Hands-on tutorials

Host a Static Website

Host a Static Website: Hands-on tutorials

Table of Contents

Host a Static Website

AWS experience	Beginner
Time to complete	10 minutes
Cost to complete	Total cost of hosting your static website on AWS is dependent on your usage • Outside of AWS Free Tier Limits: typically $1-3/mo. • Within AWS Free Tier Limits: typically $0.50/mo. To see a breakdown of the services used and their associated costs, see pricing for AWS Amplify and Amazon Route 53.
Get help	Troubleshooting Amplify
Last updated	July 16, 2024

Overview

In this tutorial, you will learn how to deploy a static website with AWS Amplify. Amplify offers a Git-based CI/CD workflow for building, deploying, and hosting websites. Static websites deliver HTML, JavaScript, images, video and other files to your website visitors. Static websites are very low cost, provide high-levels of reliability, require almost no IT administration, and scale to handle enterprise-level traffic with no additional work.

For more information, see the FAQs.

What you will accomplish

In this tutorial, you will:

- **Host a static website** using AWS Amplify in the AWS Management Console. AWS Amplify provides fully managed hosting for static websites and web apps. Amplify's hosting solution leverages Amazon CloudFront and Amazon S3 to deliver your site assets via the AWS content delivery network (CDN).
- **Set up continuous deployment:** Amplify offers a Git-based workflow with continuous deployment, allowing you to automatically deploy updates to your site on every code commit.

Prerequisites

Before starting this tutorial, you will need:

- An **AWS account**: if you don't already have one follow the Setup Your Environment tutorial.
- Your **AWS profile** configured for local development.
- **Installed** on your environment: Nodejs and npm.
- Familiarity with git and a Github account.

Implementation

Step 1: Create a new React application

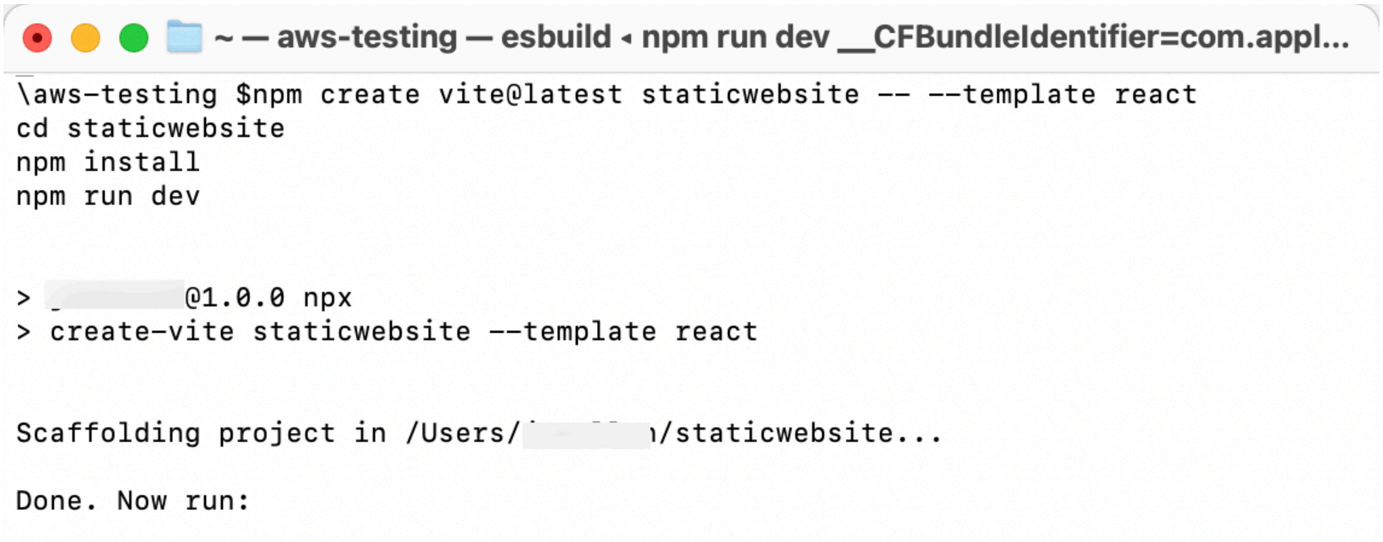
Already have a repository to connect? Skip to Initialize GitHub Repository

Want to deploy without connecting to a Git provider? Follow these instructions

1. Create the application

   In a new terminal window or command line, **run** the following command to use Vite to create a React application:

   ```
   npm create vite@latest staticwebsite -- --template react
   cd staticwebsite
   npm install
   npm run dev
   ```

