Panduan Pengguna

AWS Amplify Hosting

AWS Amplify Hosting: Panduan Pengguna

Table of Contents

Selamat datang di AWS Amplify Hosting

Hosting Amplify menyediakan alur kerja berbasis GIT untuk hosting aplikasi web nirserver full-stack dengan deployment kontinu. Amplify men-deploy aplikasi ke jaringan pengiriman konten AWS global (CDN). Panduan pengguna ini memberikan informasi yang Anda butuhkan untuk memulai Amplify Hosting.

Kerangka kerja yang didukung

Amplify Hosting mendukung banyak kerangka kerja SSR umum, kerangka kerja aplikasi satu halaman (SPA), dan generator situs statis, termasuk yang berikut ini.

Kerangka kerja SSR

- Next.js
- Nuxt
- Astrodengan adaptor komunitas
- SvelteKitdengan adaptor komunitas
- Kerangka SSR apa pun dengan adaptor khusus

Kerangka kerja SPA

- React
- Angular
- Vue.js
- Ionic
- Ember

Generator situs statis

- Eleventy
- Gatsby
- Hugo
- Jekyll

- VuePress

Fitur Hosting Amplify

Cabang fitur

Kelola lingkungan produksi dan penentuan tahap untuk frontend dan backend dengan menghubungkan cabang-cabang baru.

Domain kustom

Hubungkan aplikasi ke domain kustom.

Tarik pratinjau permintaan

Pratinjau perubahan selama tinjauan kode.

End-to-end pengujian

Tingkatkan kualitas aplikasi dengan end-to-end pengujian.

Cabang yang dilindungi kata sandi

Kata sandi melindungi aplikasi web Anda sehingga Anda dapat mengerjakan fitur baru tanpa membuatnya dapat diakses secara publik.

Mengalihkan dan menulis ulang

Siapkan penulisan ulang dan pengalihan untuk mempertahankan peringkat SEO dan lalu lintas rute berdasarkan persyaratan aplikasi klien.

Deployment atom

Deployment atom menghilangkan jendela pemeliharaan dengan memastikan bahwa aplikasi web Anda diperbarui hanya setelah seluruh deployment selesai. Deployment atom menghilangkan skenario di mana file gagal diunggah dengan benar.

Memulai dengan Amplify

Untuk memulai dengan Amplify Hosting, lihat tutorial. Memulai dengan menerapkan aplikasi ke Amplify Hosting Setelah menyelesaikan tutorial, Anda akan tahu cara menghubungkan aplikasi web di repositori Git (GitHub,, BitBucket GitLab, atau AWS CodeCommit) dan menerapkannya ke Amplify Hosting dengan penerapan berkelanjutan.

Membangun backend

AWS Amplify Gen 2 memperkenalkan pengalaman pengembang TypeScript berbasis kode pertama untuk mendefinisikan backend. Untuk mempelajari cara menggunakan Amplify Gen 2 untuk membangun dan menghubungkan backend ke aplikasi Anda, lihat Membangun & menghubungkan backend di dokumen Amplify.

Untuk lebih memahami pendekatan kode pertama AmplifyGen 2, lihat Lokakarya Amplify Gen 2 di situs web Workshop Studio.AWS Dalam tutorial komprehensif ini, Anda membangun aplikasi tanpa server dengan React dan Next.js dan mempelajari cara menggunakan pustaka Data dan Auth Amplify Gen 2 dan pustaka Amplify UI untuk menambahkan fungsionalitas ke aplikasi.

Jika Anda mencari dokumentasi untuk membangun backend untuk aplikasi Gen 1, menggunakan CLI dan Amplify Studio, lihat Membangun & menghubungkan backend di dokumen Gen 1 Amplify.

Harga Amplify

AWS Amplify Anda hanya mengenakan biaya atas apa yang Anda gunakan. Untuk informasi selengkapnya, lihat Harga AWS Amplify.

Memulai dengan menerapkan aplikasi ke Amplify Hosting

Untuk membantu Anda memahami cara kerja Amplify Hosting, tutorial berikut memandu Anda dalam membangun dan menerapkan aplikasi yang dibuat menggunakan kerangka kerja SSR umum yang didukung Amplify.

