Quickstart: Building your first static site with Azure Static Web Apps

08/13/2020 • 2 minutes to read •

In this article

Azure Static Web Apps publishes a website by building apps from a code repository. In this quickstart, you deploy an application to Azure Static Web apps using the Visual Studio Code extension.

If you don't have an Azure subscription, create a free trial account .

Prerequisites

- GitHub account
- Azure account
- Visual Studio Code
- Azure Static Web Apps extension for Visual Studio Code
- Install Git

Create a repository

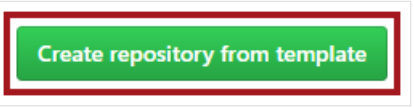
This article uses a GitHub template repository to make it easy for you to get started. The template features a starter app used to deploy using Azure Static Web Apps.

No Framework | Angular | React | Vue

1. Navigate to the following location to create a new repository:
   a. https://github.com/staticwebdev/vanilla-basic/generate
2. Name your repository **my-first-static-web-app**

> **Note**
>
> Azure Static Web Apps requires at least one HTML file to create a web app. The repository you create in this step includes a single *index.html* file.

Select **Create repository from template**.



Clone the repository

With the repository created in your GitHub account, clone the project to your local machine using the following command.

Bash Copy

```
git clone https://github.com/<YOUR_GITHUB_ACCOUNT_NAME>/my-first-static-web-app.git
```

Make sure to replace `<YOUR_GITHUB_ACCOUNT_NAME>` with your GitHub username.

Next, open Visual Studio Code and go to **File** > **Open Folder** to open the repository you cloned to your machine in the editor.

Create a static web app

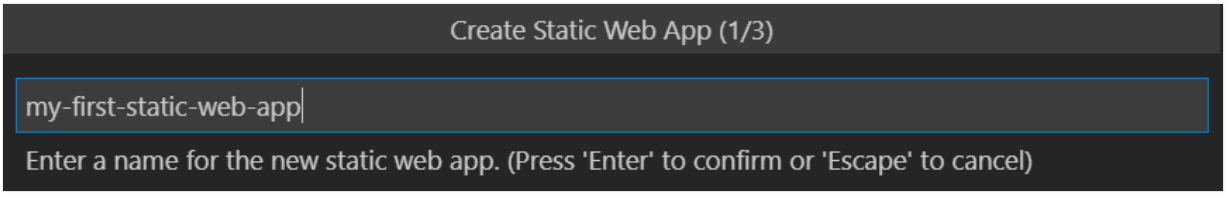
1. Inside Visual Studio Code, select the Azure logo in the Activity Bar to open the Azure extensions window.

> **Note**
>
> Azure and GitHub sign in are required. If you are not already signed in to Azure and GitHub from Visual Studio Code, the extension will prompt you to sign in to both during the creation process.

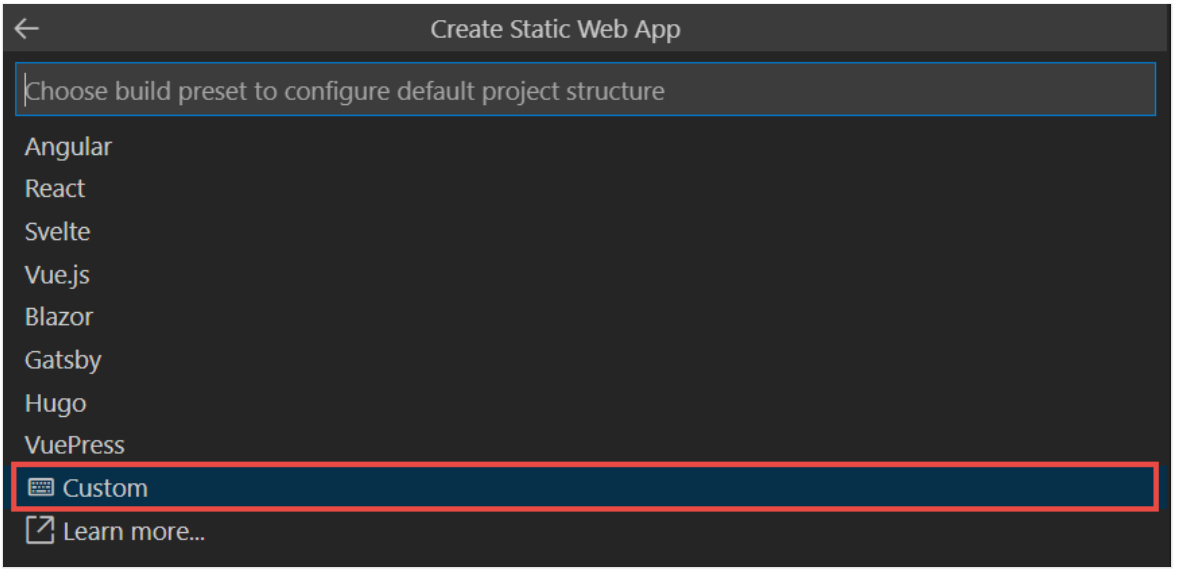
2. Under the *Static Web Apps* label, select the **plus sign**.