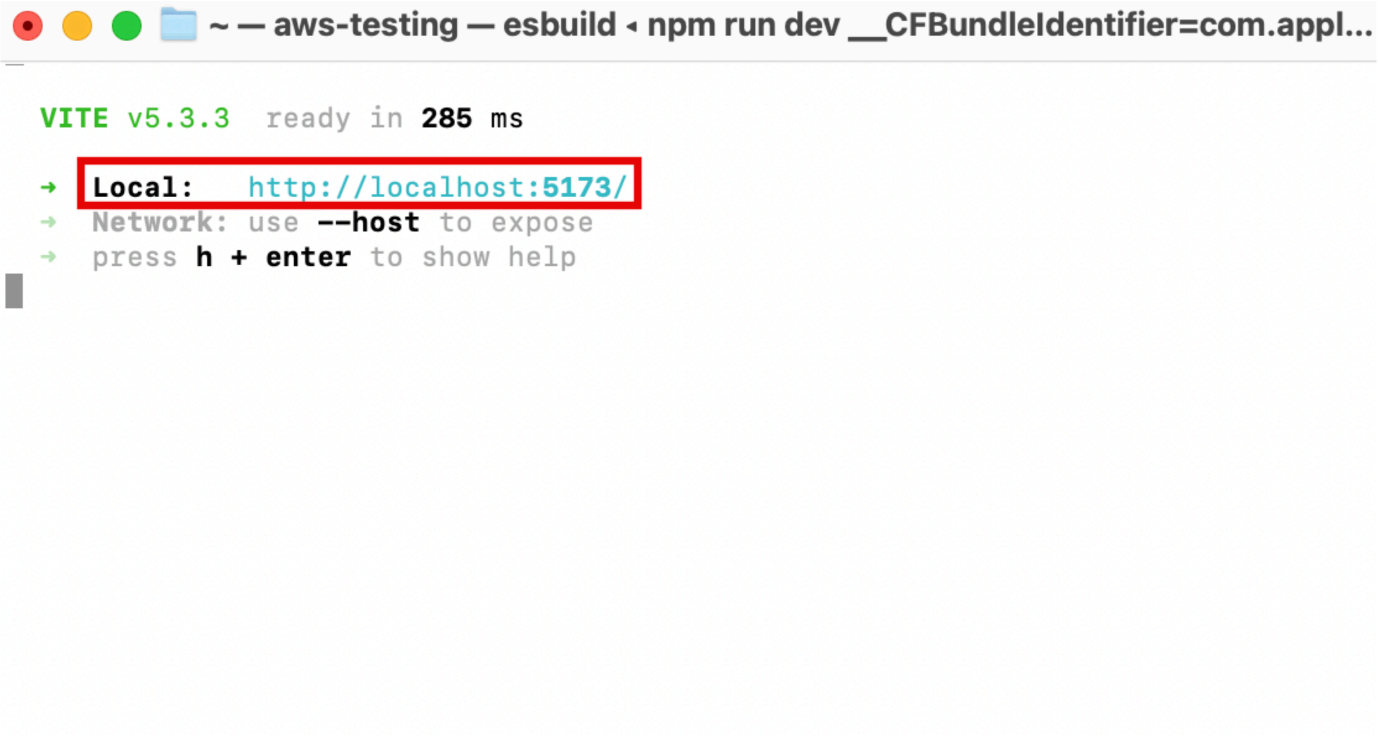

~ — aws-testing — esbuild ◂ npm run dev __CFBundleIdentifier=com.appl...

```
\aws-testing $npm create vite@latest staticwebsite -- --template react
cd staticwebsite
npm install
npm run dev


> [redacted]@1.0.0 npx
> create-vite staticwebsite --template react


Scaffolding project in /Users/[redacted]/staticwebsite...

Done. Now run:
```

2. View the application

   In the terminal window, select and open the **Local link** to view the Vite + React application.

```
npm create vite@latest staticwebsite -- --template react
cd staticwebsite
npm install
npm run dev
```



