Click on create Record this will verify the ownership
* Wait for the validation process to complete
* Once the validation process is completed successfully, ACM will issue the SSL certificate. You can view and manage your certificates in the ACM dashboard.



### Create a CloudFront distribution.

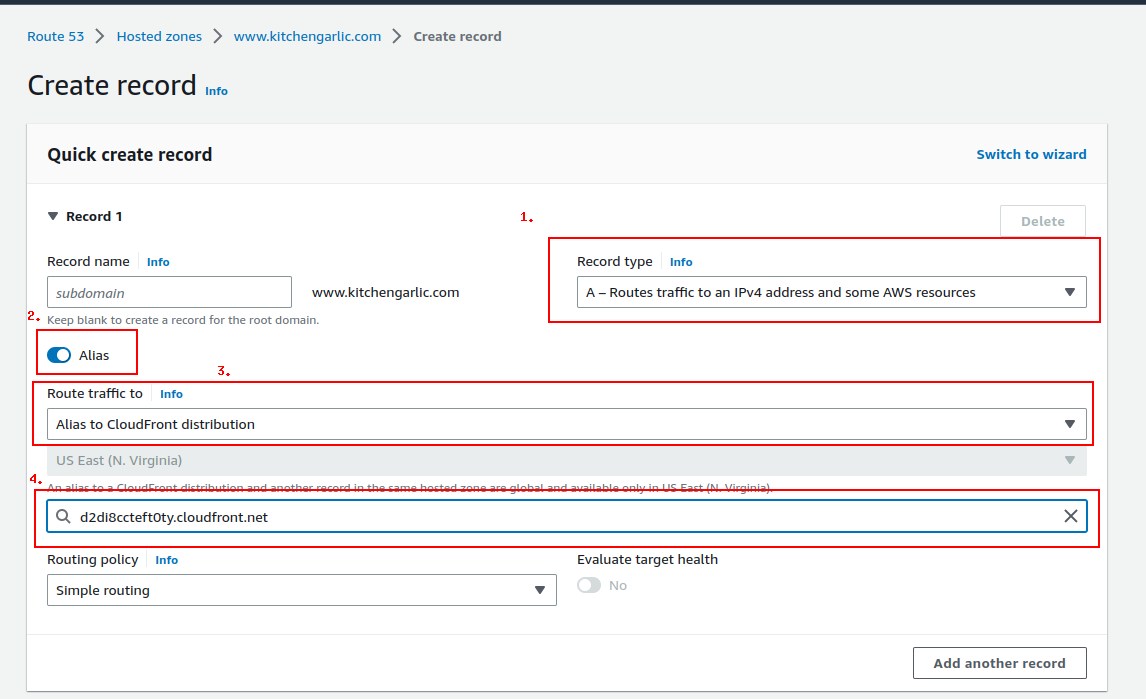
* Navigate to the CloudFront Service In the AWS Management Console, search for and select the "CloudFront" service.
* In the CloudFront dashboard, click on the "Create Distribution" button.
* Select the delivery method for your content. You can choose between "Web" and "RTMP." For static websites, select "Web."
* Origin Settings: Enter the domain name or S3 bucket endpoint of your origin server where the content is hosted.
* Set the origin access to public access
* Default Cache Behavior Settings: Configure the caching behavior and set it to redirect HTTP to HTTPS



* In the custom SSL certificate select the certificate that is created in the previous section.
* Click on create distribution and wait for it to be deployed. It may take some time for the CloudFront distribution to deploy and become fully active. You can monitor the status in the CloudFront dashboard.
* After the deployment is complete navigate to your newly created CloudFront distribution check for distribution domain name.
* Copy the link and check in browser if the website is accessible on newly created distribution end point. 

