Personal Blog on IBM Cloud Static Web Apps

BY:

SAKTHIDHASAN. K

PHASE 1:DOCUMENTATION SUBMISSION

Project Definition:

The project is to create a personal travel blog hosted on IBM Cloud Static Web Apps. The goal is to share travel adventures, tips, and captivating photos to inspire others to explore the world and create unforgettable memories. This involves designing the blog structure, creating engaging content, setting up the IBM Cloud Static Web Apps, and ensuring ease of updating the blog.

Abstract:

Our Project is a captivating personal travel blog hosted on IBM Cloud Static Web Apps. This blog is an immersive window into the adventures, insights, and captivating moments experienced by an avid explorer. From the pristine beaches of Bali to the bustling streets of Tokyo, every blog post is a narrative that takes readers on a visual journey through picturesque destinations and shares invaluable travel tips.

The blog's IBM Cloud Static Web Apps hosting ensures seamless updates, making it easy for the author to chronicle their ongoing adventures. With engaging content, stunning photography, and insightful travel guides, Our Project inspires readers to embark on their own journeys and discover the beauty of the world.

Explore the world through the eyes of a passionate traveler, and let our project be your guide to forging unforgettable memories and igniting your own wanderlust.

Hosting Static Websites on IBM Cloud:

This blog post explains how to host a static website on IBM Cloud. These websites are rendered client-side by the browser from static assets, like HTML, CSS and JS files. They do not need a server-side component to create pages dynamically at runtime. Static websites are often combined with backend APIs to create Single Page Applications.

Hosting static websites on IBM Cloud uses Cloud Object Storage (COS) and Cloud Internet Services (CIS) (with Page Rules and Edge Functions).

Serving static assets:

IBM Cloud Object Storage is a scalable storage solution for cloud applications. Files are managed through a RESTful HTTP API and stored in user-defined collections called "buckets". Bucket files are returned as HTTP responses from HTTP GET requests.

COS supports an optional "anonymous read-only access" setting for buckets. This means all files in the bucket will be accessible using anonymous HTTP GET requests.

Putting HTML, CSS and JS files in a public bucket allows static websites to be served directly by COS. Users are charged for bandwidth used and HTTP requests received for all bucket files.

Steps to setup our personal blogs:

Sign up for IBM Cloud:

Signing up with IBM cloud plays a crucial step in create our blog and IBM helps us in hosting our blog as a static Website.

Create a Static Website:

Develop your travel blog website using HTML, CSS, and JavaScript. You can use popular static site generators like Jekyll, Hugo, or Gatsby to make the development process easier.

Host Your Website on GitHub:

Host your website's source code on GitHub. GitHub is often used for version control and hosting static websites. Make sure your code is in a public GitHub repository.

Set Up IBM Cloud Static Web Apps:

In the IBM Cloud Dashboard, navigate to the "Static Web Apps" service.

Create a new Static Web App and connect it to your GitHub repository. This will enable automatic deployments whenever you update your blog.

Design Your Blog:

Design your travel blog with an appealing layout, fonts, and color schemes. Ensure it's responsive for various screen sizes.

Add Content:

Start adding your travel content, including blog posts, photos, and videos. Use HTML or markdown for your blog posts and include captivating images to illustrate your adventures.

Enable Comments:

Consider integrating a commenting system like Disqus or Facebook Comments to engage with your readers.

SEO and Analytics:

Optimize your blog for search engines (SEO) and integrate analytics tools like Google Analytics to track visitor statistics.

Social Media Integration:

Add social media sharing buttons to encourage readers to share your blog posts on their social networks.

Regular Updates:

Update your blog with new travel experiences, tips, and photos regularly to keep your audience engaged.

Promote Your Blog:

Share your blog posts on social media platforms, travel forums, and with friends and family to increase your blog's visibility.

Key Objectives for Personal Blog on IBM Cloud Static Web Apps:

Content Quality and Consistency:

Objective: Create high-quality, engaging, and informative blog posts.

Key Results: Publish a minimum number of blog posts per month, maintain a consistent posting schedule, and gather positive reader feedback.

User Experience:

Objective: Ensure a seamless and enjoyable user experience for visitors.

Key Results: Optimize website speed and performance, implement responsive design for various devices, and gather user feedback to make improvements.

SEO and Traffic Growth:

Objective: Improve search engine rankings and increase website traffic.

Key Results: Optimize blog posts for SEO, monitor keyword rankings, track organic traffic growth, and implement a content marketing strategy.

Engagement and Interactivity:

Objective: Encourage user engagement and interaction on your blog.

Key Results: Implement features like comments, social sharing buttons, and interactive elements in your blog posts, and track user engagement metrics.

Branding and Design:

Objective: Create a strong brand identity and appealing design.