~ — aws-testing — esbuild ◂ npm run dev __CFBundleIdentifier=com.appl...

```
  VITE v5.3.3  ready in 285 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

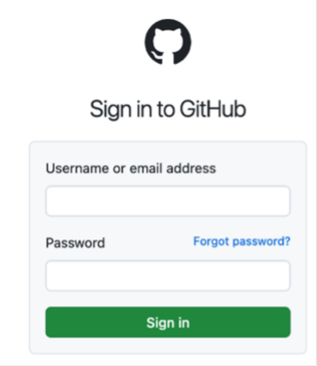
Step 2: Initialize GitHub Repository

In this step, you will create a GitHub repository and commit your code to the repository. You will need a GitHub account to complete this step, if you do not have an account, sign up here.

> **Note**
>
> If you have never used GitHub on your computer, follow the steps to generate and add an SSH key to your account before continuing to allow connection to your account.

1. Open GitHub

   Sign in to GitHub at https://github.com/.



Sign in to GitHub

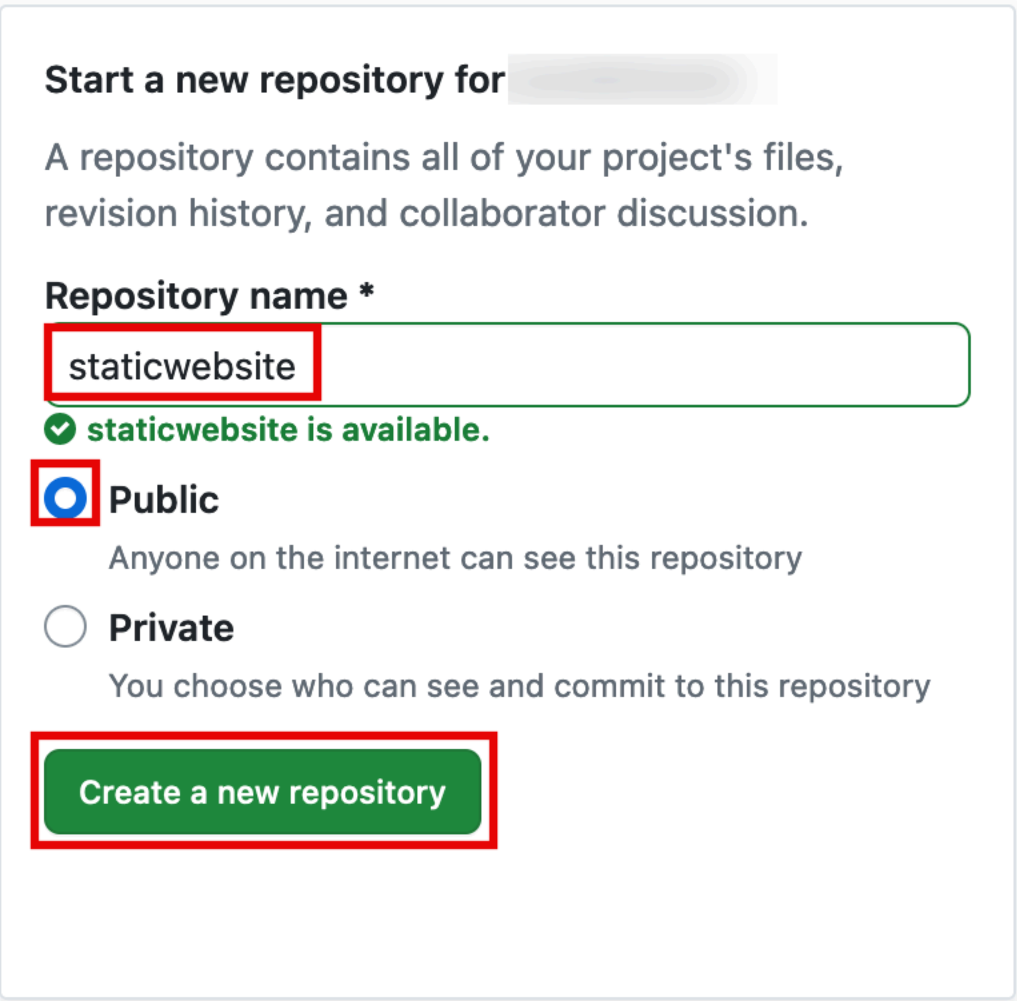
Username or email address

Password

2. Create a repository

   In the **Start a new repository** section, make the following selections:

   - For **Repository name**, enter **staticwebsite**, and choose the **Public** radio button.
   - Then select, **Create a new repository**.



Start a new repository for [redacted]

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name *

staticwebsite

✓ staticwebsite is available.

● Public

Anyone on the internet can see this repository

○ Private

You choose who can see and commit to this repository

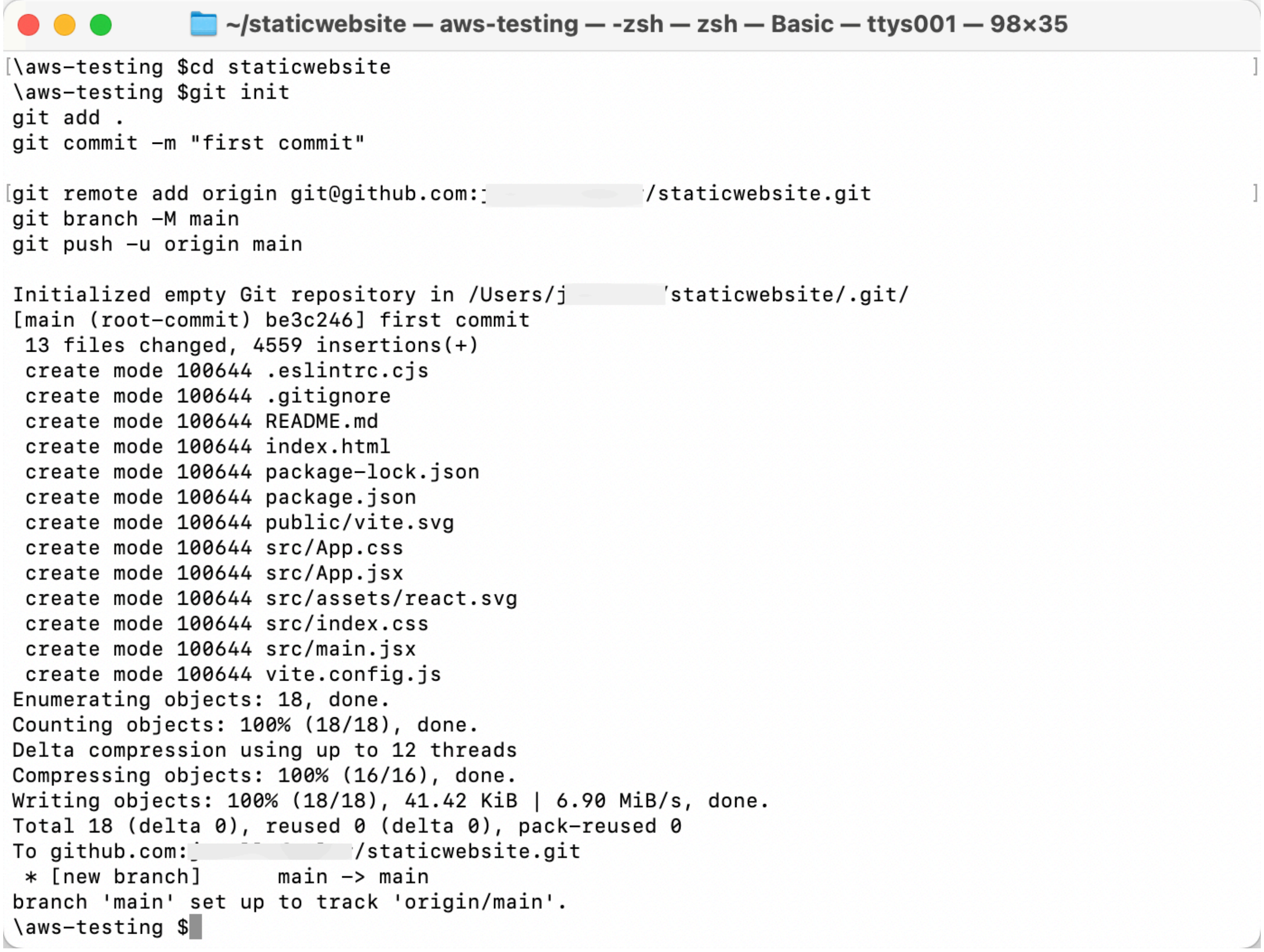
3. Push the application to GitHub

   Open a new terminal window, **navigate** to your projects root folder (**staticwebsite**), and **run** the following commands to initialize a git and push the application to the new GitHub repo:

   > **Note**
   >
   > Replace the **SSH GitHub URL** in the command with your SSH GitHub URL.

   ```
   git init
   git add .
   git commit -m "first commit"
   git remote add origin git@github.com:<your-username>/staticwebsite.git
   ```

