EXP NO : HOSTING A STATIC WEBSITE ON GOOGLE APP ENGINE

DATE :

AIM:

To use GAE launcher to launch the web applications.

PROCEDURE:

You can use Google App Engine to host a static website. Static web pages can contain client-side technologies such as HTML, CSS, and JavaScript. Hosting your static site on App Engine can cost less than using a traditional hosting provider, as App Engine provides a free tier. Sites hosted on App Engine are hosted on the REGION_ID.r.appspot.com sub domain, such as [my-project-id].uc.r.appspot.com. After you deploy your site, you can map your own domain name to your App Engine-hosted website.

Before you begin

Before you can host your website on Google App Engine:

1. Create a new Cloud Console project or retrieve the project ID of an existing project to use: Go to the Project page

2. Install and then initialize the Google Cloud SDK: Download the SDK

3. Creating a website to host on Google App Engine Basic structure for the project

This guide uses the following structure for the project:

- app.yaml: Configure the settings of your App Engine application.
- Www/: Directory to store all of your static files, such as HTML, CSS, images, and JavaScript.
- css/: Directory to store style sheets
- style.css: Basic style sheet that formats the look and feel of your site.
- images/: Optional directory to store images.
- index.html: An HTML file that displays content for your website.
- js/: Optional directory to store JavaScript files.

Creating the app.yaml file

The app.yaml file is a configuration file that tells App Engine how to map URLs to your static files. In the following steps, you will add handlers that will load www/index.html when someone visits your website, and all static files will be stored in and called from the www directory.

1. Create the app.yaml file in your application's root directory:
2. Create a directory that has the same name as your project ID. You can find your project ID in the Console.
3. In directory that you just created, create a file named app.yaml.
4. Edit the app.yaml file and add the following code to the file:

```
runtime: python39
entrypoint: python main.py

handlers:
- url: /static
  static_dir: static

- url: /.*
  script: auto
```

Creating the index.html file

Create an HTML file that will be served when someone navigates to the root page of your website. Store this file in your **static** directory.

```
<!DOCTYPE html>
<html>
<head>
  <title>My Static Website</title>
  <style>
```

```
    h1{
      text-align: center;
      margin-top: 150px;
      animation: glow 1s ease-in-out infinite;
      font-family: arial;
      font-size: 100px;
    }
    @keyframes glow {
      0%{text-shadow: 0px 0px 10px rgb(255, 0, 0);}
      50%{text-shadow: 0px 0px 10px rgb(0, 255, 0);}
      100%{text-shadow: 0px 0px 10px rgb(0, 0, 255);}
    }
  </style>
</head>
<body>
  <h1>CLOUD COMPUTING</h1>
</body>
</html>
```

Deploying your application to App Engine

When you deploy your application files, your website will be uploaded to App Engine. To deploy your app, run the following command from within the root directory of your application where the app.yaml file is located

Include the --project flag to specify an alternate Cloud Console project ID to what you initialized as the default in the gcloud tool. Example: --project [YOUR_PROJECT_ID]
To learn more about deploying your app from the command line, see Deploying a Python 2 App.

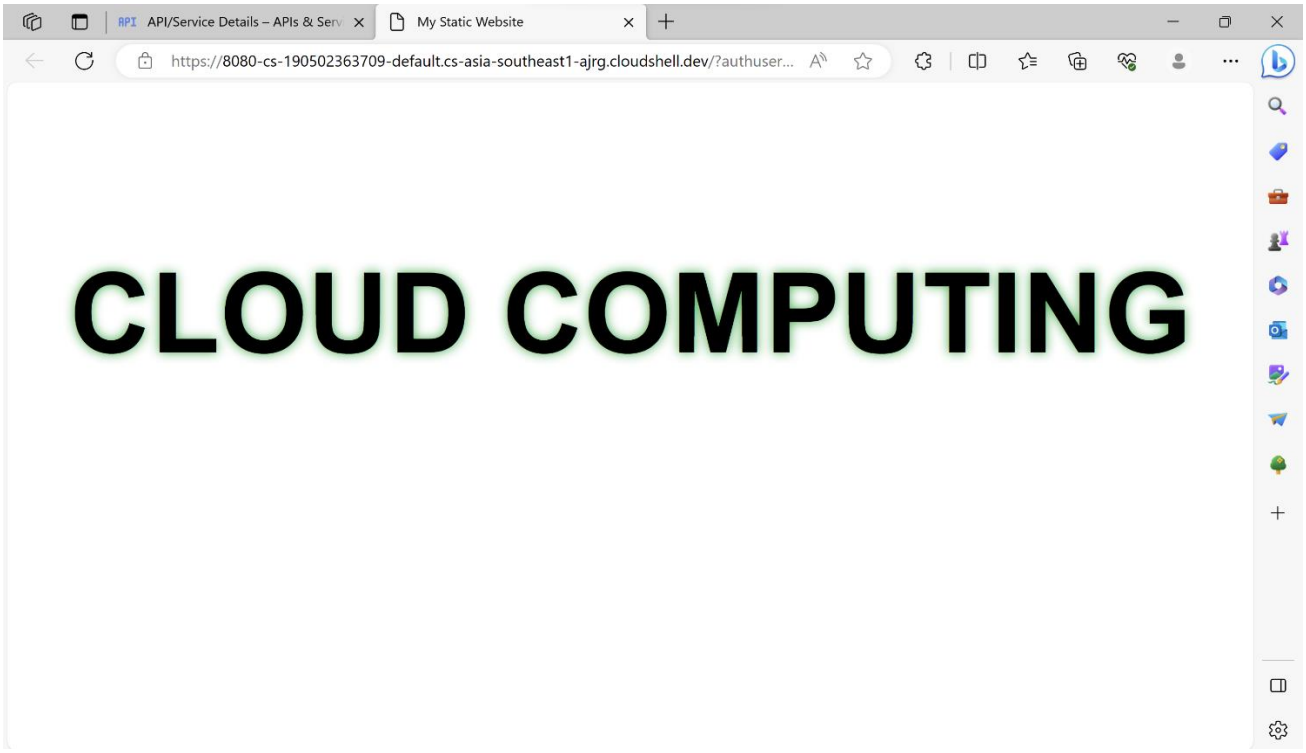
Viewing your application

To launch your browser and view the app at

https://PROJECT_ID.REGION_ID.r.appspot.com, run the following command:

➔ gcloud app browse

OUTPUT :



RESULT :

Thus the GAE launcher used to launch the web applications.