**Multi-Service Configuration on AWS Including Website Hosting**

**Project Report**

Project (IAI-851)

Degree

**BACHELOR OF TECHNOLOGY (CSE)**

|  |  |
| --- | --- |
| PROJECT GUIDE:  **Mr. Ghufran Khan**  **Senior Faculty** | SUBMITTED BY:  **Siddhartha Mishra (TCA196006)** |

May, 2023



**FACULTY OF ENGINEERING & COMPUTING SCIENCES**

**TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD**

**DECLARATION**

We hereby declare that this Project Report titled **Multi-Services Configuration on AWS Cloud** submitted by us and approved by our project guide, Faculty of Engineering &Computing Sciences. Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Student Name:** | Siddhartha Mishra | | | Signature | |
|  | |
|  | | |  | |
|  |  | | |  | |
| **Project Guide :** | Mr. Ghufran Khan | | | Signature | |

Table of Contents

[1 Project Title 4](#_Toc31392321)

[2 Problem Statement 4-5](#_Toc31392322)

[3 Project Description 6](#_Toc31392323)

[3.1 Scope of the Work](#_Toc31392324) 6

[3.2 Project Modules](#_Toc31392325) 7-15

[3.3 Context Diagram (High Level)](#_Toc31392326) 16

[4 Implementation Methodology 17](#_Toc31392327)

[5 Technologies to be used 17](#_Toc31392328)

[5.1 Software Platform 17](#_Toc31392329)

[5.2 Hardware Platform 17](#_Toc31392330)

[5.3 Tools, if any 18](#_Toc31392331)

[6 Advantages of this Project 19](#_Toc31392332)

[7 Future Scope and further enhancement of the Project 19](#_Toc31392334)

[8 Project Repository Location 20](#_Toc31392335)

[9 Conclusion 21](#_Toc31392337)

[10 References 22](#_Toc31392338)

**Appendix**

**A:Data Flow Diagram (DFD)**

**B:Entity Relationship Diagram (ERD)**

# Project Title

Multi-Service Configuration on AWS Including Website Hosting

# Problem Statement

When you enable static website hosting for your bucket, you enter the name of the error document (for example, **404.html**). After you enable static website hosting for the bucket, you upload an HTML file with this error document name to your bucket.

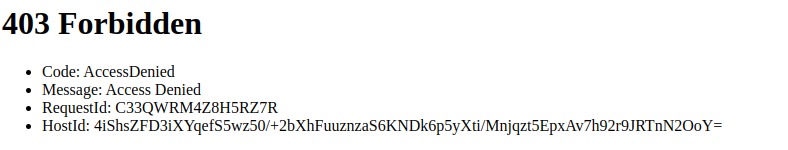
1. Create an error document, for example 404.html.
2. Save the error document file locally.

The error document name is case sensitive and must exactly match the name that you enter when you enable static website hosting. For example, if you enter 404.html for the **Error document** name in the **Static website hosting** dialog box, your error document file name must also be 404.html.

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** list, choose the name of the bucket that you want to use to host a static website.
3. Enable static website hosting for your bucket, and enter the exact name of your error document (for example, 404.html). For more information, see Enabling website hosting.

After enabling static website hosting, proceed to step 6.

1. To upload the error document to your bucket, do one of the following:
   * Drag and drop the error document file into the console bucket listing.
   * Choose **Upload**, and follow the prompts to choose and upload the index file.



**3. Project Description**

The goal of this project is to host a highly secure and reliable website using AWS S3 and configure with Cloudfront and configure with Route53 by domain name. It is a virtual private server that is used to host numerous websites. You can experiment with working on AWS by hosting a website. You can create a website connected to the Route53. To host the website building task easier, you can use S3 with AWS Cloudfront. It provides SSD-based storage and comes pre-configured.

