AWS Static Website Hosting with Amazon S3

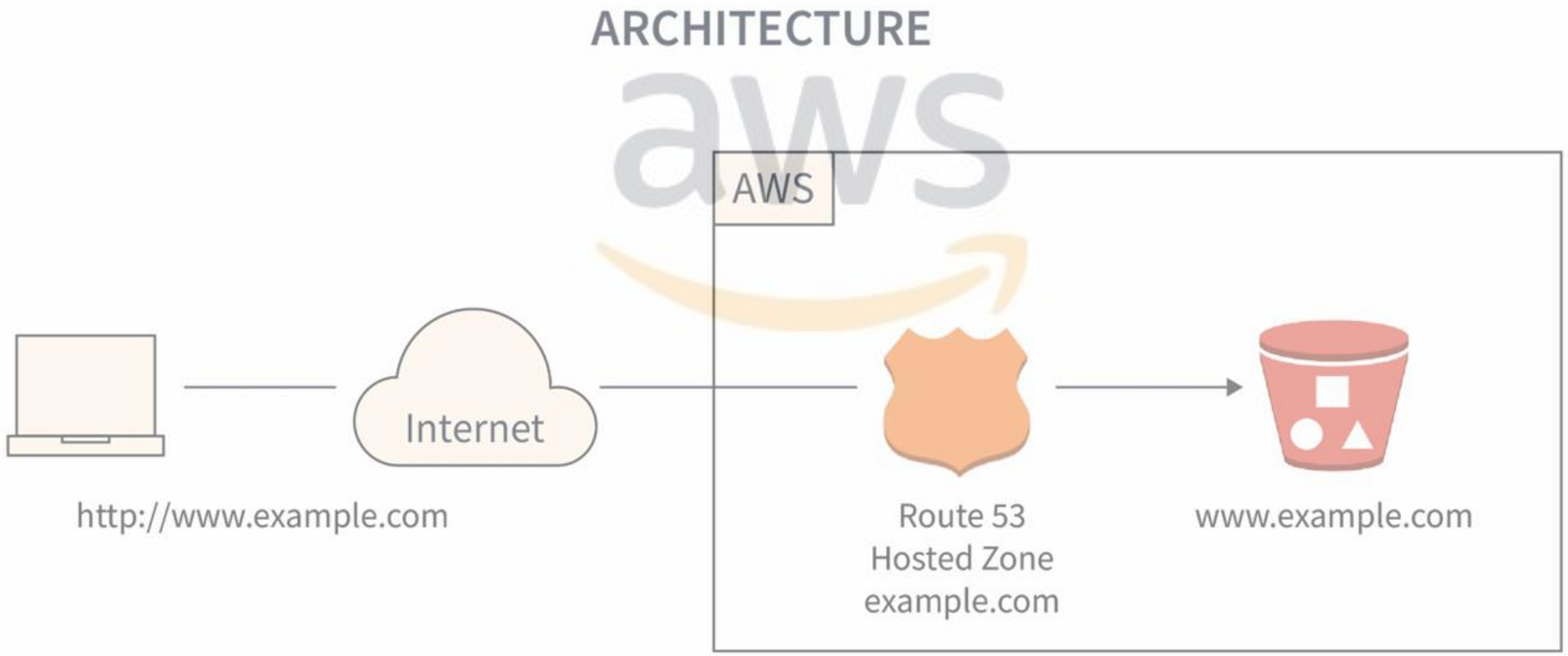
Project Overview:-

- This project involves creating and deploying a static website using Amazon Simple Storage Service(S3) for hosting.
- A static website is a collection of HTML, CSS, JavaScript, and other assets that are pre-built and do not require server-side processing.
- Leveraging the power of Amazon S3, this project showcases that simplicity and cost-effectiveness of hosting static content in the cloud.

Services used:-

Route53, Simple Storage Service(S3) , Content Delivery Network(CDN), CloudWatch

Architecture diagram 🏗 :-



Explanation:-

step 1: Create an S3 Bucket

Open the AWS Management Console:

Go to the AWS Management Console.

