Hosting a Static File Portfolio on IBM Cloud

**(Elective Module: 1)**

Submitted in partial fulfillment of completion of the course

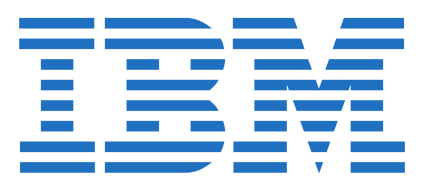
Advanced Diploma in IT, Networking and Cloud Computing

Submitted by:

**Aashika Raj**

Under Guidance of:

**Arpita Roy (Edu net)**

Year 2024

**PROJECT OVERVIEW**

Acknowledgement

Requirements

Technology Stack

Hardware and Software

Objective

Key Features

Setup and Configuration

Benefits

Use Cases

Considerations

Uploading Portfolio to IBM Cloud Storage

Additional Tips

Additional Resource

Different pricing models offered by cloud providers

Terms or components

Conclusion

Links to projects on GitHub

**Acknowledgement**

I would like to express my sincere gratitude to all those who have supported in my project and I’m truly thankful to supervisor Arpita Roy for her insights and feedback and their guidance, expertise, and unwavering support and direction of the project. Their expertise and commitment significantly enriched the quality of this project.

**Requirements**

* Windows
* IBM Cloud

**Technology Stack**

* HTML, CSS and JavaScript
* IBM Cloud Object Storage

**Hardware and Software**

Laptop/Desktop

OS

Html CSS and JavaScript

Web Browser

IBM cloud account

Inernet

VS code editor

GitHub

**Objective**:

This document provides a comprehensive overview of cloud object storage, its key features, benefits, use cases, and considerations. It is designed to guide developers, IT professionals, and anyone interested in learning about this powerful data storage solution.

What is Cloud Object Storage?

Cloud object storage is a data storage architecture designed for massive scalability and flexibility. It stores data in individual units called objects, each identified by a unique ID and associated with descriptive metadata. Unlike traditional file systems, objects are not organized in a hierarchical structure, allowing for efficient access and retrieval from anywhere with an internet connection.

IBM Cloud Object Storage (COS) offers the capability to host static websites directly from buckets, eliminating the need for separate web servers. This feature provides a cost-effective, scalable, and highly available solution for deploying static websites and web applications

**Key Features**:

Scalability: Object storage can seamlessly scale to handle petabytes or even extra bytes of data without limitations.

Durability: Data is replicated across multiple servers and locations, ensuring high availability and fault tolerance.

Cost-effectiveness: Object storage often offers lower costs than traditional storage solutions, especially for infrequently accessed data.

Accessibility: Objects are accessible from anywhere through simple RESTful APIs and web interfaces.

Security: Advanced encryption and access control mechanisms ensure data privacy and integrity.

Flexibility: Object storage can store a wide variety of data, including unstructured data like images, videos, and emails.

Simple Configuration: Easily configure buckets for static website hosting using the IBM Cloud console, CLI, or API.

Public Access: Websites are accessible to anyone with the URL, without authentication.

Index and Error Documents: Specify default files for the home page and error handling.

Redirect and Routing Rules: Control traffic flow and page redirections.

Custom Domain Support: Use your own domain names with IBM Cloud Internet Services (CIS).

Secure Sockets Layer (SSL): Enable HTTPS for secure communication using CIS.

Globally Distributed Content Delivery Network (CDN): Optional CDN integration for faster content delivery and improved performance.

**Setup and Configuration**

1. Create a COS Bucket:

- Use the IBM Cloud console, CLI, or API to create a new bucket.

2. Enable Static Website Hosting:

- In the bucket configuration, enable the "Static website hosting" option.

3. Specify Index and Error Documents:

- Set the names of your index and error documents (e.g., index.html, error.html).

4. Configure Routing Rules (Optional):

- Create routing rules to redirect traffic or handle specific requests.

5. Upload Website Content:

- Upload your static website files (HTML, CSS, JavaScript, images, etc.) to the bucket.

6. Set up Custom Domain (Optional):

- Use CIS to configure a custom domain name for your website.

7. Enable SSL (Optional):

- Use CIS to enable HTTPS for secure communication.

**Benefits:**

Reduced infrastructure management: Cloud providers manage the underlying hardware and software, minimizing operational burden.

Improved disaster recovery: Replicated data across multiple locations ensures availability even during outages.

Simplified backups and archiving: Large data volumes can be easily archived and retrieved when needed.

Enhanced data insights: Object storage integrates with big data analytics tools for efficient data processing and analysis.

**Use Cases:**

Media and entertainment: Storing large volumes of multimedia content like video files and music libraries.

Healthcare: Archiving medical records and images for long-term access and compliance.

Big data analytics: Storing and analyzing massive datasets for business intelligence and scientific research.

Backup and disaster recovery: Protecting critical data against accidental loss or hardware failures.

IoT and sensor data: Storing and processing data generated by connected devices and sensors.

**Considerations:**

Egress charges: Some cloud providers charge additional fees for downloading data beyond a certain amount.