Key Results: Maintain a consistent brand across your blog, use a visually appealing layout, and update your blog's design as needed.

Security and Privacy:

Objective: Ensure the security and privacy of your blog and user data.

Key Results: Regularly update plugins and themes, use HTTPS, implement security best practices, and comply with privacy regulations like GDPR.

Analytics and Insights:

Objective: Gather data to make informed decisions and measure your blog's success.

Key Results: Set up analytics tools (e.g., Google Analytics), track user behavior, monitor conversion rates, and use data to refine your content and strategy.

Community Building:

Objective: Build a community around your blog.

Key Results: Engage with your readers through comments and social media, respond to feedback, and foster a loyal readership.

Backup and Recovery:

Objective: Ensure the safety of your blog's data.

Key Results: Implement regular backups of your website and have a recovery plan in place in case of data loss or site issues.

Accessibility and Inclusivity:

Objective: Make your blog accessible to all users.

Key Results: Ensure compliance with web accessibility standards (e.g., WCAG), and regularly test your blog for accessibility improvements.

Create IBM Cloud Object Storage instance:

If you already have an instance of Cloud Object Storage you can skip this step…

=>Provision a new instance of IBM Cloud Object Storage

Create IBM Cloud Object Storage Bucket:

=>Open the COS instance from the Resource List.

=>Create a new COS bucket to host the static site files.

   =>Choose a Bucket name

=>Choose the Resiliency, Location and Storage Class options for the bucket.

Create IBM Cloud Object Storage Bucket

=>Open the COS instance from the Resource List.

=>Create a new COS bucket to host the static site files.

>=>Choose a Bucket name

>=>Choose the Resiliency, Location and Storage Class options for the bucket.

Upload Static Assets To Bucket

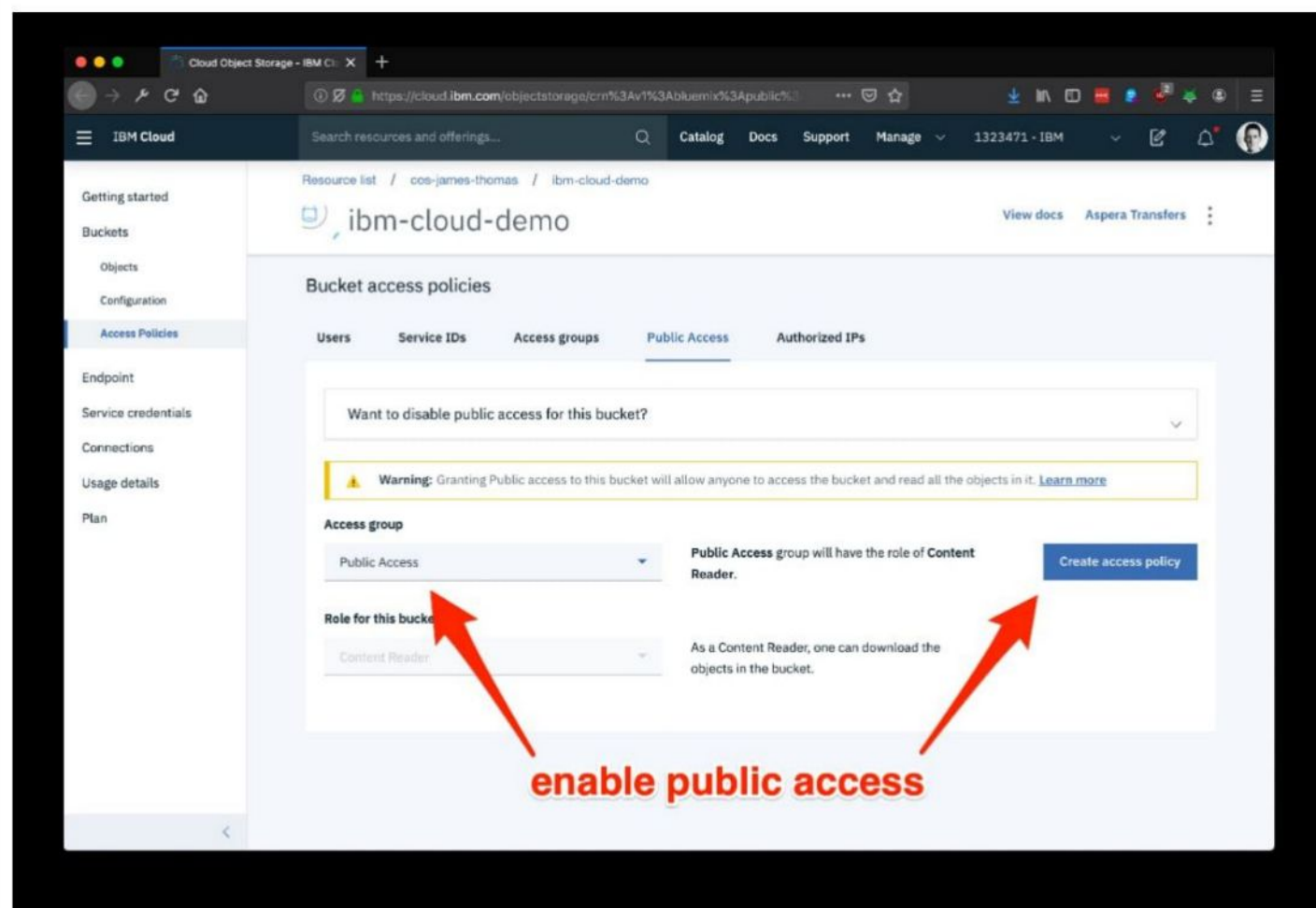=>Upload static file assets to the new bucket.