1. **Navigate to S3:**
   In the AWS Management Console, search for and select "S3" under the "Storage" category.

2. **Create a Bucket:**
   1. Click on the "Create bucket" button.
   2. Choose a globally unique name for your bucket (e.g., "your-website-name").
   3. Select the region for your bucket.
   4. Click "Next" through the remaining steps, leaving the default settings.

3. **Enable Static Website Hosting:**
   1. Select the newly created bucket.
   2. Navigate to the "Properties" tab.
   3. Click on "Static website hosting.“

4. **Configure Static Website Hosting:**
   1. Choose "Use this bucket to host a website."
   2. Set the "Index document" to the main HTML file (e.g., "index.html").
   3. Optionally, set the "Error document" for custom error pages.
   4. Click "Save."

Step 2: (Upload Your Website Files)🗁

1. Navigate to the "Overview" Tab:

In the S3 Management Console, click on the "Overview" tab for your bucket.

2. Upload Files:

1. Click the "Upload" button.
2. Select your HTML, CSS, JavaScript, and other static files.
3. Click "Next" through the remaining steps, leaving the default settings.
4. Click "Upload" to upload your files.

Step 3: (Configure Bucket Policy for Public Access)

1. Navigate to the "Permissions" Tab:

In the S3 Management Console, click on the "Permissions" tab for your bucket.

2. Bucket Policy:

1. Click on "Bucket Policy."
2. Add a policy that grants public read access to your objects. Here's a sample policy:

Bucket Policy Code:-

```
{
        "Version": "2012-10-17",
        "Statement“: [
                {
                        "Effect": "Allow",
                        "Principal": "*",
                        "Action": "s3:GetObject",
                        "Resource": "arn:aws:s3:::your-website-name/*"
                }
        ]
}
```

- Replace "your-website-name" with your actual bucket name.
- Click “save” to apply the policy.

Step 4: (Configure Website URL)

1. **Access Website Endpoint:**
    1. In the S3 Management Console, go to the "Static website hosting" tab.
    2. Find the "Endpoint" URL provided.

2. **Verify Your Website:**
    1. Open a web browser and enter the endpoint URL.
    2. You should see your static website.

Step 5: (Optional) (Set Up a Custom Domain with Route 53)

- If you want to use a custom domain

1. **Register a Domain:**

    Register a domain through a domain registrar (Freenom).

1. **Configure Route 53:**
    1. Open the Route 53 console.
    2. Create a hosted zone for your domain.
    3. Add a record set with the S3 website endpoint as the alias target.

2. **Update Domain Registrar Settings:**
    1. Update the domain registrar's DNS settings to use the Route 53 name servers.

Step 6: (Configure CloudFront (Optional)

- Create a CloudFront distribution with your S3 bucket as the origin.
- Configure CloudFront to connect our website globally.

CloudWatch:-

- It provides valuable insights into the operational health of the static website, aiding in efficient management and troubleshooting.

Challenges faced during implementation:-

1. Permissions and Security Configuration.
2. Cross-Origin Resource Sharing.
3. Custom Domain Setup.

❁ Solutions For Faced Challenges:-

Permissions and Security Configuration:

1. Properly configuring S3 bucket policies
2. Ensuring that your website files are publicly accessible while maintaining the security of sensitive data is crucial.

Cross-origin Resource Sharing (CORS):

- If your static website makes requests to resources on different domains, configuring CORS headers on your S3 bucket may be necessary to avoid browser security restrictions.

Custom Domain Setup:

- Configuring a custom domain, including DNS settings and integration with services like Route 53, can sometimes be complex. Domain registration, DNS propagation, and proper configuration are important considerations.

Future considerations 🔮 :-

- Enhanced Security Measures
- Integration of Advanced Features

Conclusion:-

🔑 The successful implementation of the static website hosting project on Amazon S3 has provide secure, scalable, and cost-effective solution.

The use of AWS services, including S3, enabled efficient data management.

The static website hosted on S3, coupled with a custom domain and HTTPS support, enhances the overall user experience, providing a secure and easily accessible platform.