### Create A Record of CloudFront distribution in Route53.

* Navigate to the Route 53 Service: In the AWS Management Console, search for and select the "Route 53" service.
* Select the Hosted Zone Click on the "Create Record Set" button to create a new record set.
* Check the Alias toggle button and add A type record.
* Select “Alias to CloudFront distribution” from Route traffic to dialoge box.
* Selecte the CloudFront distribution endpoint from the list of available endpoints.
* Click add another record and now you can access your static website on your domain name i.e [www.kitchengarlic.com](http://www.kitchengarlic.com)



# Troubleshooting:

Here are the common problems that can occur during static website hosting and how to solve them.

## Website not accessible after configuring S3 bucket for website hosting:

* Ensure that the S3 bucket is configured properly for static website hosting, including index documents and error documents.
* Verify that the S3 bucket and its objects have the correct permissions set to allow public access.
* Double-check the bucket name and the correct region settings.

## **DNS records not propagating or taking a long time to propagate:**

* DNS propagation can take time, typically ranging from a few minutes to 24 hours. Wait for some time and check again.
* Confirm that the DNS records are set correctly, including the correct record type (such as A, CNAME, or ALIAS) and the target value.
* Clear the DNS cache on your local machine or try accessing the website from a different network or device.

## Website not accessible via the domain/subdomain:

* Check the DNS records to ensure that they are correctly configured to point to the S3 bucket or CloudFront distribution.
* Confirm that the domain/subdomain is correctly registered and managed with the DNS provider (e.g., Namecheap, GoDady).
* Validate that the domain/subdomain is correctly spelled and does not contain any typos.
* Verify that any SSL certificates are properly configured and associated with the domain/subdomain.

## SSL certificate errors or HTTPS not working:

* Ensure that the SSL certificate is valid and properly associated with the domain/subdomain.
* Double-check that the certificate is issued for the correct domain/subdomain and has not expired.
* Confirm that the SSL certificate is correctly configured in the load balancer, CloudFront distribution, or web server settings.
* Verify that the DNS records (A, CNAME, or ALIAS) for the domain/subdomain are correctly configured to point to the appropriate HTTPS endpoint.

## CloudFront distribution issues:

* Confirm that the CloudFront distribution is properly configured and associated with the correct S3 bucket or other origin.
* Check the CloudFront settings, such as caching behavior, origin settings, and SSL certificate configuration.
* Ensure that the CloudFront distribution is deployed and in the "Enabled" state.
* Validate that the domain/subdomain is correctly configured in the CloudFront distribution's alternate domain names (CNAMEs) or SSL certificate settings.

# Conclusion:

In conclusion, the task involved deploying a static website using AWS services, including S3, Route 53, and CloudFront. The process included configuring the S3 bucket for website hosting, uploading the website files, setting up domain configuration and DNS records, enabling HTTPS using SSL certificates, and implementing caching and content delivery mechanisms for better performance.

Throughout the task, we followed best practices and ensured proper configuration to create a secure, scalable, and highly available static website. We encountered common challenges such as DNS propagation, SSL certificate configuration, and CloudFront distribution setup, but we addressed these issues through troubleshooting and appropriate solutions.

By successfully completing the task, we have achieved our goal of deploying a static website that is accessible via a domain/subdomain, secured with HTTPS, and optimized for performance using AWS services. The detailed documentation provided will serve as a reference for future maintenance and updates to the website.

Overall, this task has provided valuable hands-on experience in working with AWS services and configuring a static website, showcasing the ability to deploy and manage web applications effectively in a cloud environment.

# **References:**

Here are some references from AWS documentation and NameCheap's support articles:

1. AWS S3 Bucket:

- AWS S3 Developer Guide: <https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>

1. AWS CloudFront:

- AWS CloudFront Developer Guide: <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Welcome.html>

1. AWS Route 53:

- AWS Route 53 Developer Guide: <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>

1. AWS Certificate Manager (ACM):

- ACM User Guide:

<https://docs.aws.amazon.com/acm/latest/userguide/welcome.html>

1. NameCheap:

- NameCheap Knowledgebase - How to Register Private Nameservers:

<https://www.namecheap.com/support/knowledgebase/article.aspx/767/10/how-to-register-private-nameservers>

These resources provide detailed information, step-by-step guides, and best practices for setting up and configuring the respective AWS services and NameCheap features.