REPORT

Task:

- To create a static website, which is a portfolio website and host the static website using Amazon S3. By creating a bucket, upload the essential files for hosting the static website and also add the necessary bucket policies.

Introduction:

- Amazon S3 (Simple Storage Service) is a highly scalable and secure cloud storage service provided by Amazon Web Services (AWS). While S3 is primarily used for storing and retrieving data, it can also be used to host static websites efficiently and cost-effectively.
- A static website is a website that consists of HTML, CSS, JavaScript, and other client-side files that are served to users as-is, without any server-side processing. These websites are ideal for content that doesn't change frequently, such as blogs, portfolios, documentation, or simple business websites.
- Hosting a static website in Amazon S3 offers several benefits:
    1. Cost-effective: Amazon S3 provides a cost-effective solution for hosting static websites, especially for low-traffic or personal sites. We should pay only for the storage used and the data transferred, making it highly affordable.
    2. Scalability: S3 is designed to handle massive amounts of data and high levels of traffic. It can scale effortlessly to accommodate increased website visitors without compromising performance.
    3. High availability: Amazon S3 guarantees high availability and durability for your website. It stores multiple copies of our data across multiple data centers, ensuring that your website remains accessible even if there are hardware failures or natural disasters.
- To host a static website in Amazon S3, you need to follow these steps:
    1. Create an S3 bucket: In the AWS Management Console, create a new S3 bucket and give it a unique name. This bucket will be used to store our website's files.
    2. Configure bucket properties: Enable static website hosting for the bucket by going to the bucket's properties and selecting the "Static website hosting" option. Specify the index document (e.g., index.html) that will be served when a user accesses our website.
    3. Upload your website files: Upload our HTML, CSS, JavaScript, and other static files to the S3 bucket. You can use the AWS Management Console, AWS CLI, or SDKs/APIs to upload the files.
    4. Set permissions: Make sure the bucket and its objects are publicly readable so that visitors can access our website. Configure the appropriate permissions in the bucket's permissions settings.
    5. Configure DNS: To make our website accessible through a domain name (e.g., www.example.com), we need to configure DNS settings. We can either create a new Route 53 hosted zone or use a third-party DNS provider to point our domain name to the S3 bucket's website endpoint.
    6. Testing and troubleshooting: Once the DNS settings propagate, we can test our website by accessing it through the domain name. If we encounter any issues, check our bucket permissions, file names, and DNS configurations.

- The static website hosting in Amazon S3 is suitable for websites that don't require dynamic server-side processing or database interactions. If our website needs server-side functionality, we may need to consider other AWS services like Amazon EC2 or AWS Lambda.

BODY:

- Step 1: Create a bucket
    1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
    2. Choose Create bucket.
    3. Enter the Bucket name as (myfirststaticwebsite1).
    4. Choose the Region as (EU (Stockholm) eu-north-1).
    5. Then I chose a Region that is geographically close to me to minimize latency and costs, or to address regulatory requirements. The Region that I chose determines my Amazon S3 website endpoint.
    6. Then accept the default settings and create the bucket, choose Create.

- Step 2: Enable static website hosting
    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, I chose the name of my bucket (myfirststaticwebsite1) that I want to enable static website hosting for.
    3. Then click Properties.
    4. Under Static website hosting, click Edit.
    5. Choose Use this bucket to host a website.
    6. Under Static website hosting, click Enable.
    7. In Index document, enter the file name of the index document , typically Html.html.
    8. The index document name is case sensitive and must exactly match the file name of the HTML index document that we plan to upload to our S3 bucket. When we configure a bucket for website hosting, we must specify an index document. Amazon S3 returns this index document when requests are made to the root domain or any of the subfolders.
    9. To provide our own custom error document for 4XX class errors, in Error document, enter the custom error document file name.
    10. The error document name is case sensitive and must exactly match the file name of the HTML error document that we plan to upload to your S3 bucket. If we don't specify a

    custom error document and an error occurs, Amazon S3 returns a default HTML error document.
    11. Choose Save changes.
    12. Amazon S3 enables static website hosting for our bucket. At the bottom of the page, under Static website hosting, we can see the website endpoint for our bucket.
    13. Under Static website hosting, note the Endpoint.
    14. The Endpoint is the Amazon S3 website endpoint for our bucket. After we finish configuring our bucket as a static website, we can use this endpoint to test our website.

