Title: Deploying a Static Web Application on Azure.

1. Objective:

The objective of this cloud lab experiment is to deploy a static web application on Microsoft Azure using an Azure free account. This involves integrating code from GitHub to streamline the deployment process and demonstrate the ease of use and efficiency of Azure for hosting static websites.

2. Background:

Theory/Concepts:

- **Static Web Applications:** These are web applications that deliver fixed content to users, often consisting of HTML, CSS, and JavaScript. They do not require server-side processing.
- **Azure Static Web Apps:** A service provided by Azure specifically for hosting static web applications with features like automatic scaling, global distribution, and built-in CI/CD (Continuous Integration/Continuous Deployment) using GitHub Actions.
- **Service Models:** The deployment utilizes the Platform as a Service (PaaS) model, where Azure manages the infrastructure, allowing developers to focus on code rather than hardware management.

3. Tools and Services

- **Azure Static Web Apps:** Used to deploy and host the static web application.
- **GitHub:** Repository for the web application's source code.
- **Azure CLI:** Command-line interface for managing Azure resources.
- **Visual Studio Code (optional):** An IDE for editing the web application's code.

4. Experiment Setup

Step-by-Step Configuration:

Cloud Account Setup:

1. Navigate to the Azure Free Account page.
2. Click on "Start free" and follow the prompts to create an Azure account, providing necessary details such as email and payment information (you will not be charged during the trial).

Environment Configuration:

1. After logging in, go to the Azure portal.
2. Select "Create a resource" and choose "Static Web App."
3. Fill in the necessary details:
    - **Name:** Choose a unique name for the static web app.
    - **Region:** Select a suitable region for hosting.
    - **Source:** Choose GitHub as the source control.
    - **GitHub Account:** Authorize Azure to access your GitHub account.

5. Execution

Tasks Performed:

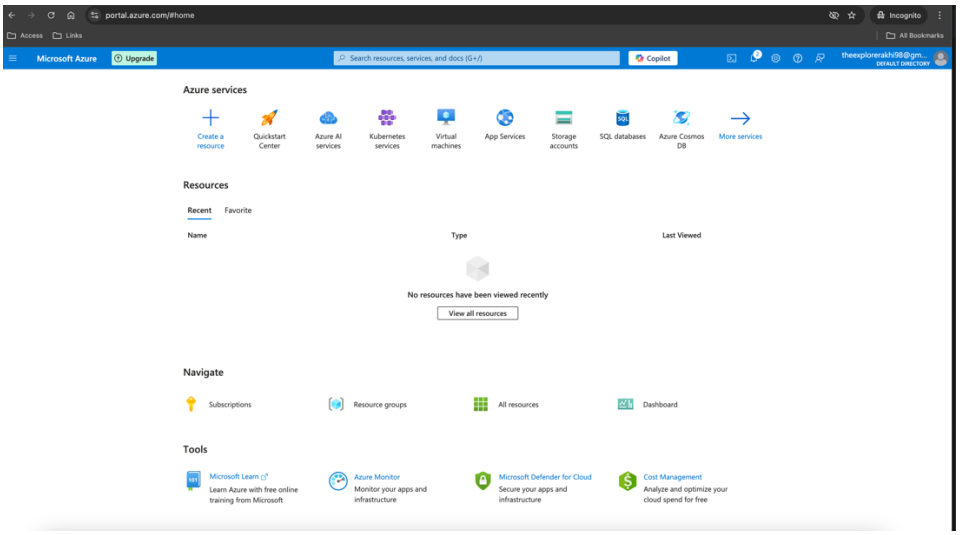
1. Created a free Azure account.
2. Configured the Azure Static Web App service.
3. Integrated the GitHub repository for continuous deployment.
4. Monitored the deployment process via the Azure portal.

Monitoring:

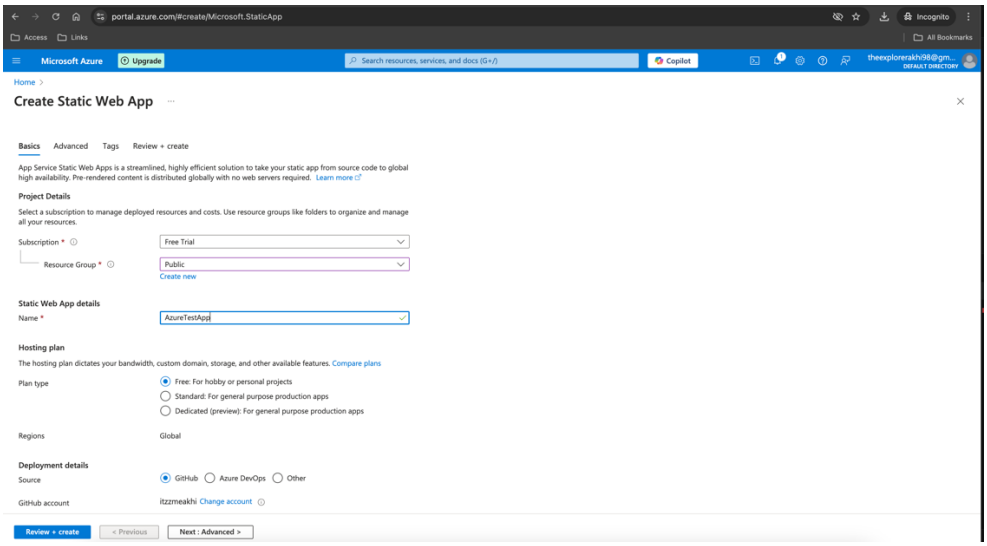
- Use Azure Monitor to track the performance and health of the deployed application. Set up alerts for issues such as downtime or performance degradation.