```
git branch -M main
git push -u origin main
```


~/staticwebsite — aws-testing — -zsh — zsh — Basic — ttys001 — 98×35

```
[\aws-testing $cd staticwebsite
\aws-testing $git init
git add .
git commit -m "first commit"

[git remote add origin git@github.com:[redacted]/staticwebsite.git
git branch -M main
git push -u origin main

Initialized empty Git repository in /Users/j[redacted]'staticwebsite/.git/
[main (root-commit) be3c246] first commit
 13 files changed, 4559 insertions(+)
 create mode 100644 .eslintrc.cjs
 create mode 100644 .gitignore
 create mode 100644 README.md
 create mode 100644 index.html
 create mode 100644 package-lock.json
 create mode 100644 package.json
 create mode 100644 public/vite.svg
 create mode 100644 src/App.css
 create mode 100644 src/App.jsx
 create mode 100644 src/assets/react.svg
 create mode 100644 src/index.css
 create mode 100644 src/main.jsx
 create mode 100644 vite.config.js
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 41.42 KiB | 6.90 MiB/s, done.
Total 18 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:[redacted]/staticwebsite.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
\aws-testing $
```

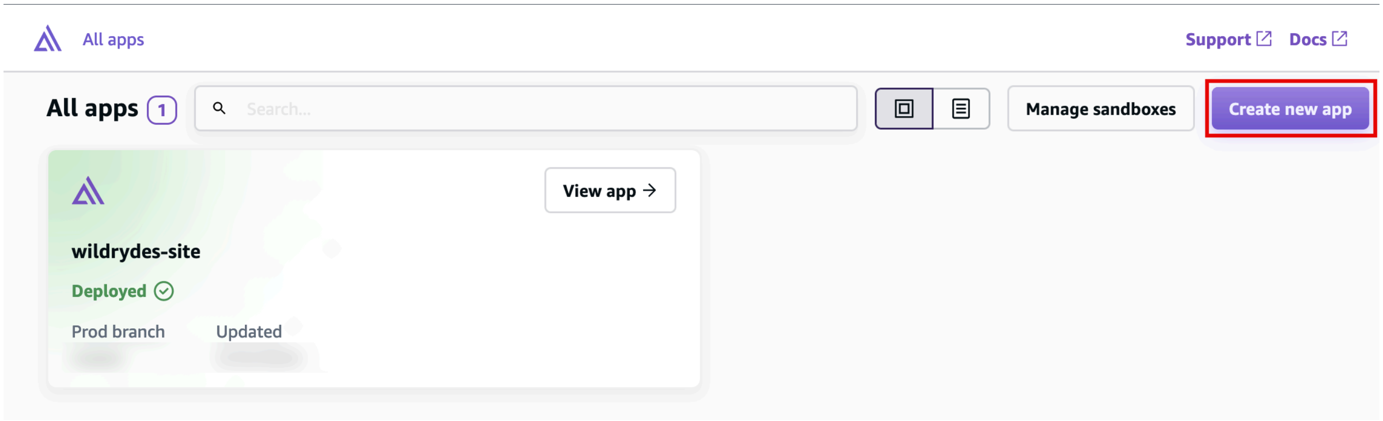
Step 3: Deploy your app with AWS Amplify

In this step, you will connect the GitHub repository you just created to AWS Amplify. This will enable you to build, deploy, and host your app on AWS.

1. Create an Amplify app

   Sign in to the AWS Management Console in a new browser window, and open the AWS Amplify console at https://console.aws.amazon.com/amplify/apps.

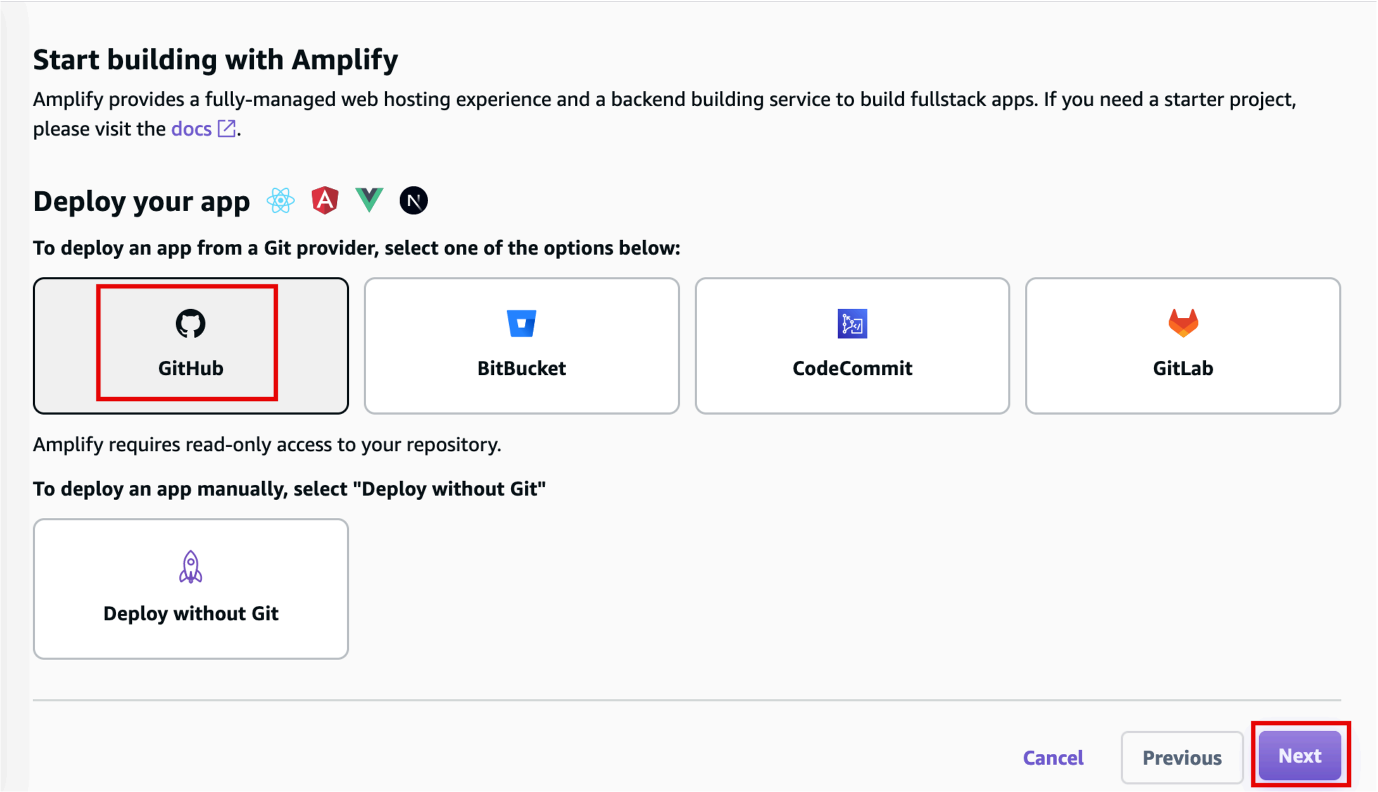
   Choose **Create new app.**



2. Choose the GitHub repository

On the **Start building with Amplify** page, for **Deploy your app**, select **GitHub**, and select **Next**.

> **Notes**
>