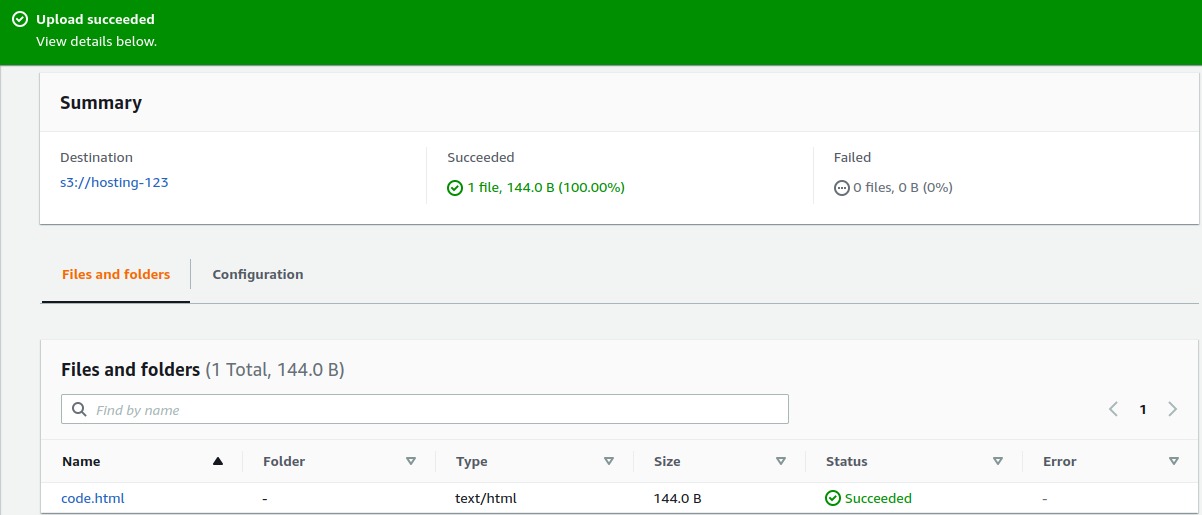
## 3.1 Scope of the Work

We include generally available services in the scope of our compliance efforts based on the expected use case, feedback and demand. If a service is not currently listed as in scope of the most recent assessment, it does not mean that you cannot use the service. It is part of the [shared responsibility](https://aws.amazon.com/compliance/shared-responsibility-model/) for your organization to determine the nature of the data. Based on the nature of what you are building on AWS, you should determine if the service will process or store customer data and how it will or will not impact the compliance of your customer data environment.