3. The command palate opens at the top of the editor and prompts you to name your application.

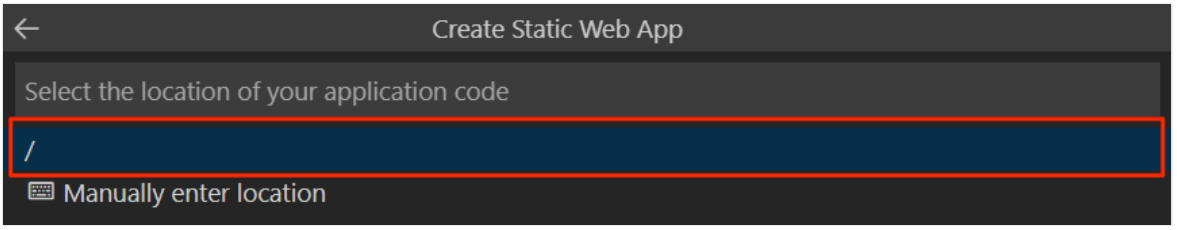
   Type **my-first-static-web-app** and press **Enter**.



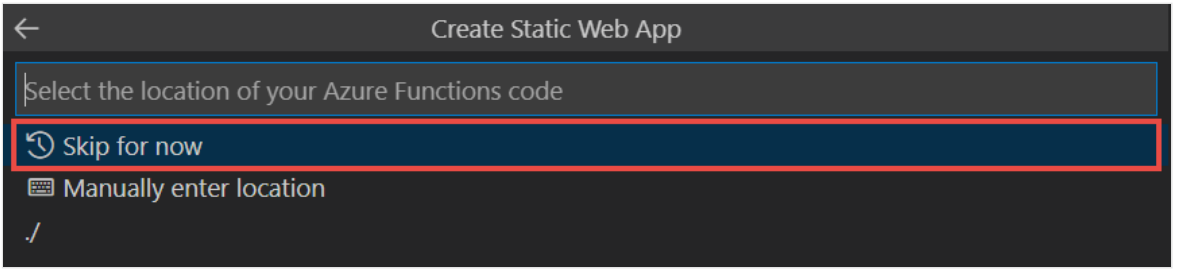
4. Select the presets that match your application type.