> - If you are using an existing repository, connect your GitHub, Bitbucket, GitLab, or AWS CodeCommit repositories.
> - You also have the option of manually uploading your build artifacts without connecting a Git repository (see Manual Deploys).
> - After you authorize the Amplify console, Amplify fetches an access token from the repository provider, but it doesn't store the token on the AWS servers. Amplify accesses your repository using deploy keys installed in a specific repository only.



Start building with Amplify

Amplify provides a fully-managed web hosting experience and a backend building service to build fullstack apps. If you need a starter project, please visit the docs.

Deploy your app

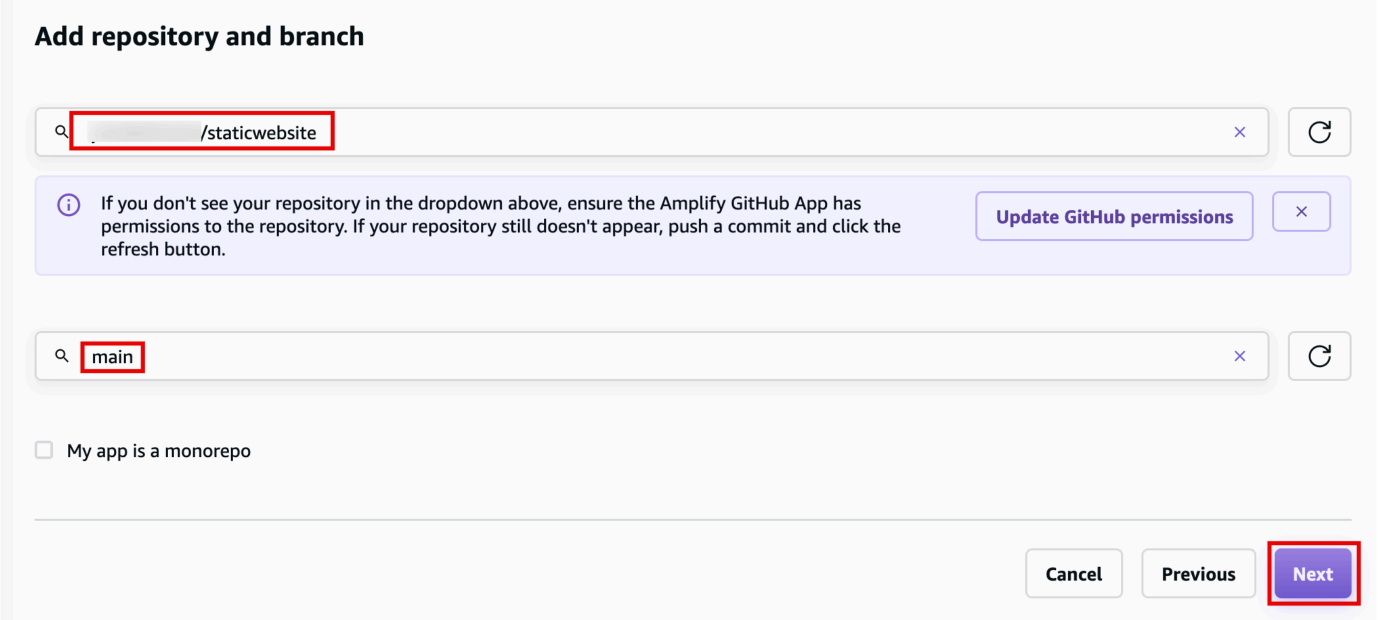
To deploy an app from a Git provider, select one of the options below:

Amplify requires read-only access to your repository.

To deploy an app manually, select "Deploy without Git"

3. Select repository branch

   When prompted, **authenticate** with GitHub. You will be automatically redirected back to the Amplify console. Choose the **repository** and **main branch** you created earlier. Then select **Next**.



Add repository and branch

[redacted]/staticwebsite

If you don't see your repository in the dropdown above, ensure the Amplify GitHub App has permissions to the repository. If your repository still doesn't appear, push a commit and click the refresh button.

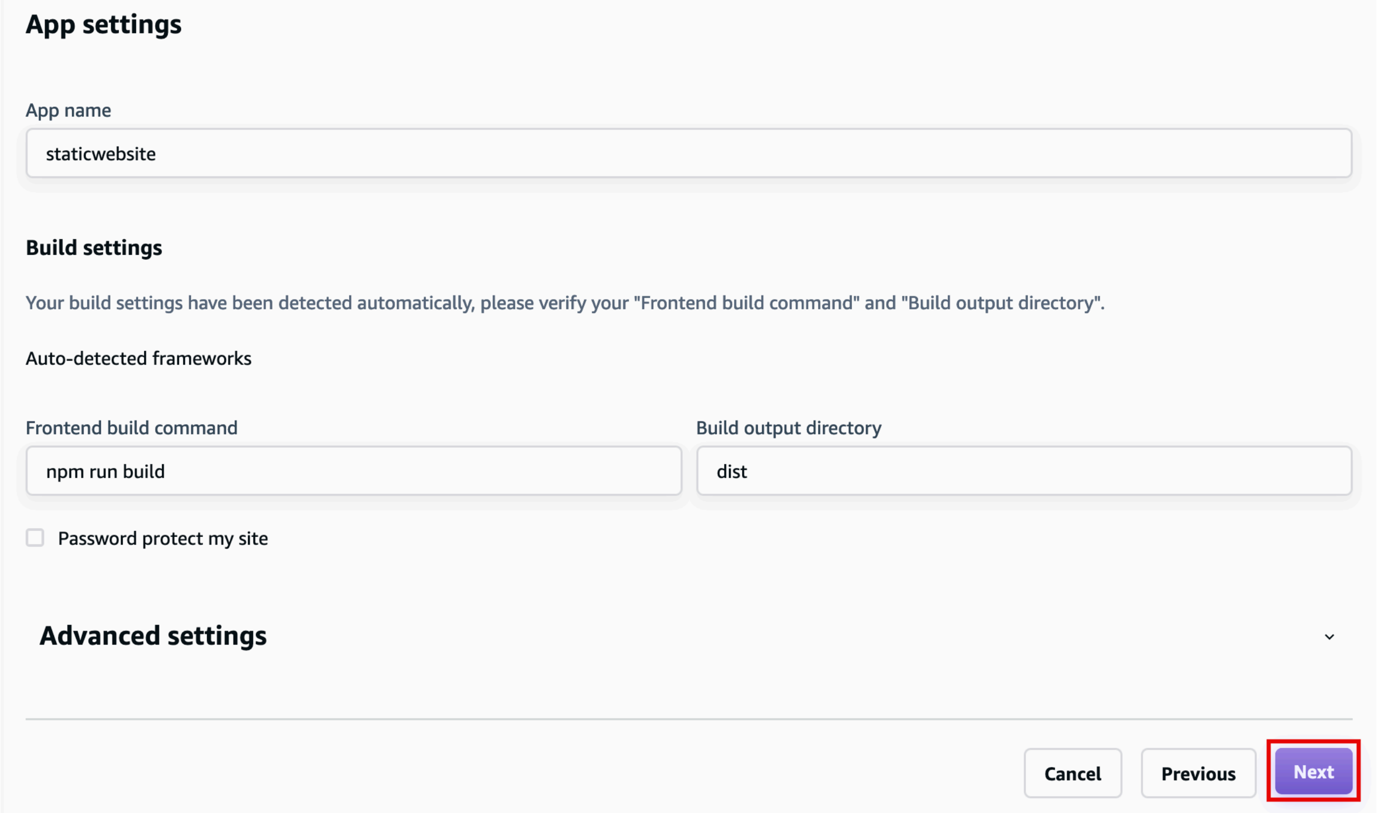
main

□ My app is a monorepo

4. Review build settings

Leave the default **build settings** and select **Next**.

- Amplify inspects your repository to automatically detect the sequence of build commands to be invoked.



App settings

App name

staticwebsite

Build settings

Your build settings have been detected automatically, please verify your "Frontend build command" and "Build output directory".