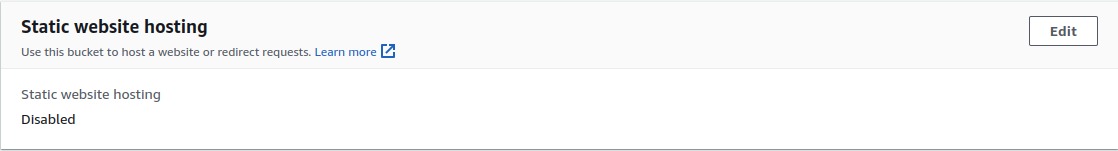
## 

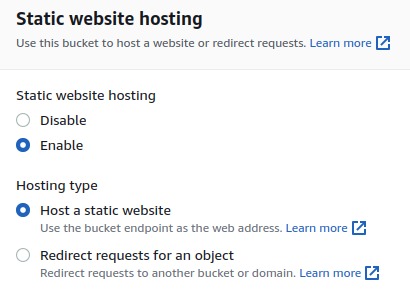
## 3.2 Project Modules

1. In this project create an s3 bucket in AWS after that upload the HTML file.



2. Go to the file properties and click static website hosting.

3. Click the Enabled and select the host a static website.

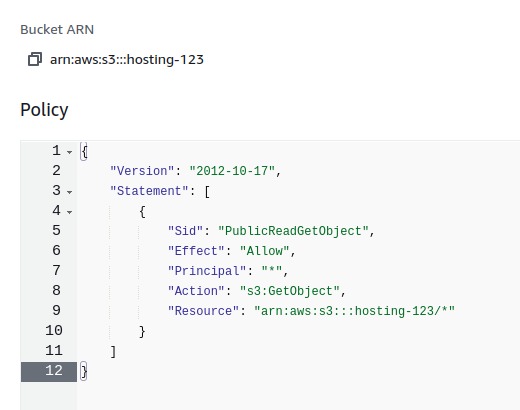


4. Click the Save Changes.

5. Go to the permissions and click the bucket policy.

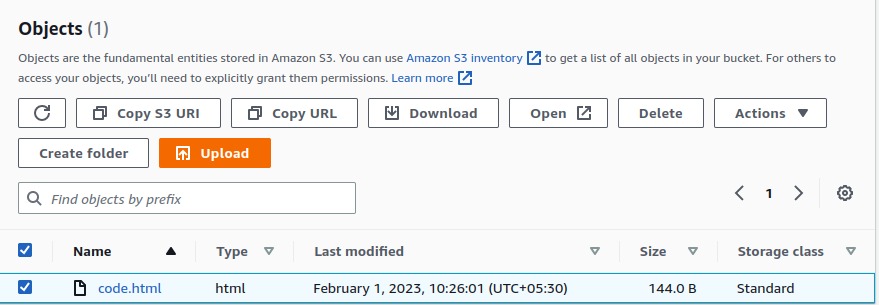
6. Write a JSON file according to use and give the permissions.

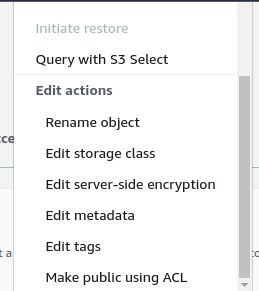
7. Change the ARN (Amazon Resource Name).

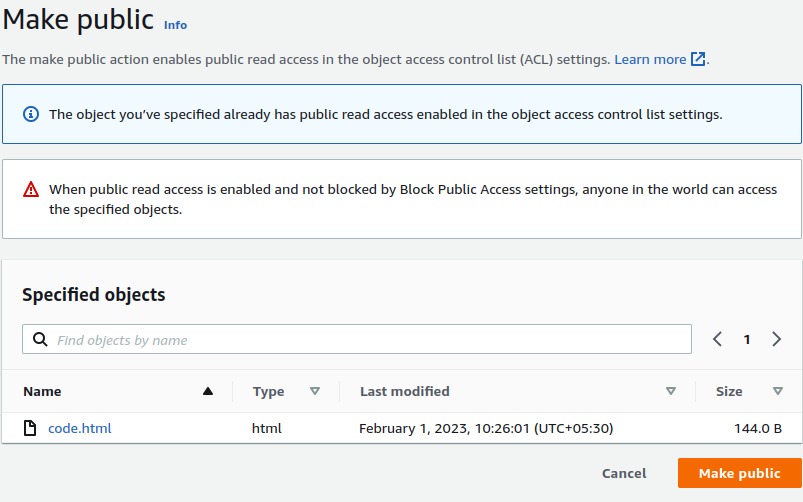


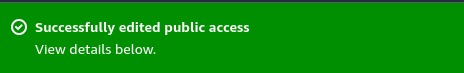
8. Select the HTML File and go to the actions.

9. Click on the Make a Public Using ACL.

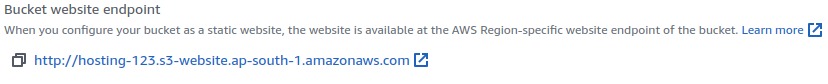






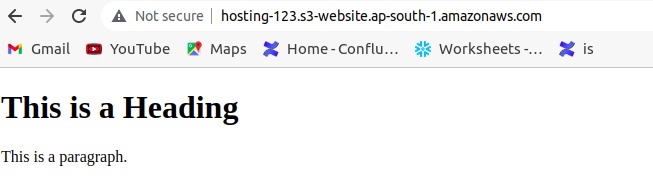


10. After this process, Click on the Properties and Copy the Static Website URL.



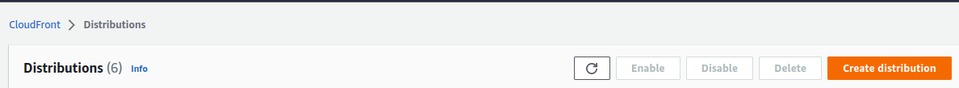
11. Paste the URL Browser.