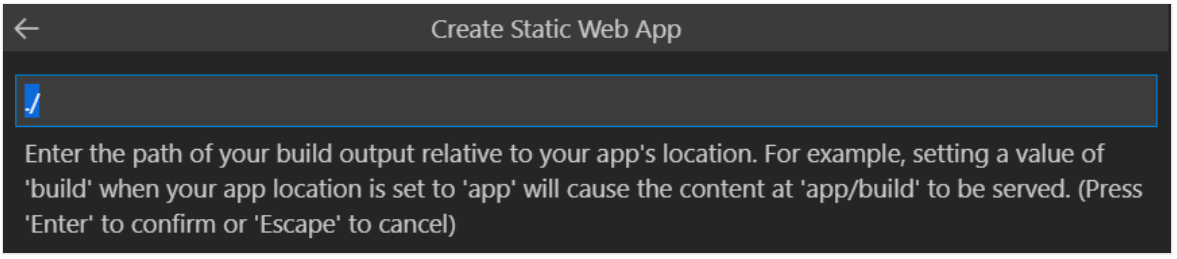
Enter **./** as the location for the application files.



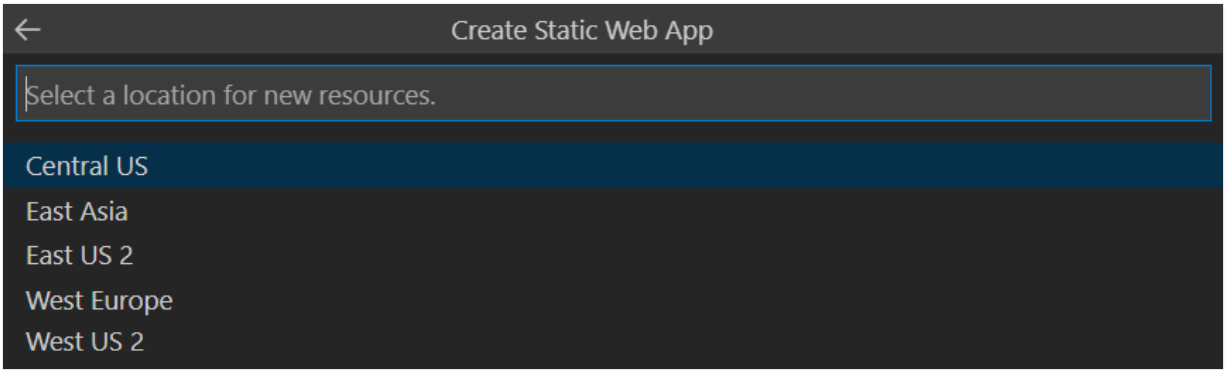
Select **Skip for now** as the location for the Azure Functions API.



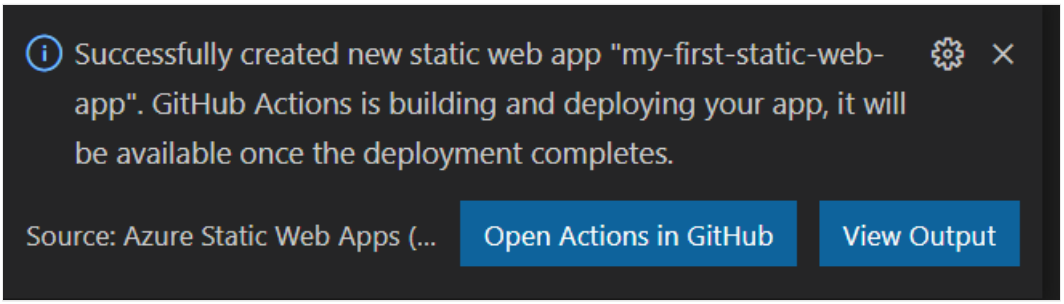
Enter **./** as the build output location.