Tutorial

- Menerapkan aplikasi Next.js untuk Amplify Hosting
- Menerapkan aplikasi Nuxt.js untuk Amplify Hosting
- Menerapkan aplikasi Astro.js untuk Amplify Hosting
- Menerapkan SvelteKit aplikasi untuk Amplify Hosting

Menerapkan aplikasi Next.js untuk Amplify Hosting

Tutorial ini memandu Anda melalui membangun dan menyebarkan aplikasi Next.js dari repositori Git.

Sebelum Anda memulai tutorial ini, lengkapi prasyarat berikut.

Mendaftar untuk Akun AWS

Jika Anda belum menjadi AWS pelanggan, Anda perlu membuat Akun AWS dengan mengikuti instruksi online. Mendaftar memungkinkan Anda mengakses Amplify dan AWS layanan lain yang dapat Anda gunakan dengan aplikasi Anda.

Membuat aplikasi

Buat aplikasi Next.js dasar untuk digunakan untuk tutorial ini, menggunakan create-next-appinstruksi dalam dokumentasi Next.js.

Buat repositori Git

Amplify mendukung GitHub, Bitbucket,, GitLab dan. AWS CodeCommit Dorong `create-next-app` aplikasi Anda ke repositori Git Anda.

Langkah 1: Hubungkan repositori Git

Pada langkah ini, Anda menghubungkan aplikasi Next.js Anda di repositori Git ke Amplify Hosting.

Untuk menghubungkan aplikasi di repositori Git

1. Buka konsol Amplify.
2. Jika Anda menerapkan aplikasi pertama Anda di Wilayah saat ini, secara default Anda akan mulai dari halaman AWS Amplifylayanan.

   Pilih Menerapkan aplikasi di bagian atas halaman.
3. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.

   Untuk GitHub repositori, Amplify menggunakan fitur GitHub Apps untuk mengotorisasi akses Amplify. Untuk informasi selengkapnya tentang menginstal dan mengotorisasi GitHub Aplikasi, lihatMenyiapkan akses Amplify ke repositori GitHub.

   > **Note**
   >
   > Setelah Anda mengotorisasi konsol Amplify dengan Bitbucket GitLab,, atau AWS CodeCommit, Amplify mengambil token akses dari penyedia repositori, tetapi token tersebut tidak menyimpan token di server. AWS Amplify mengakses repositori Anda menggunakan kunci deploy yang terinstal di repositori tertentu saja.
4. Pada halaman Add repository branch lakukan hal berikut:
   a. Pilih nama repositori untuk terhubung.
   b. Pilih nama cabang repositori untuk terhubung.
   c. Pilih Berikutnya.

Langkah 2: Konfirmasikan pengaturan build

Amplify secara otomatis mendeteksi urutan perintah build yang akan dijalankan untuk cabang yang Anda terapkan. Pada langkah ini Anda meninjau dan mengonfirmasi setelan build Anda.

Untuk mengonfirmasi setelan build untuk aplikasi

1. Di halaman Pengaturan aplikasi, cari bagian Pengaturan build.

   Verifikasi bahwa perintah build Frontend dan direktori keluaran Build sudah benar. Untuk aplikasi contoh Next.js ini, direktori keluaran Build disetel ke`.next`.

2. Prosedur untuk menambahkan peran layanan bervariasi tergantung pada apakah Anda ingin membuat peran baru atau menggunakan peran yang sudah ada.
   - Untuk membuat peran baru:
     - Pilih Buat dan gunakan peran layanan baru.
   - Untuk menggunakan peran yang ada:
     a. Pilih Gunakan peran yang ada.
     b. Dalam daftar peran layanan, pilih peran yang akan digunakan.
3. Pilih Berikutnya.

Langkah 3: Deploy aplikasi

Pada langkah ini, Anda menerapkan aplikasi ke jaringan pengiriman konten AWS global (CDN).