12. After this Successfully Host.

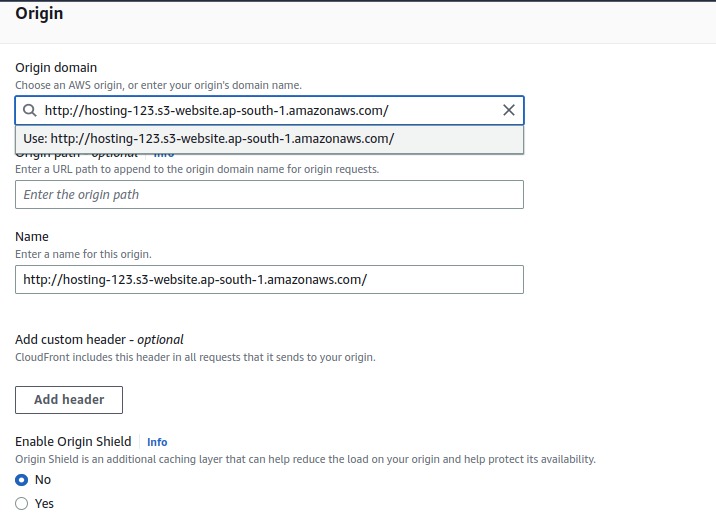


13. After the website host successfully through the s3 bucket.

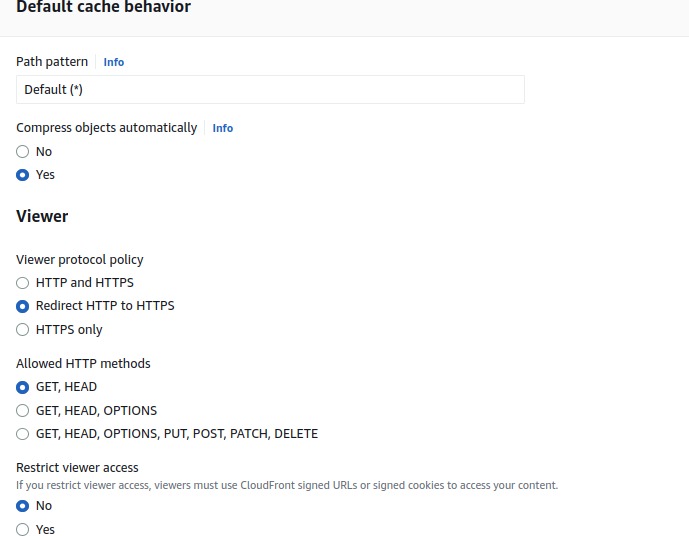
14. Go to the CloudFront and create a Distribution.