Auto-detected frameworks

Frontend build command: npm run build

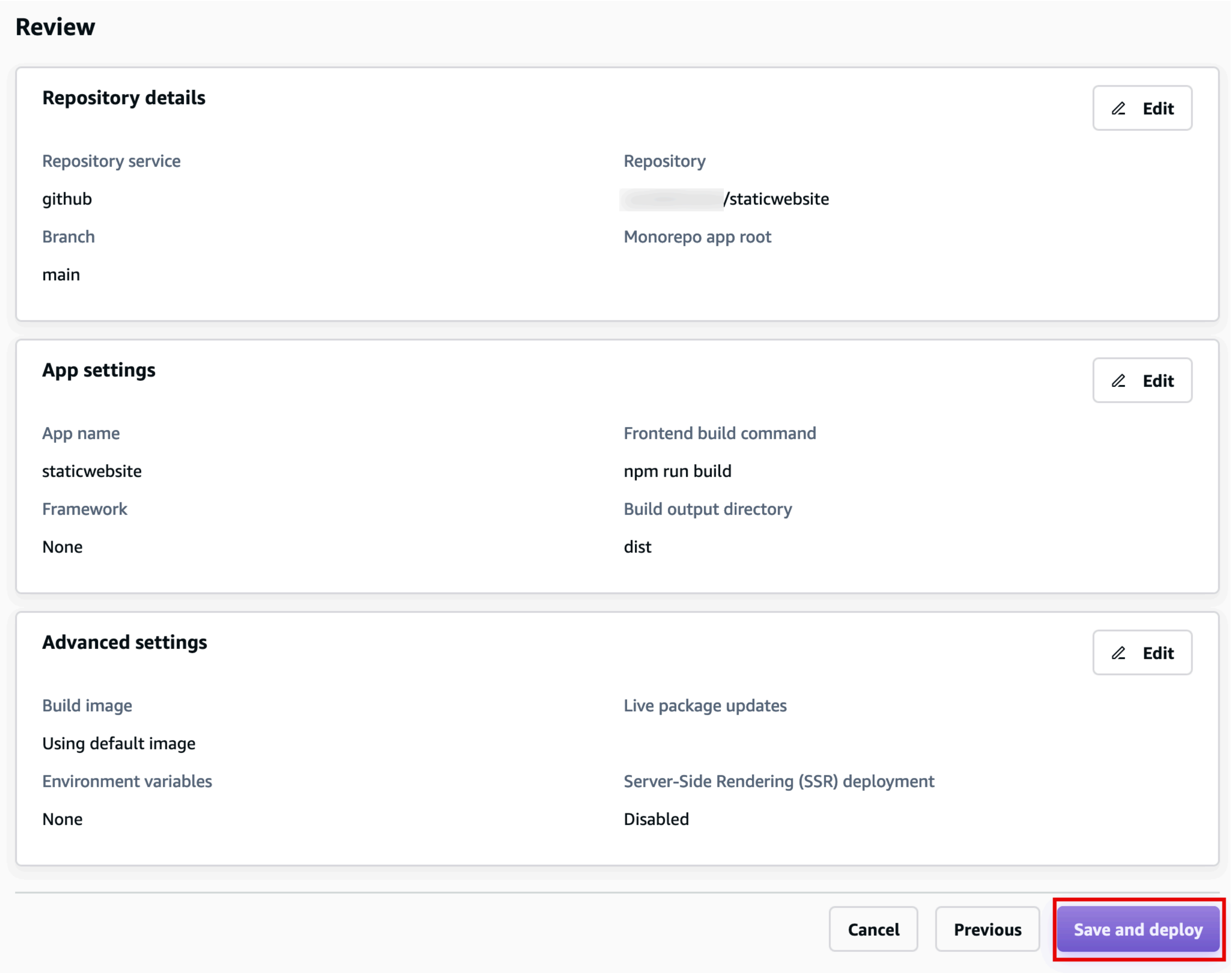
Build output directory: dist

□ Password protect my site

Advanced settings

5. Deploy the app

Review the inputs selected, and choose **Save and deploy** to deploy your web app to a global content delivery network (CDN).



Review

Repository details

Repository service
github

Repository
[redacted]/staticwebsite

Branch
main

Monorepo app root

App settings

App name
staticwebsite

Frontend build command
npm run build

Framework
None

Build output directory
dist

Advanced settings

Build image
Using default image

Live package updates

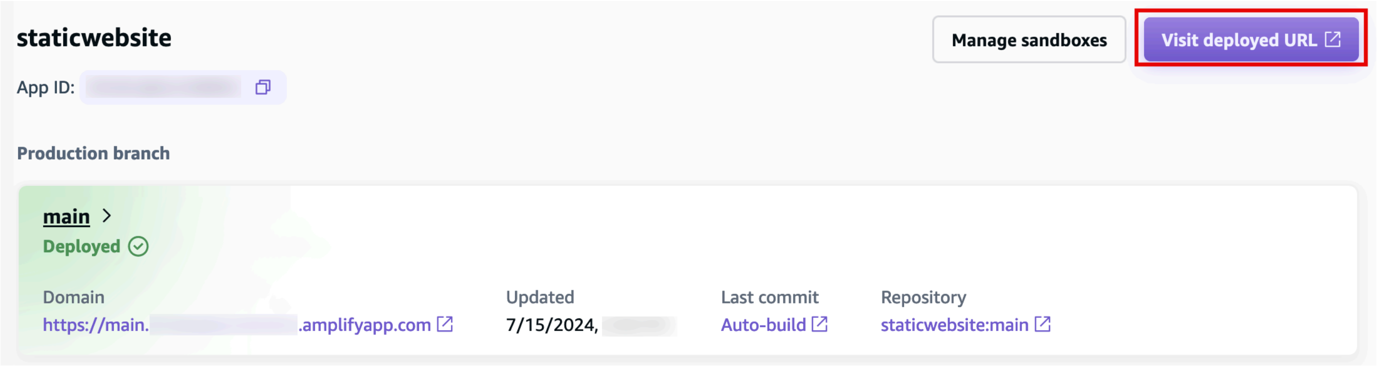
Environment variables
None

Server-Side Rendering (SSR) deployment
Disabled

6. View your deployed app

AWS Amplify will now build your source code and deploy your app at **https://...amplifyapp.com**, and on every git push your deployment instance will update. It may take 2-5 minutes to deploy your app based on the size.

Once the build completes, select the **Visit deployed URL** button to see your web app up and running live.



staticwebsite

App ID: [redacted]

Production branch

main >

Deployed

Domain: https://main.[redacted].amplifyapp.com

Updated: 7/15/2024, [redacted]

Last commit: Auto-build

Repository: staticwebsite:main

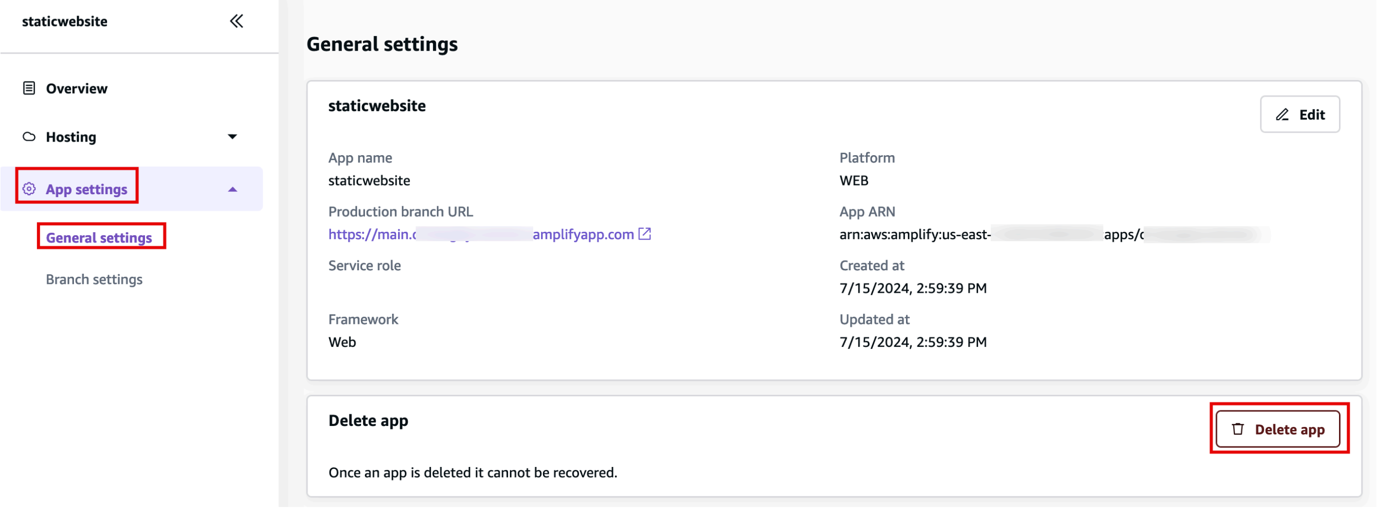
Clean up resources

It is recommended that you delete the app and the backend resources that you created during this tutorial to prevent unexpected costs.

- Delete the app

  In the Amplify console, in the left-hand navigation for the **staticwebsite** app, choose **App settings**, and select **General settings**.

  In the General settings section, choose **Delete app**.



General settings

