# CLOUD APPLICATION DEVELOPMENT

# GROUP-3

# Project 1: Personal Blog on IBM Cloud Static Web Apps

# Phase4: Development Part-2

## IBM CLOUD STATIC WEB APP SETUP

1. **Sign in to IBM Cloud:**

   Sign in to the IBM Cloud console.

2. **Creating a New Static Web App:**

   - In the IBM Cloud dashboard, "Creating Resource" or "Creating App."
   - Search for "Static Web App" and select it.

3. **Setting Up the Static Web App:**

   - Choosing a region for app's deployment.
   - Providing a name and description for app.
   - Configuring the URL for app.

4. **Connecting a Repository:**

   - prompted to connect app to a repository where code is hosted (e.g., GitHub, GitLab, Bitbucket).

5. **Configuring the Build Pipeline:**

   - Set up a build pipeline to automatically build and deploy static web app when changes are pushed to the repository.

**CODE :-**

```
deploy:
needs: build
runs-on: ubuntu-latest
steps:
- name: Deploy to Netlify
uses: netlify/actions/cli@v2
with:
site_id: <YOUR_NETLIFY_SITE_ID>
deploy_key: <YOUR_NETLIFY_DEPLOY_KEY>
publish_branch: master
```

- Configuring the build settings, such as specifying the build command and output directory for app.

**6.Configuring Deployment Options:**

- Choosing deployment options. This may include selecting a target environment (e.g., IBM Cloud Object Storage) where your static files will be hosted.
- Configuring any environment variables or settings required for app.

```
name: Deploy Jekyll Site
 deploy:
  needs: build
  runs-on: ubuntu-latest
  steps:
   - name: Deploy to GitHub Pages
```

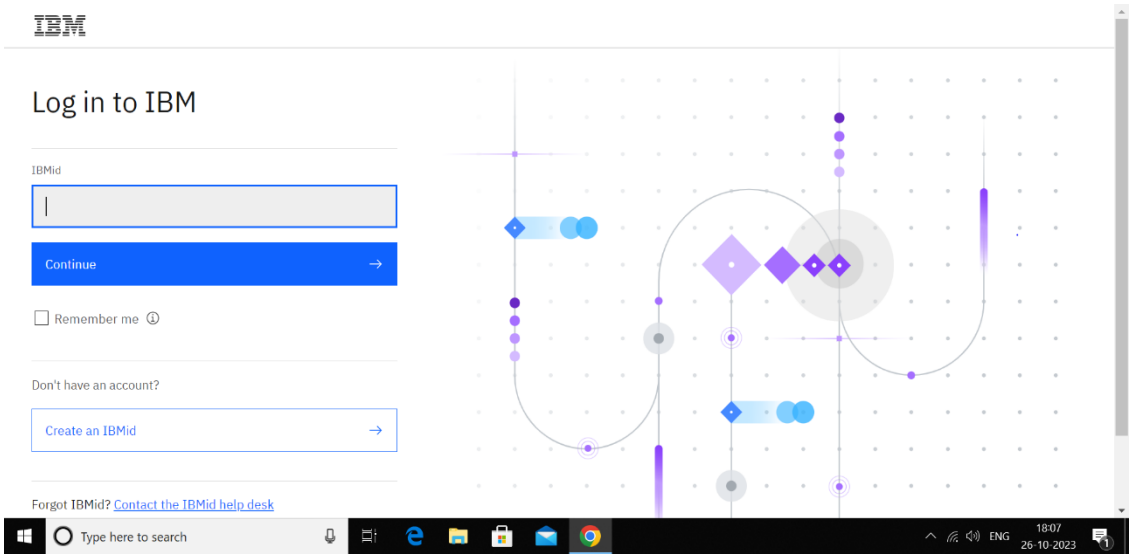
```
uses: actions/github-script@v5
with:
  script: |
    github.rest.actions.artifacts.upload({
      artifact_name: 'jekyll-site',
      repository: '${{ github.repository }}',
      branch: '${{ github.ref }}',
      name: 'jekyll-site',
```