15. Copy the URL of website and paste it origin domain.

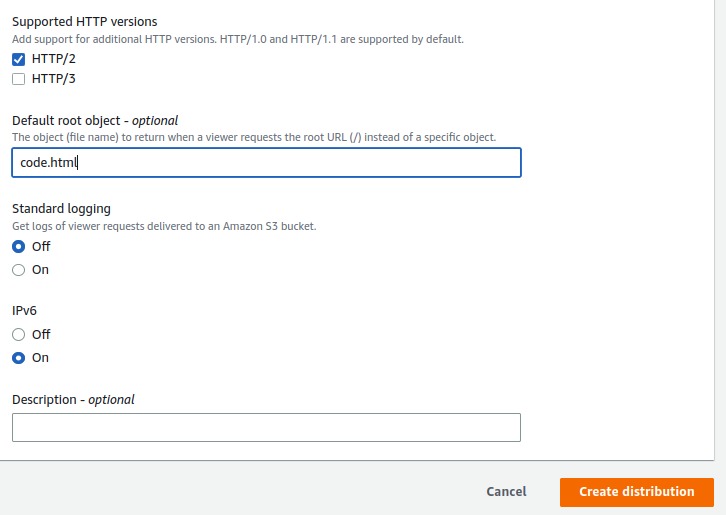


16. Change the viewer protocol policy and click redirect HTTP to HTTPS



16. Fill the file name .

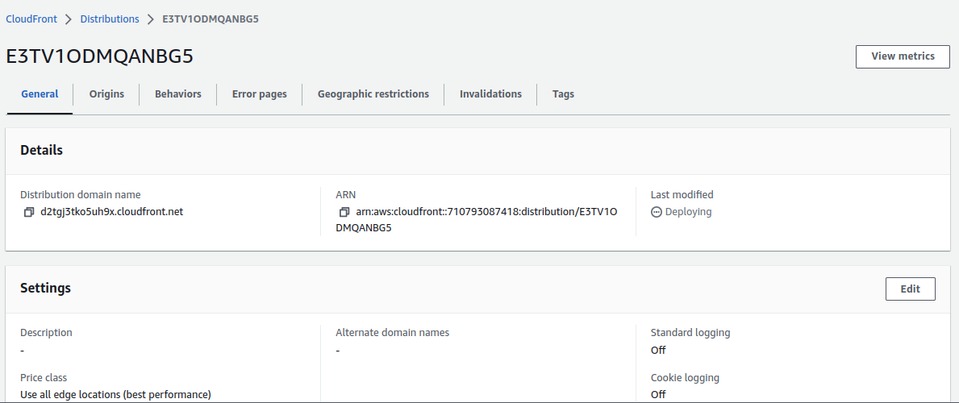
17. Click the create distribution.



18. Distribution create successfully.

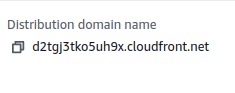
WhatsApp Image 2023-04-26 at 8.42.34 PM (1).jpeg

19. They take a some time deploying the distribution.



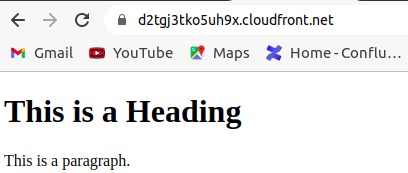
20. After deploying that go to the distribution.

21. Copy to the domain name.

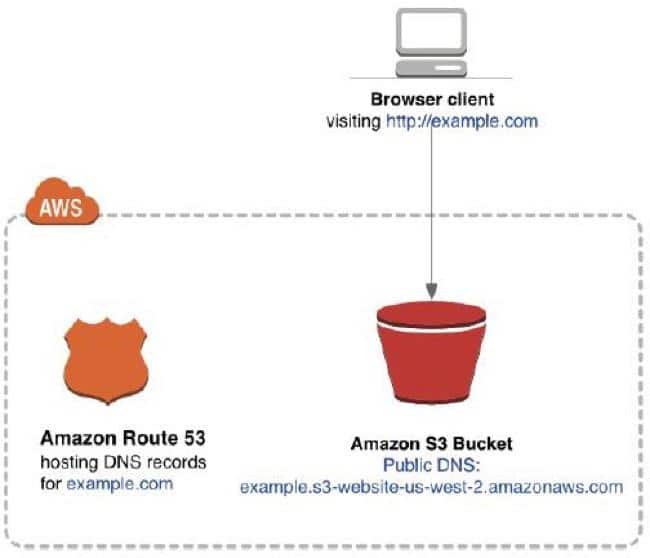


22. Paste the URL for a Browser.

23. Show the successfully.



## 3.3 Context Diagram (High Level)



# 4. Implementation Methodology

Static websites deliver HTML, JavaScript, images, video and other files to your website visitors and contain no server-side application code, like PHP or ASP.NET. They typically are used to deliver personal or marketing sites.

Static websites are very low cost, provide high-levels of reliability, require no server administration, and scale to handle enterprise-level traffic with no additional work.

