Project 1: Personal Blog on IBM Cloud Static Web Apps

Phase 4 : Development Part 2

Introduction:

In this phase, we have continued building our travel blog by leveraging IBM Cloud's Static Web App and incorporating a static site generator like Jekyll or Hugo. The process involves signing up for an IBM Cloud account, connecting our blog's repository, configuring a build pipeline, and deploying our website. By choosing a static site generator, we'll be able to transform our existing HTML content into template files that are easily manageable and updatable. This introduction outlines the steps we'll follow to make our travel blog more dynamic and user-friendly.

Setting up IBM account:

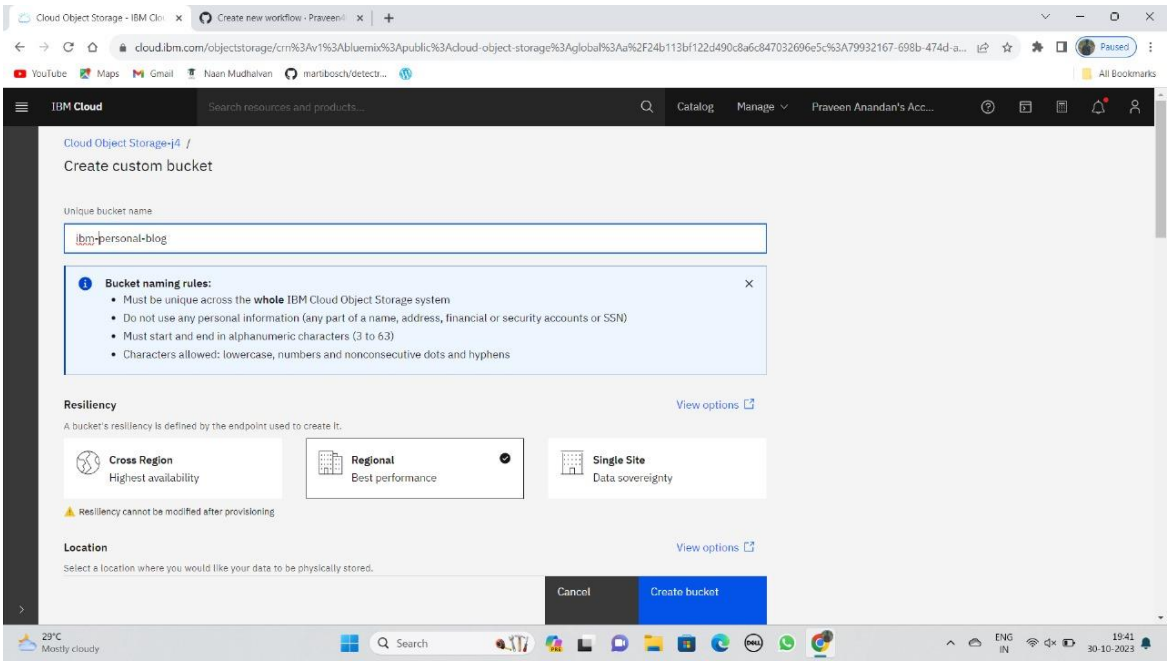
- The process involved visiting the IBM Cloud website, clicking 'Sign Up,' providing personal information, verifying the email address through a confirmation link, accepting the terms and conditions.
- By using feature code given by the IBM, we get an access to the IBM Cloud dashboard. With the account now established, the next steps can be taken to manage resources, including the Static Web App for the travel blog.

Create an IBM Cloud Object Storage Instance:

- Log in to our IBM Cloud account and access the IBM Cloud Dashboard, where we manage cloud services. Click "Create Resource," then search for "Object Storage" in the catalog.
- Now, configure our Object Storage by providing details like the service name, tags, and the associated resource group.
- After customizing these settings, click "Create" to officially create our Object Storage service. This initiates the provisioning of our Object Storage instance in the IBM Cloud environment.