Untuk menyimpan dan menerapkan aplikasi

1. Pada halaman Tinjauan, konfirmasikan bahwa detail repositori dan pengaturan aplikasi Anda sudah benar.
2. Pilih Simpan dan deploy. Front end build Anda biasanya membutuhkan waktu 1 hingga 2 menit, tetapi dapat bervariasi berdasarkan ukuran aplikasi.
3. Saat penerapan selesai, Anda dapat melihat aplikasi menggunakan tautan ke domain `amplifyapp.com` default.

> **Note**
>
> Untuk meningkatkan keamanan aplikasi Amplify Anda, domain amplifyapp.com terdaftar di Daftar Akhiran Publik (PSL). Untuk keamanan lebih lanjut, kami menyarankan Anda menggunakan cookie dengan `__Host-` awalan jika Anda perlu mengatur cookie sensitif di nama domain default untuk aplikasi Amplify Anda. Praktik ini akan membantu mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat halaman Set-Cookie di Jaringan Pengembang Mozilla.

Langkah 4: (Opsional) membersihkan sumber daya

Jika Anda tidak lagi membutuhkan aplikasi yang Anda gunakan untuk tutorial, Anda dapat menghapusnya. Langkah ini membantu memastikan bahwa Anda tidak akan dikenakan biaya untuk sumber daya yang tidak Anda gunakan.

Untuk menghapus aplikasi

1. Dari menu Pengaturan aplikasi di panel navigasi, pilih Pengaturan umum.
2. Pada halaman Pengaturan umum, pilih Hapus aplikasi.
3. Di jendela konfirmasi, masukkan**`delete`**. Kemudian pilih Hapus aplikasi.

Menambahkan fitur ke aplikasi

Sekarang setelah aplikasi disebarkan ke Amplify, Anda dapat menjelajahi beberapa fitur berikut yang tersedia untuk aplikasi yang dihosting.

Variabel-variabel lingkungan

Aplikasi sering membutuhkan informasi konfigurasi saat runtime. Konfigurasi ini dapat berupa detail koneksi database, kunci API, atau parameter. Variabel lingkungan menyediakan cara untuk mengekspos konfigurasi ini pada waktu pembuatan. Untuk informasi selengkapnya, lihat Variabel lingkungan.

Domain kustom

Dalam tutorial ini, Amplify meng-host aplikasi Anda untuk Anda di `amplifyapp.com` domain default dengan URL seperti. `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com` Saat Anda menghubungkan aplikasi ke domain khusus, pengguna akan melihat bahwa aplikasi Anda di-host di URL khusus, misalnya`https://www.example.com`. Untuk informasi selengkapnya, lihat Menyiapkan domain kustom.

Tarik pratinjau permintaan

Pratinjau permintaan tarik web menawarkan tim cara untuk melihat pratinjau perubahan dari permintaan tarik (PRs) sebelum menggabungkan kode ke cabang produksi atau integrasi. Untuk informasi selengkapnya, lihat pratinjau Web untuk permintaan tarik.

Mengelola beberapa lingkungan

Untuk mempelajari cara Amplify bekerja dengan cabang fitur dan GitFlow alur kerja untuk mendukung beberapa penerapan, lihat Penerapan cabang fitur dan alur kerja tim.

Menerapkan aplikasi Nuxt.js untuk Amplify Hosting

Gunakan petunjuk berikut untuk menyebarkan aplikasi Nuxt.js ke Amplify Hosting. Nuxt telah menerapkan adaptor preset menggunakan server Nitro. Ini memungkinkan Anda untuk menerapkan proyek Nuxt tanpa konfigurasi tambahan apa pun.

Untuk menerapkan aplikasi Nuxt ke Amplify Hosting

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.
4. Di halaman Tambahkan cabang repositori, lakukan langkah berikut:
   a. Pilih nama repositori untuk terhubung.
   b. Pilih nama cabang repositori untuk terhubung.
   c. Pilih Berikutnya.
5. Jika Anda ingin Amplify dapat mengirimkan log aplikasi ke Amazon CloudWatch Logs, Anda harus mengaktifkannya secara eksplisit di konsol. Buka bagian Pengaturan lanjutan, lalu pilih Aktifkan log aplikasi SSR di bagian penyebaran Server-Side Rendering (SSR).
6. Pilih Berikutnya.
7. Di halaman Tinjauan, pilih Simpan dan deploy.