Access latency: Object access may have slightly higher latency compared to local storage options for frequently accessed data.

Data lifecycle management: Implementing policies for data retention and deletion is crucial for cost optimization.

Choosing the right provider: Evaluate features, pricing models, and security compliance to meet your specific needs.

**Workflow:**

The high-level workflow of the Static Website Hosting feature is as follows:

* An Operator enables the Static Website Hosting feature on the Vault.
* An Operator configures the Static Website Virtual Host Suffix in the Access Pools, which are deployed to the Vaults which have the feature enabled.
* An Operator configures their DNS servers to perform proper routing of static website virtual host style addresses to the appropriate Accessor node or Load Balancer IP address.
* A User adds the Website Configuration Policy to any or all buckets that have the Static Website Hosting feature enabled.
* A User accesses the objects in the COS buckets with Static Website Hosting via READ requests (HEAD/ GET) of the Objects by using virtual host-based addressing requests sent to the Static Website Virtual Host Suffix. – For example, if the Static Website Virtual Host Suffix is static-website.example.com, then to access the static website for the COS bucket named ‘bucket name’,
* use https://cloud-object-storage-cos-static-web-hosting-mq1.s3.jp-tok.cloud-object-storage.appdomain.cloud/index.html

**Uploading Portfolio to Cloud Object Storage**

A Documentation Guide Storing your portfolio in cloud object storage like Google Cloud Storage, Amazon S3, or Azure Blob Storage offers several benefits: scalability, security, and accessibility from anywhere. This guide will walk you through the process of uploading your portfolio to a cloud object storage platform, ensuring its safe and efficient preservation.

1. Choose Your Platform:

Each cloud object storage platform has its own features and pricing plans.

Consider factors like:

Cost: Compare storage fees and data transfer charges before choosing a platform.

Accessibility: Ensure your chosen platform offers easy access through web interfaces or APIs.

Integrations: Check if the platform integrates with other tools you might use, like portfolio builders or content management systems.

2. Prepare Your Portfolio Files:

Organize your files: Create folders for different sections of your portfolio (e.g., projects, skills, testimonials).

Optimize file sizes: Reduce image and video file sizes for efficient uploads and downloads.

Backup your files: Always maintain a local backup of your portfolio in case of unexpected issues.

3. Create a Bucket:

A bucket is a container for your files in cloud object storage. Choose a descriptive name that reflects your portfolio (e.g., "your-name-portfolio").

4. Upload Your Files:

Most platforms offer drag-and-drop upload functionality or allow bulk uploading via folders. You can also use command-line tools or APIs for programmatic uploads.

5. Configure Public Access (Optional):

If you want your portfolio to be publicly accessible, enable public read access for your bucket. For private portfolios, keep access restricted.

6. Set a Custom Domain (Optional):

Consider linking your portfolio to a custom domain name for a more professional appearance. Most platforms offer domain name registration and configuration options.

7. Share Your Portfolio:

Once uploaded, you can share your portfolio by providing the URL or integrating it into your website or online platforms like LinkedIn.

8. Security Considerations:

Enable encryption: Choose server-side encryption to protect your files at rest.

Use strong passwords: Set strong passwords for your cloud storage account and grant access cautiously.

Monitor access logs: Regularly check access logs to detect any suspicious activity.

**Additional Tips**:

Version control: Implement version control to track changes and revert to previous versions if needed.

Metadata: Add descriptive metadata to your files for easier searching and organization.

Lifecycle management: Set up lifecycle policies to automatically archive or delete older files to optimize storage costs.

By following these steps and best practices, you can securely and efficiently upload your portfolio to cloud object storage, ensuring its accessibility and longevity. Remember to tailor the process to your specific platform and needs for a seamless and successful experiences.

**Additional Resources:**

AWS Object Storage

Google Cloud Storage

Microsoft Azure Blob Storage

**Different pricing models offered by cloud providers**.

Advanced features like data lifecycle management and security options.

Best practices for implementing and managing cloud object storage.

Case studies showcasing successful deployments of cloud object storage.

**Terms or components**

Website Configuration Policy

* A Policy which is added to a COS bucket to configure the bucket as a website.
* Policy could include configuring the index and error objects and optionally could include a redirect all (or) a collection of granular redirect rules.
* Static Website a static website consists of HTML, JavaScript, images, video and other files that do not require any server-side application processing.
* Static websites are typically used in cases where the website requires minimal to no server administration, and where the website has few authors and requires infrequent updates, and websites which need to automatically scale for intermittent increase in traffic.
* Static Website Virtual Host Suffix A specifically configured virtual host suffix that is used to access COS buckets with the Static Website Hosting feature.
* If the Static Website Virtual Host Suffix is static-website.example.com, then to access the COS bucket named ‘bucket name’, use <http://bucketname.static-website.example.com/>.

**Conclusion:**

Cloud object storage offers a powerful and cost-effective solution for storing and managing large volumes of unstructured data. With its scalability, durability, and flexibility, it is a compelling choice for a wide range of use cases across various industries.

**Links to projects on GitHub**

https://aashikaraj.github.io/website\_port/