- Step 3: Edit Block Public Access settings
    1. Choose the name of the bucket (myfirststaticwebsite1) that we have configured as a static website.
    2. Choose Permissions.
    3. Under Block public access (bucket settings), choose Edit.
    4. Clear Block all public access, and choose Save changes.

- Step 4: Add a bucket policy that makes our bucket content publicly available 1. Under Buckets, choose the name of our bucket (myfirststaticwebsite1).
    2. Choose Permissions.
    3. Under Bucket Policy, choose Edit.
    4. To grant public read access for our website, we needs to write JSON script for enabling policies.
    5. Update the Resource to our bucket name.
    6. Choose Save changes

- Step 5: Configure an index document
    1. When we enable static website hosting for our bucket (myfirststaticwebsite1), we enter the name of the index document (Html.html). After we enable static website hosting for the bucket, we upload an HTML file with this index document name to our bucket (myfirststaticwebsite1).
    2. Create an Html.html file.
    3. Save the index file locally.
    4. The index document file name must exactly match the index document name that we enter in the Static website hosting dialog box. The index document name is case sensitive.
    5. In the Buckets list, choose the name of the bucket (myfirststaticwebsite1) that we want to use to host a static website.
    6. Enable static website hosting for our bucket, and enter the exact name of our index document (Html.html).

- Step 6: Configure an error document
    1. Create an error document, for example 404.html.
    2. Save the error document file locally.
    3. The error document name is case sensitive and must exactly match the name that we enter when we enable static website hosting
    4. In the Buckets list, choose the name of the bucket (myfirststaticwebsite1) that we want to use to host a static website.
    5. Enable static website hosting for our bucket, and enter the exact name of our error.
    6. After enabling static website hosting, proceed to next step.

- Step 7: Test the website's endpoint
    1. Under Buckets, choose the name of our bucket (myfirststaticwebsite1).
    2. Choose Properties.
    3. At the bottom of the page, under Static website hosting, choose our Bucket website endpoint (http://myfirststaticwebsite1.s3-website.eu-north-1.amazonaws.com).
    4. Our index document opens in a separate browser window.

- Step 8: Clean up
    1. We had created our static website only as a analyzing exercise, then delete the AWS resources that we allocated so that we no longer accrue charges. After we delete our AWS resources, our website is no longer available.

CONCLUSION:

- Hence, finally we have created and hosted the portfolio static website in Amazon S3 (http://myfirststaticwebsite1.s3-website.eu-north-1.amazonaws.com). The task was accomplished successfully.
- After hosting a static website in Amazon S3, we can conclude the following:
    1. Reliable and scalable hosting: Amazon S3 provides a reliable and scalable hosting solution for static websites. It can handle high traffic volumes and ensures the availability of our website.
    2. Cost-effective: Hosting a static website in Amazon S3 is cost-effective, especially for low to moderate traffic websites. With its pay-as-you-go pricing model, you only pay for the storage and data transfer we use.
    3. Simple setup and management: Setting up and managing a static website in Amazon S3 is straightforward. We can easily upload and organize our website's files using the AWS Management Console or through APIs.
    4. Global content delivery: By integrating Amazon S3 with Amazon CloudFront, we can distribute our website's content globally and reduce latency. This improves the performance and user experience of our website for visitors from different locations.
    5. Security and compliance: Amazon S3 offers robust security features, including access controls, encryption, and compliance certifications. We can ensure the privacy and integrity of our website's data, making it suitable for various security requirements.
    6. Versioning and backup: With Amazon S3's versioning capability, We can store multiple versions of our website's files. This provides a backup mechanism and allows us to track changes, revert to previous versions, and recover from accidental modifications.
    7. Integration with other AWS services: Hosting our website on Amazon S3 allows us to leverage other AWS services for additional functionality. We can easily integrate with services like AWS Lambda, Amazon API Gateway, or Amazon DynamoDB to enhance our website's capabilities.

REFERENCE:

- I had referred the official website of AWS in order to accomplish this task. The links I had referred are disclosed below;
  https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

  https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

  https://docs.aws.amazon.com/AmazonS3/latest/userguide/creating-bucket.html

  https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html