**Personal Blog on IBM Cloud Static Wed app :**

**1. Sign up for an IBM Cloud account:**

If you don't already have an IBM Cloud account, you'll need to sign up for one. Visit the IBM Cloud website (https://cloud.ibm.com/), and follow the registration process.

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

Once you have an IBM Cloud account, follow these steps to create a new Static Web App:

- Log in to your IBM Cloud account.

- Go to the IBM Cloud Dashboard.

- Click on the "Create Resource" button.

- In the search bar, type "Static Web App" and select the Static Web App service.

- Follow the prompts to set up your Static Web App. This will involve choosing a region, giving your app a name, and setting up other configurations.

**3. Set Up the Repository:**

You'll need a repository that contains your travel blog code. If you're starting from scratch, you can create a new repository on a platform like GitHub, GitLab, or Bitbucket. If you already have a repository, ensure it contains your website files, including HTML, CSS, and any other assets.

**4. Build Pipeline:**

In your IBM Cloud Static Web App configuration, set up a build pipeline to build your website. You can choose a static site generator like Jekyll or Hugo for this purpose. Here's how to do it:

- In the build configuration, specify the repository where your code is located.

- Choose the build environment. For Jekyll or Hugo, you might need to specify the build command. For example, for Jekyll, you can use `jekyll build`.

**5. Deployment Options:**

Configure the deployment options in your IBM Cloud Static Web App:

- Set up the deployment branch. This is the branch that will trigger deployments when you push changes to it.

- Choose your deployment region.

- Define any environment variables or configuration settings that your website may require.

**6. Convert HTML Content to Template Files:**

Since you mentioned using a static site generator like Jekyll or Hugo, you'll need to convert your HTML content into template files. These generators use markdown files to create content, which makes it easier to manage and update your blog posts. Here's a basic guide for Jekyll:

- Create a `\_posts` folder in your repository.

- Convert each of your HTML blog posts into markdown files with the following naming convention: `YYYY-MM-DD-post-title.md`.

- Use markdown syntax for formatting your content, and include YAML front matter to specify metadata like the post title and date.

Here's an example of a Jekyll markdown post:

```markdown

---

layout: post

title: "My Travel Adventure"

date: 2023-10-25

---

Welcome to my travel blog post. This is where I share my latest adventure.

<!-- Your content goes here -->

```

**7. Push Code and Deploy:**

Once your setup is complete, push your code to the specified deployment branch. Your Static Web App will automatically build and deploy your website based on the changes you push.

That's a high-level overview of setting up a travel blog with IBM Cloud Static Web App and using a static site generator like Jekyll or Hugo to manage your content. Make sure to consult IBM Cloud's documentation and the documentation of your chosen static site generator for more detailed instructions on each step.

**8. Create a New Static Web App:**

- When creating a new Static Web App, choose a region that's geographically close to your target audience. This helps improve website performance for your visitors.

- Configure a custom domain if you have one. This allows you to use your own domain name (e.g., www.yourtravelblog.com) for a more professional look.

**9. Choose a Static Site Generator (Jekyll):**

- Install Jekyll: Ensure you have Ruby and RubyGems installed on your computer. You can check by running `ruby -v` and `gem -v` in your terminal. If you haven't installed them, you'll need to do so first.

- Create a New Jekyll Site: The `jekyll new your-blog-name` command creates a basic Jekyll site. You can then navigate to the blog's directory using `cd your-blog-name`.

- Install Dependencies: Run `bundle install` to install the required dependencies specified in the `Gemfile`. This step ensures you have all the necessary plugins and themes.

**10. Push Code to Repository:**

- Make sure your repository is set to "Public" or that you have the necessary access controls in place if it's private. The Static Web App build process requires access to your repository.

- You can create a `.gitignore` file in your Jekyll project to exclude build artifacts and temporary files. Here's an example `.gitignore` for Jekyll:

```

\_site/

.sass-cache/

.jekyll-cache/

.DS\_Store

Gemfile.lock

```

**11. Configure Build Pipeline and Deployment Options:**

- In the build environment configuration, ensure the Static Web App service can access your repository. You may need to set up access tokens or SSH keys as needed for authentication.

- The deployment branch can be set to the default branch (often "main" or "master"), or any branch you prefer. Make sure that your changes are pushed to this branch for automatic deployment.

- Consider setting up environment variables for sensitive data like API keys. These variables can be accessed within your Jekyll site's configuration files.

**12. Deploy Your Blog:**

- After pushing your changes to the deployment branch, monitor the build and deployment process in your Static Web App dashboard. You can check build logs for any errors or issues that may arise during the build process.

- Your blog should be accessible at the URL provided by IBM Cloud Static Web App once the deployment is complete. You can visit this URL to view your live travel blog.

Now that you've set up your travel blog, keep in mind that you can continuously update and manage your content by creating new Jekyll posts or pages, editing the configuration, and customizing the design to suit your style. Jekyll simplifies content management and makes it easier to maintain your travel blog over time.