staticwebsite

App name: staticwebsite

Platform: WEB

Production branch URL: https://main.[redacted]amplifyapp.com

App ARN: arn:aws:amplify:us-east-[redacted]apps/[redacted]

Service role

Created at: 7/15/2024, 2:59:39 PM

Framework: Web

Updated at: 7/15/2024, 2:59:39 PM

Delete app

Once an app is deleted it cannot be recovered.

Congratulations

You have finished the **Host a Static Website on AWS** tutorial!

Frequently Asked Questions

Understanding static websites

What is a static website?

A static website consists of a set of HTML, CSS, and JavaScript files that serve website content. Simple static websites have no dynamic functionality, and are best used for personal or marketing sites. Modern static websites (or single page web apps) built with JavaScript frameworks such as React and Angular, allow developers to build rich app experiences on the web that were traditionally only possible with server-side languages such as PHP and ASP.NET. Hosting a static website does not require web servers and can use content delivery networks to store content (HTML, CSS and JavaScript files), making it easy to scale with increased user traffic.

How are modern static websites different from server-rendered websites?

Most individuals use text editors or website design software to build their websites. If you don't have the skills or you'd rather not develop the site, you can hire an agency to build your site.

What is web site hosting?

The infrastructure environment used to enable end users to request, receive, and interact with your website is commonly referred to as "hosting." AWS Amplify is a fully managed hosting service that creates and maintains the hosting environment for your site. AWS Amplify provides a seamless developer experience and automatic workflow for hosting websites and leverages Amazon S3 and Amazon CloudFront for storage and global content delivery.