6. Observations

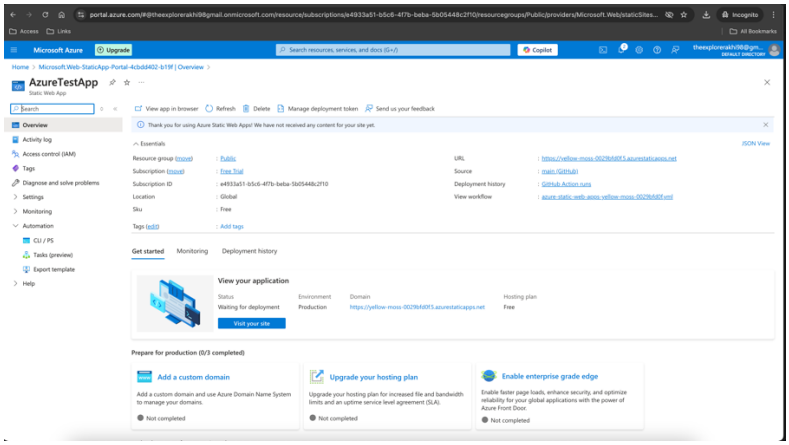
After creating account, login into azure dashboard



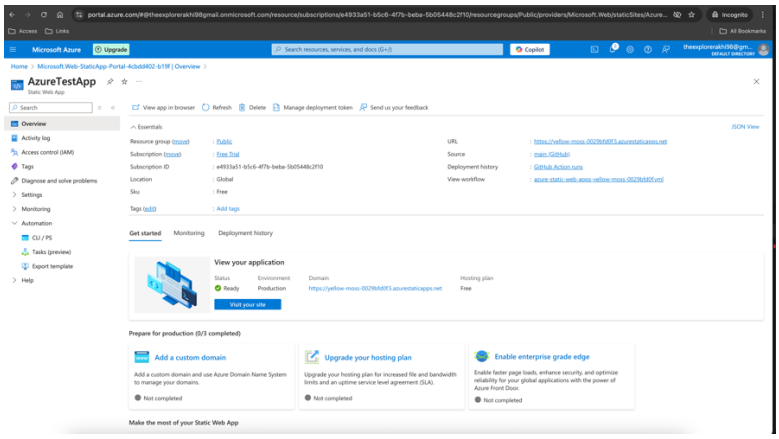
Creating a static web app



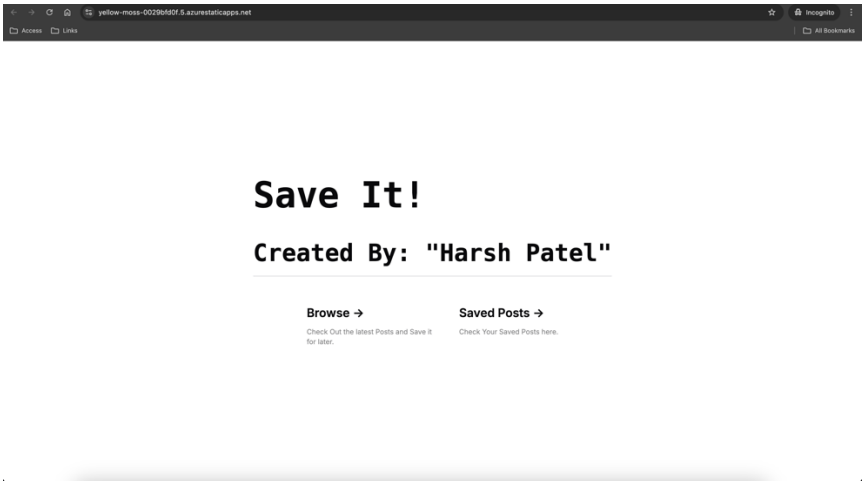
Deploying web app by integrating GitHub



After the successful deployment, we can hit the link mentioned to access webapp



Accessing the web app with azure deployed link



7. Results

- The static web application was successfully deployed on Azure Static Web Apps, functioning as expected. The integration with GitHub allowed for a seamless CI/CD process, enabling automatic updates upon code changes.