5. Select a location nearest you and press **Enter**.



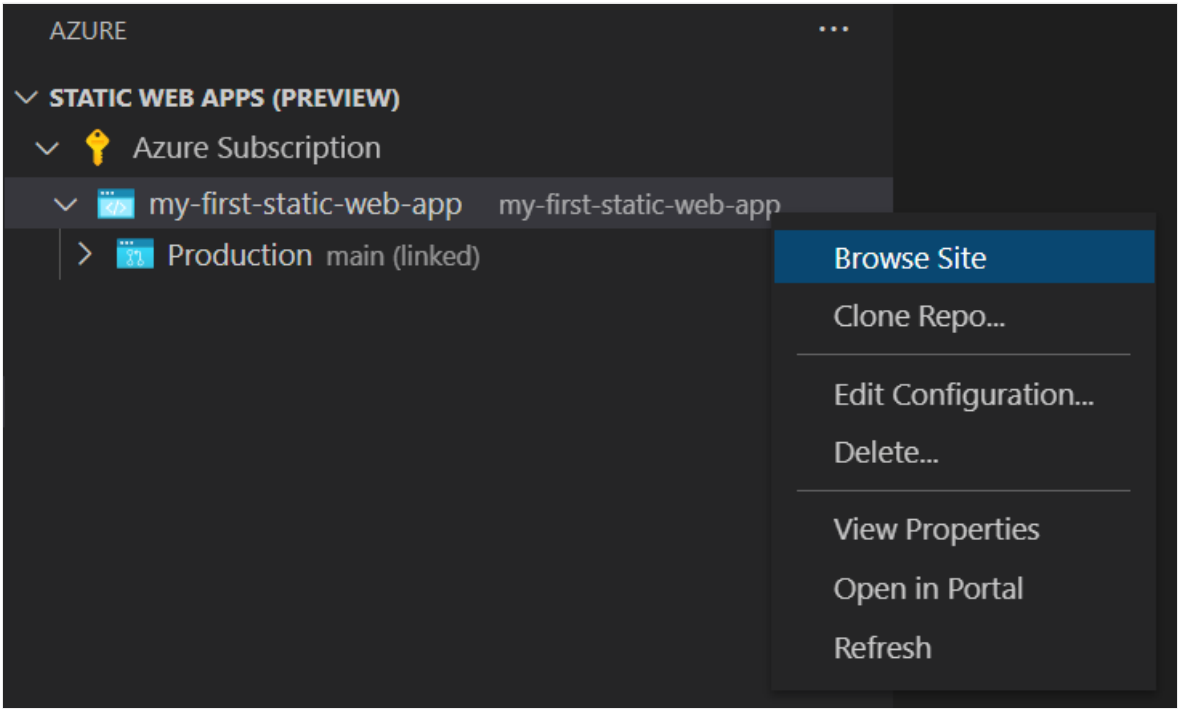
6. Once the app is created, a confirmation notification is shown in Visual Studio Code.



   Next, click on the button **Open Actions in GitHub**. This page shows you the build status of the application.

   Once the GitHub Action is complete, then you can browse to the published website.

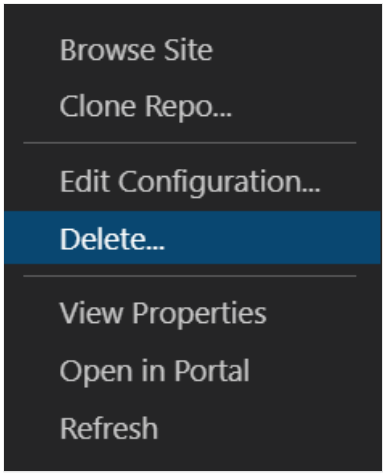
7. To view the website in the browser, right-click on the project in the Static Web Apps extension, and select **Browse Site**.



Clean up resources

If you're not going to continue to use this application, you can delete the Azure Static Web Apps instance through the extension.

In the Visual Studio Code Explorer window, return to the *Static Web Apps* section and right-click on **my-first-static-web-app** and select **Delete**.



Next steps

Add an API

Is this page helpful?

Yes No

Recommended content

Quickstart: Building your first static web app with Azure Static Web Apps using the Azure portal

Learn to deploy a static site to Azure Static Web Apps with the Azure portal.

GitHub Actions workflows for Azure Static Web Apps

Learn how to use GitHub repositories to set up continuous deployment to Azure Static Web Apps.

Configure front-end frameworks with Azure Static Web Apps

Settings for popular front-end frameworks needed for Azure Static Web Apps

What is Azure Static Web Apps?

The key features and functionality of Azure Static Web Apps.

Show more