# 5. Technologies to be used

## 5.1 Software Platform

1. AWS Platform

**b**) Operating System(Linux)

## 5.2 Hardware Platform

* RAM - Minimum 4gb.
* Hard Disk - Minimum 32gb.
* Processor i-3, AMD 3 upwards**.**

## 5.3 Tools, if any

* **S3 bucket**-A bucket is a container for objects stored in Amazon S3. You can store any number of objects in a bucket and can have up to 100 buckets in your account. To request an increase, visit the Service Quotas Console.
* **CloudFront**- Amazon CloudFront is a web service that gives businesses and web application developers an easy and cost effective way to distribute content with low latency and high data transfer speeds.
* **Route53**-Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. Route 53 connects user requests to internet applications running on AWS or on-premises

# 6. Advantages of this Project

* User-friendly.
* Flexible.
* Secure.
* Cost-effective.
* Reliable.
* Scalable and Elastic.
* Highly Performance.

**7. Future Scope and Further enhancement of the project**

The AWS Certification helps you pursue career paths like AWS Solutions Architect, AWS Engineer, DevOps Engineer, and Cloud Architect among others. In order to avail of these opportunities, you need structured AWS training with an updated curriculum as per current industry requirements and best practices.

# 

# 8. Project Repository Location

| **S#** | **Project Artifacts (softcopy)** | **Location** (Mention Lab-ID, Server ID, Folder Name etc.) | **Verified by Project Guide** | **Verified by Lab In-Charge** |
| --- | --- | --- | --- | --- |
|  | Project Synopsis Report (Final Version) | Github | Name and Signature | Name and Signature |
|  | Project Progress updates | Github | Name and Signature | Name and Signature |
|  | Project Requirement specifications | Github | Name and Signature | Name and Signature |
|  | Project Report (Final Version) | Github | Name and Signature | Name and Signature |
|  | Test Repository | Github | Name and Signature | Name and Signature |
|  | Project Source Code (final version) with executable | Github | Name and Signature | Name and Signature |

# 9. Conclusion

There are numerous architectural and conceptual considerations when you are contemplating migrating your web application to the AWS Cloud. The benefits of having a cost-effective, highly scalable, and fault-tolerant infrastructure that grows with your business far outstrips the efforts of migrating to the AWS Cloud.