Enable Public Access to bucket files

=>Click the "Access Policies" menu item from the bucket level menu.

=>Click the "Public Access" tab from the bucket access policy page.

=>Check the Access Group drop-down has "Public Access" option selected.

=>Click the "Create access policy" and then "Enable" on the pop menu.

Check bucket files are accessible:

Bucket files should now be accessible using the service endpoint URL, bucket id and file names. COS supports providing the bucket name in the URL path or a sub-domain on the service endpoint.

=>Open the "Configuration" panel on the bucket page.

Retrieve the public endpoint shown, e.g. s3.<REGION>.cloud-object-storage.appdomain.cloud

vhost addressing:

=> <BUCKET_NAME>.s3.eu-gb.cloud-object-storage.appdomain.cloud/index.html

url path addressing

=>s3.<REGION>.cloud-object-storage.appdomain.cloud/<BUCKET_NANME>/index.html

Configurable Index and Error pages:

=>COS will now serve static assets from a custom sub-domain, where file names are explicitly included in the URL, e.g. http(s)://<CUSTOM_DOMAIN>/index.html. This works fine for static websites with two exceptions, the default document for the web site and the error page.

=>When a user visits the COS bucket sub-domain without an explicit file path (http(s)://<CUSTOM_DOMAIN>), the COS service will return the bucket file list, rather than the site index page. Additionally, if a user requests a missing file, COS returns an XML error message rather than a custom error page.

=>Both issues can be resolved using Edge Functions, a new feature in Cloud Internet Services.

Configure Page Rules and DNS records:

Cloud Internet Services can automatically set up Page Rules and DNS records needed to forward custom domain traffic to COS buckets. This automatically exposes the bucket as `bucket-name.your-domain.com`. If you want to change this default sub-domain name, follow the manual steps in the next section.

- Click the Performance drop-down menu and click the *"Page Rules"* link.
- Click the *"Create rule"* button from the table.
- Select the Rule Behaviour Setting as *"Resolve Override with COS"*
- Select the correct COS instance and bucket.
- Click the *"Create"* button.

Getting started
Account ⌄
Overview
Metrics ⌄
Security ⌄
Reliability ⌄
Performance ⌃
  Caching
  Page Rules
  Routing
Edge Functions
Plan

Resource list /

● cis-james-thomas

Resource group: default   Location: Global   [james-thomas] ✎

Page Rule

URL match   **Enabled**

.ibm-cloud-demo.xyz/   Off —●— On

New page rules are placed at lowest priority.

Page Rule Setting

Rule Behavior

CNAME Sub-Domain

Setting   **Set as**

Resolve Override with COS ▼   ibm-cloud-demo.ibm-cloud-demo.xyz   ⊗

Cloud Object Storage Instance

cos-james-thomas ▼

Bucket

ibm-cloud-demo ▼

Host Header Override

ibm-cloud-demo.s3.eu-gb.cloud-object-storage.appdomain.cloud

A CNAME record will be created automatically for ibm-cloud-demo.s3.eu-gb.cloud-object-storage.appdomain.cloud as ibm-cloud-demo.

Add setting ⊕

COS Bucket Instance

[Cancel]   [**Create**]

Summary

Static web sites can be hosted on IBM Cloud using Cloud Object Storage and Cloud Internet Services.

Cloud Object stores page files needed to render the static website. Anonymous bucket file access means files are accessible as public HTTP endpoints, without having to run infrastructure to serve the assets.

Cloud Internet Services forwards HTTP traffic from a custom domain to the bucket hostname. DNS CNAME records are used to resolve the sub-domain as the custom bucket hostname. Page Rules override HTTP request headers to make this work. Edge Functions are used to implement configurable Index and Error documents, by dynamically modifying in-flight requests with custom JavaScript.

Hosting static web sites using this method can be much cheaper (and easier) than traditional infrastructure. Developers only get charged for actual site usage, based on bandwidth and HTTP requests.