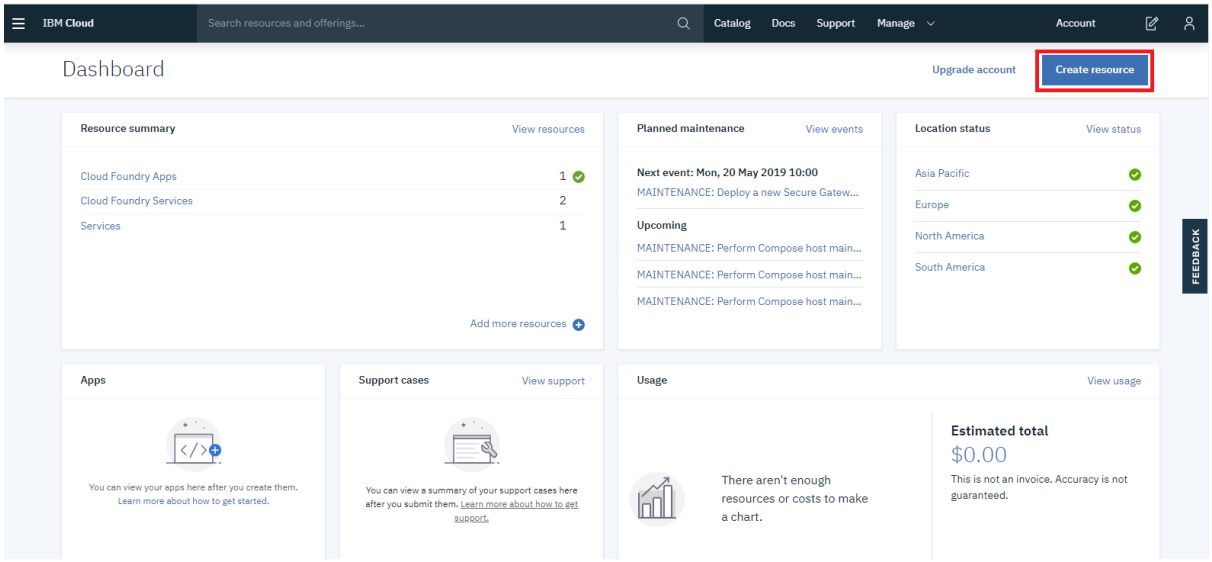
**7.Creating and Deploying:**

- Once confirmed, IBM Cloud will create static web app and set up the deployment pipeline.
- The app will be deployed, and will receive the app's URL

**8.Continuous Integration/Continuous Deployment (CI/CD):**

- With building pipeline in place, every time pushing changes in the repository, the pipeline will automatically build and deploy the app






## CONTENT MANAGEMENT

1. **Choosing a Static Site Generator:**

   Firstly, selectng a static site generator that suits. Jekyll and Hugo are popular choices, but there are others like Gatsby, Hexo, or Pelican.

2. **Installing the Static Site Generator:**

   Installing the chosen static site generator on local development environment.

3. **Creating a Directory Structure:**

   Organizing project directory with subfolders for content, layouts, assets, and configuration files. For travel blog having folders for different.

   countries or destinations.

4. **Content Creation:**

Writing travel blog posts in Markdown or a format supported by chosen generator. Including front matter in each post, which provides metadata such as title, date, and tags.

5. **Templates and Layouts:**

   Creating templates and layouts to define the structure and design of blog. Static site generators use templating engines like Liquid, Go HTML templates, or Handlebars. Customizing these templates to match blog's look and feel**.**

6. **Configuration:**

   Configuring static site generator by setting options and preferences in the configuration files. This might include specifying the destination folder for the generated site, defining URLs, and configuring plugins if generator supports them.

7. **Themes:**

   Choosing or creating a theme for travel blog. pre-designed themes for Jekyll and Hugo in their respective theme customize one that fits.

8. **Build the Site:**

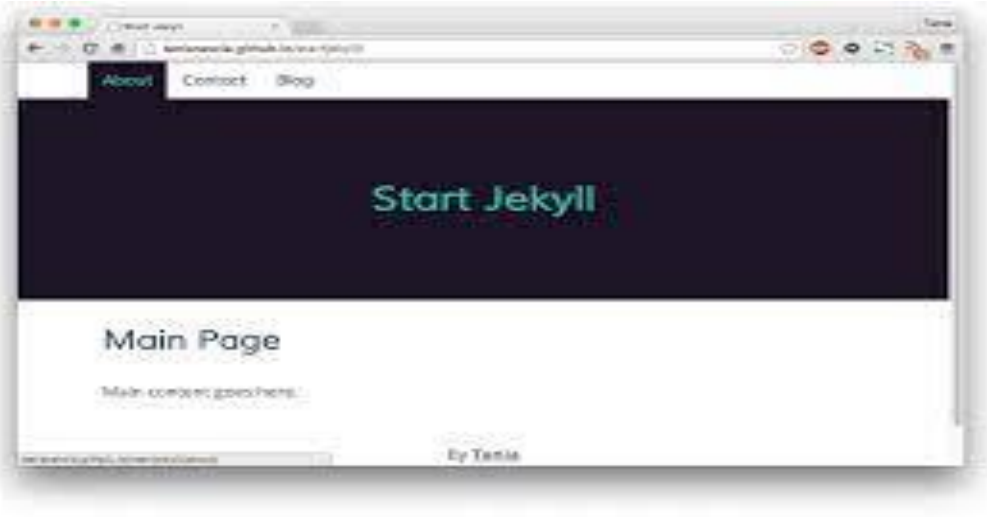
   Using the static site generator to build website from the source files. This typically involves running a command like hugo or jekyll build in terminal.

9. **Deployment:**

   Once the site is built deploy it to a web hosting platform or content delivery network (CDN). Popular options include Netlify, GitHub Pages. Ensuring that hosting supports static sites.

10. **Domain and SEO:**

Setting up a custom domain for your travel blog, and optimize it for search engines. Utilize SEO best practices in your content and meta tags.




CONCLUSION :-

Static web apps in the cloud offer a number of advantages over traditional dynamic web apps. They are typically faster, more scalable, more secure, and more cost-effective. Static web apps in the cloud are a good choice for a variety of websites and web applications.