# 

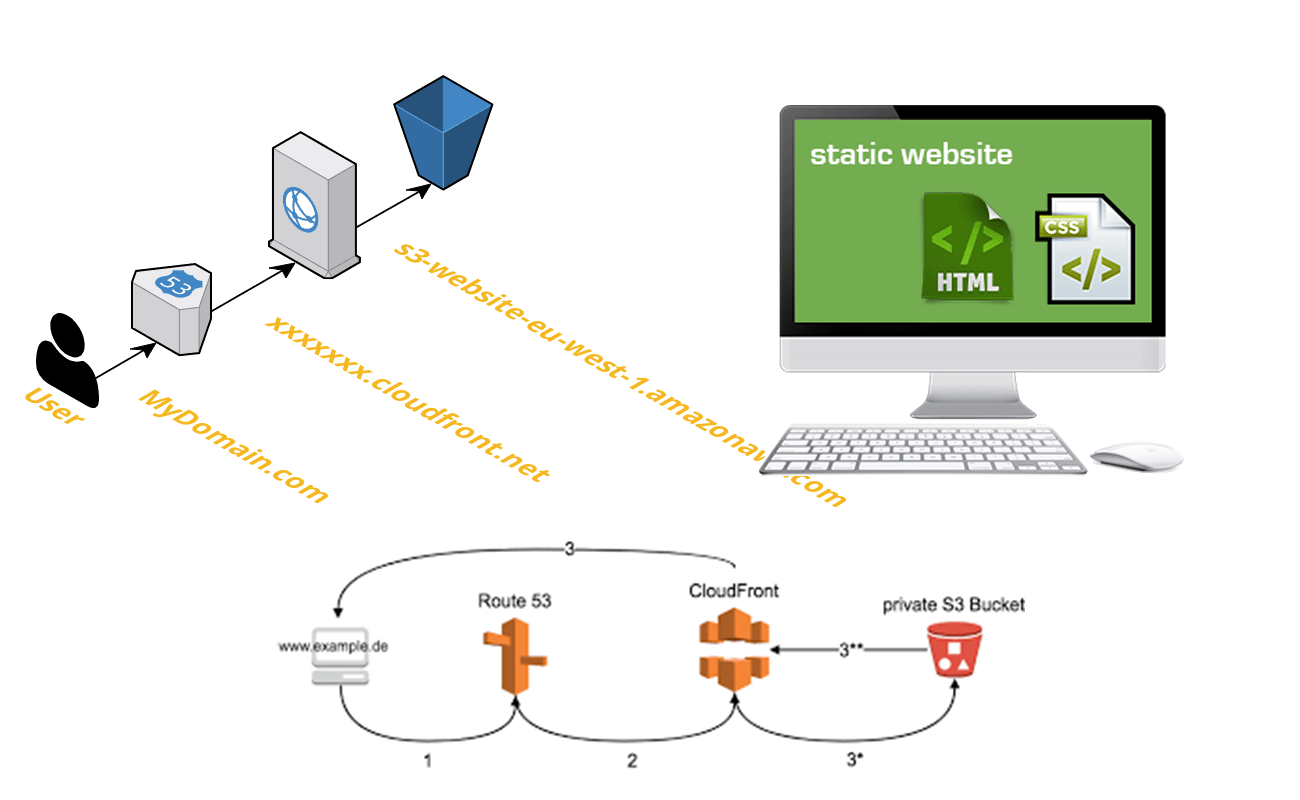
# 10. References

* [*https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html*](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html)
* [*https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html)
* [*https://medium.com/@kyle.galbraith/how-to-host-a-website-on-s3-without-getting-lost-in-the-sea-e2b82aa6cd38*](https://medium.com/@kyle.galbraith/how-to-host-a-website-on-s3-without-getting-lost-in-the-sea-e2b82aa6cd38)

**Annexure A**

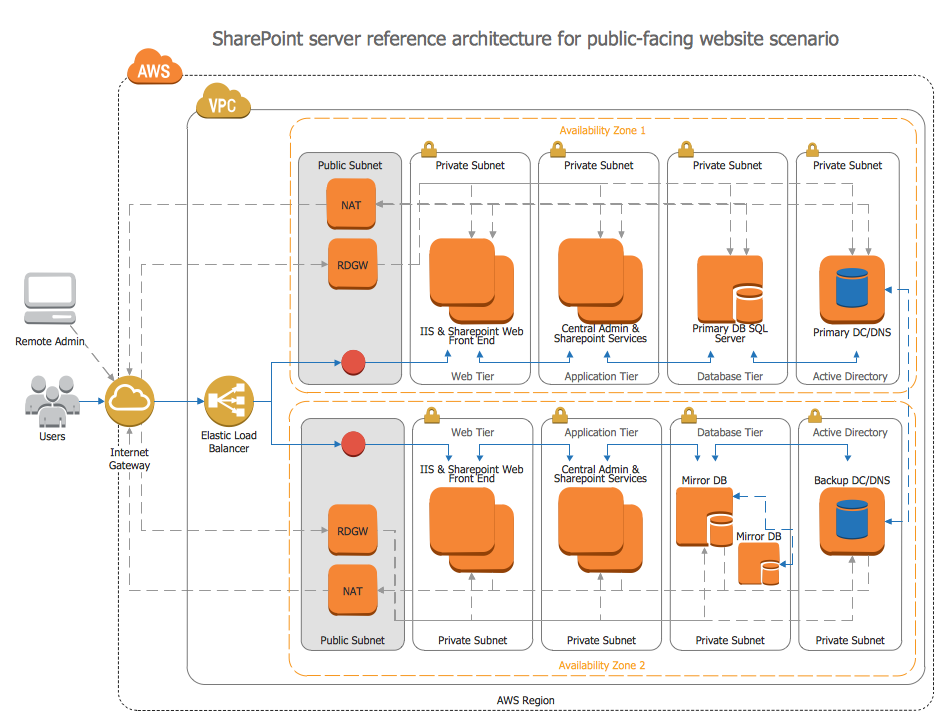
**Data Flow Diagram (DFD)**

**(Mandatory)**

****

**Annexure B**

**Entity-Relationship Diagram**

****