Summit Cloud Solutions Portfolio

This repository contains example projects demonstrating practical AWS and cloud solutions. Each folder represents a separate "gig" example for portfolio purposes.

Projects Overview

1. **EC2 Setup**

- **Folder:** `gig-ec2-setup`
- **Description:** Demonstrates provisioning an EC2 instance running Amazon Linux 2, installing Nginx, and serving a dynamic webpage.
- **Included:**
  - `ec2-user-data.sh` — User data script for instance initialization.
  - Sample HTML page showing instance region and current time.
- **Setup Instructions:**
  1. Launch an EC2 instance (Amazon Linux 2).
  2. Paste `ec2-user-data.sh` into the User Data field.
  3. Access your public IP in a browser to view the page.

---

2. **S3 Static Site**

- **Folder:** `gig-s3-static-site`
- **Description:** Demonstrates hosting a fully functional static website on AWS S3.
- **Included:**
  - `website/index.html`, `style.css`, `script.js`
  - Button displays current client-side time.
- **Setup Instructions:**
  1. Create an S3 bucket.
  2. Enable static website hosting.
  3. Upload the contents of the `website` folder.
  4. Visit the S3 website endpoint to view the site.

---

3. **AWS Amplify React/Vite**

- **Folder:** `gig-aws-amplify/amplify-project`
- **Description:** Demonstrates deploying a React app using AWS Amplify.
- **Included:**
  - React/Vite project (`src/` & `public/`)
  - Package configuration for local development
- **Setup Instructions:**
  1. Install dependencies: `npm install`
  2. Run locally: `npm run dev`
  3. Connect this repo to Amplify Console to deploy the app.

4. **EC2 Express App**

- **Folder:** `gig-ec2-express-app`
- **Description:** Demonstrates running a Node.js Express application on EC2 with a simple API.
- **Included:**
  - `app.js` — Main Express app
  - `routes/index.js` — Homepage route
  - `routes/api.js` — API endpoint returning server time
- **Setup Instructions:**
  1. Launch an EC2 instance (Amazon Linux 2 or Ubuntu).
  2. Install Node.js: `sudo yum install -y nodejs npm`
  3. Upload this project to the instance.
  4. Install dependencies: `npm install`
  5. Start the app: `npm start`
  6. Access:
     - `http://<EC2_PUBLIC_IP>:3000` — homepage
     - `http://<EC2_PUBLIC_IP>:3000/api/time` — server time API

License

MIT License — see LICENSE