Benefits and cost

Why should I use a static website hosting environment?

Static website hosting is the lowest cost and lowest maintenance option (for example, there are no servers to maintain) and it provides high levels of reliability and scale.

How much will it cost to host my website?

The total cost of hosting your personal website on AWS will vary depending on your usage. Typically, it will cost $1-3/month if you are outside the AWS Free Tier limits. If you are eligible for

AWS Free Tier and within the limits, hosting your personal website will cost around $0.50/month. To see a breakdown of the services used and their associated costs, see pricing for AWS Amplify and Amazon Route 53 .

Getting started and maintenance

I don't have a website, how should I build it?

Most individuals use text editors or website design software to build their websites. If you don't have the skills or you'd rather not develop the site, you can hire an agency to build your site.

How much effort will it take to maintain my website on AWS?

Static websites don't require you to maintain servers, and no ongoing maintenance is required for the two services you'll use to host your website: AWS Amplify and Amazon Route 53. The time you spend editing and deploying new content will depend on how frequently you change your website.

Customization and enhancement

Can I customize my website once it is hosted on AWS?

Yes. Once you website is up and running, there are multiple ways you can add additional AWS services and functionality to your site. For example, you can add authentication to your website enabling users to log into your website with AWS Amplify and evolve it into a static web application with cloud functionality.