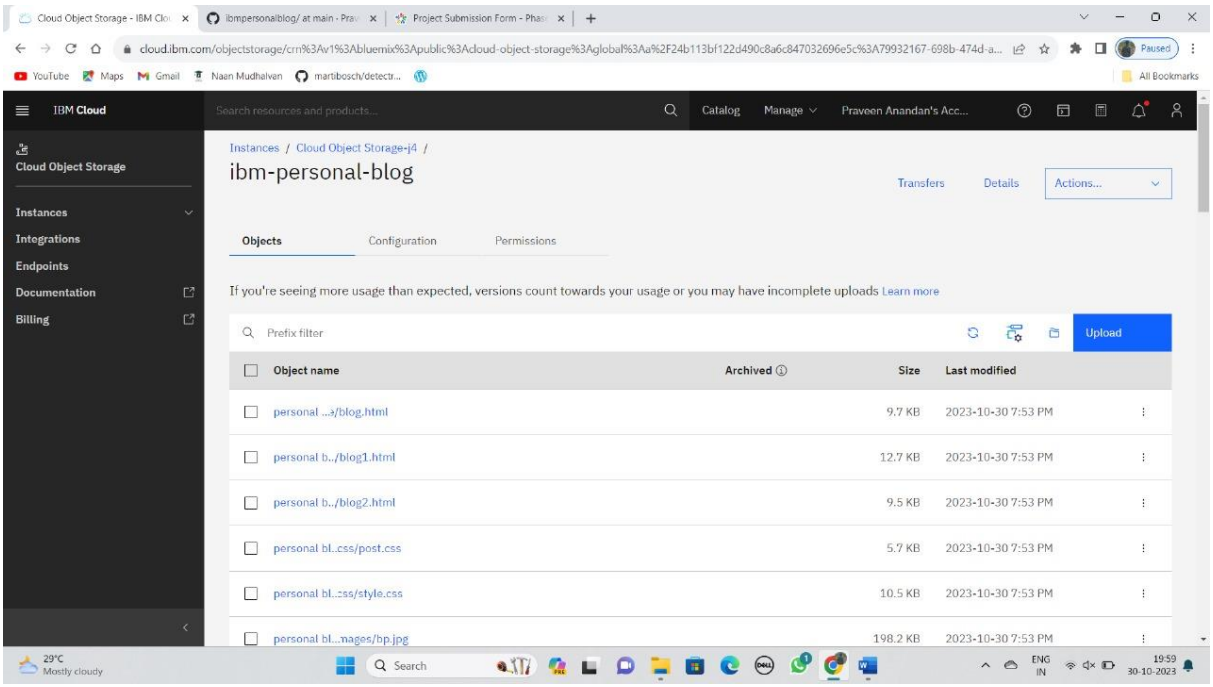
Create a New Bucket:

- Within the Object Storage instance we've created, we establish a new bucket. This bucket serves as a container for organizing and storing our static web content.
- This bucket acts as a virtual folder, enabling us to categorize our web content, such as web pages, images, stylesheets, and more, in an orderly manner. It's like having separate drawers in a filing cabinet, each containing specific types of documents for easy retrieval.
- By creating this bucket, we establish a structured framework for our web content, making it more manageable and accessible.



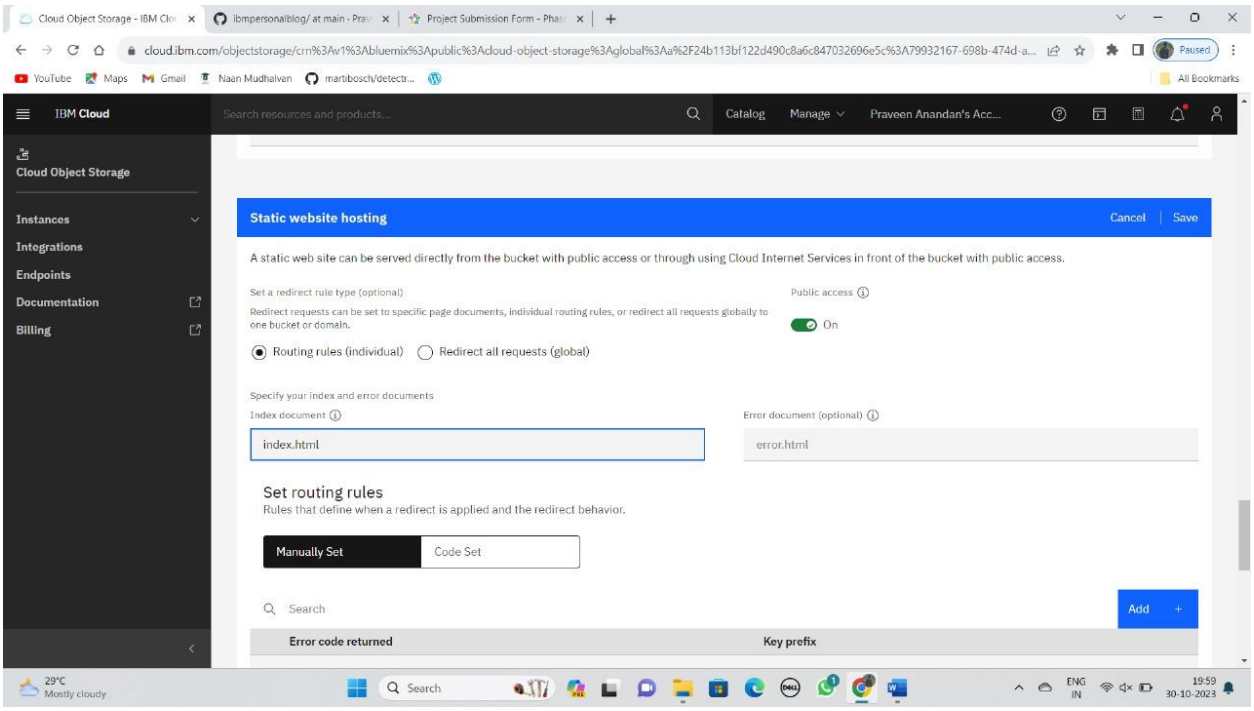
Upload Our Static Web Content:

- Then I uploaded our static web content files to the newly created bucket. This process ensures that our web content is securely stored within our Object Storage instance, ready to be served to our users.
- We've uploaded our static web content files, including 'index.html,' to the newly created bucket. These files are securely stored in our Object Storage instance, ready to be served to our website's visitors.



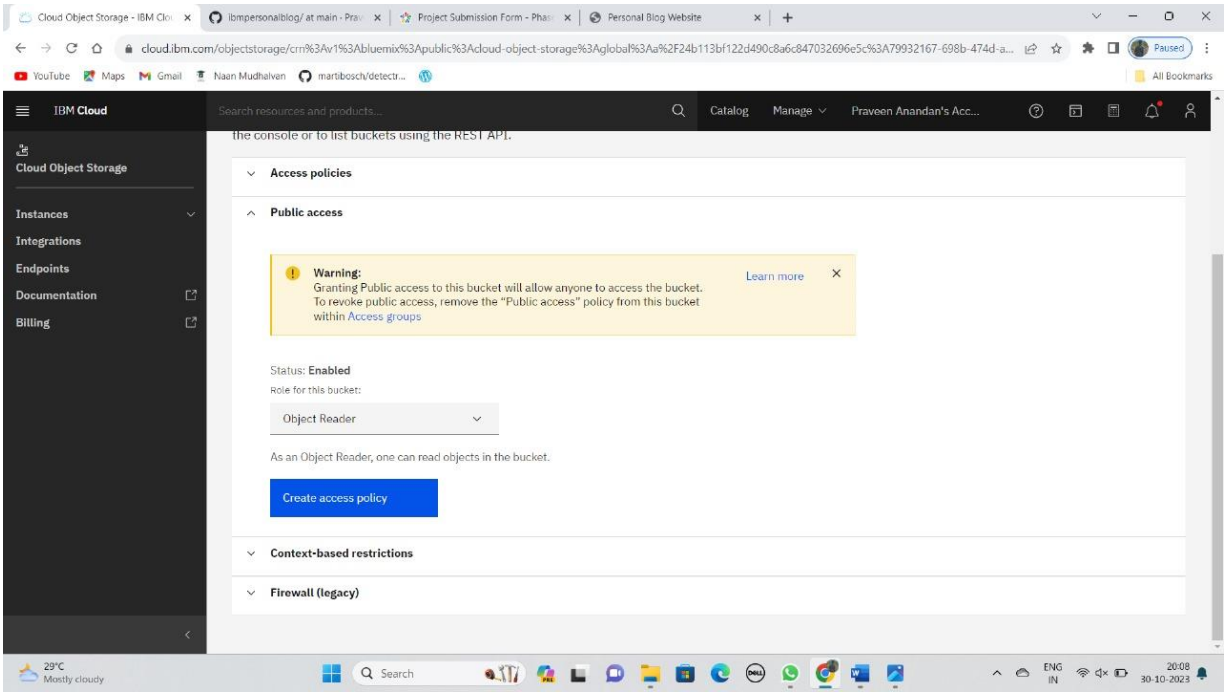
Create a static web app:

- A static web app is a type of web application that consists of fixed, unchanging content. Unlike dynamic web apps, which generate content on the server in real-time.
- Static web apps serve pre-existing HTML, CSS, and JavaScript files directly to users. This means that the content of a static web app doesn't change based on user interactions or data from a database.
- Once I turned on the static web hosting, I have given the public access.
- Then I mentioned the index file name and error file name.



Enabling Public Access:

- We've configured our Object Storage bucket to have public access, which means that anyone with the correct URLs can access the web content we've stored in this bucket.
- This public access setting is particularly useful for making our website and its content openly available to the internet without requiring authentication or special permissions.
- Visitors can simply enter the specific URLs associated with our web content to view our website's pages, images, or other resources.
- This accessibility ensures that our website is easily reachable and navigable by users from around the world, fostering a seamless and user-friendly experience.



Sample output :

- Here is the sample output after we deployed our website