Menerapkan aplikasi Astro.js untuk Amplify Hosting

Gunakan petunjuk berikut untuk menyebarkan aplikasi Astro.js ke Amplify Hosting. Anda dapat menggunakan aplikasi yang sudah ada, atau membuat aplikasi starter menggunakan salah satu contoh resmi yang disediakan Astro. Untuk membuat aplikasi pemula, lihat Menggunakan tema atau template starter dalam dokumentasi Astro.

Untuk menyebarkan situs Astro dengan SSR ke Amplify Hosting, Anda harus menambahkan adaptor ke aplikasi Anda. Kami tidak memelihara adaptor milik Amplify untuk kerangka kerja Astro. Tutorial ini menggunakan `astro-aws-amplify` adaptor yang dibuat oleh anggota komunitas. Adaptor ini tersedia di github. com/alexnguyennz/astro-aws-amplify di situs web. GitHub AWS tidak mempertahankan adaptor ini.

Untuk menerapkan aplikasi Astro ke Amplify Hosting

1. Di komputer lokal Anda, navigasikan ke aplikasi Astro untuk menyebarkan.
2. Untuk menginstal adaptor, buka jendela terminal dan jalankan perintah berikut. Contoh ini menggunakan adaptor komunitas yang tersedia di github. com/alexnguyennz/astro-aws-amplify. Anda dapat mengganti *`astro-aws-amplify`* dengan nama adaptor yang Anda gunakan.

   ```
   npm install astro-aws-amplify
   ```

3. Di folder proyek untuk aplikasi Astro Anda, buka `astro.config.mjs` file. Perbarui file untuk menambahkan adaptor. File akan terlihat seperti berikut.

   ```
   import { defineConfig } from 'astro/config';
   import mdx from '@astrojs/mdx';
   import awsAmplify from 'astro-aws-amplify';

   import sitemap from '@astrojs/sitemap';

   // https://astro.build/config
   export default defineConfig({
     site: 'https://example.com',
     integrations: [mdx(), sitemap()],
     adapter: awsAmplify(),
     output: 'server',
   });
   ```

4. Komit perubahan dan dorong proyek ke repositori Git Anda.

   Sekarang Anda siap untuk menerapkan aplikasi Astro Anda ke Amplify.

5. Masuk ke AWS Management Console dan buka konsol Amplify.
6. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
7. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.

8. Di halaman Tambahkan cabang repositori, lakukan langkah berikut:

   a. Pilih nama repositori untuk terhubung.

   b. Pilih nama cabang repositori untuk terhubung.

   c. Pilih Berikutnya.

9. Di halaman Pengaturan aplikasi, cari bagian Pengaturan build. Untuk direktori keluaran Build, masukkan **`.amplify-hosting`**.

10. Anda juga harus memperbarui perintah build frontend aplikasi dalam spesifikasi build. Untuk membuka spesifikasi build, pilih Edit file YHTML.

11. Dalam `amplify.yml` file, cari bagian perintah build frontend. Masuk **`mv node_modules ./.amplify-hosting/compute/default`**

    File pengaturan build Anda akan terlihat seperti berikut ini.

```
version: 1
frontend:
    phases:
        preBuild:
            commands:
                - 'npm ci --cache .npm --prefer-offline'
        build:
            commands:
                - 'npm run build'
                - 'mv node_modules ./.amplify-hosting/compute/default'
    artifacts:
        baseDirectory: .amplify-hosting
        files:
            - '**/*'
    cache:
        paths:
            - '.npm/**/*'
```

12. Pilih Simpan.

13. Jika Anda ingin Amplify dapat mengirimkan log aplikasi ke Amazon CloudWatch Logs, Anda harus mengaktifkannya secara eksplisit di konsol. Buka bagian Pengaturan lanjutan, lalu pilih Aktifkan log aplikasi SSR di bagian penyebaran Server-Side Rendering (SSR).

14. Pilih Berikutnya.

15. Di halaman Tinjauan, pilih Simpan dan deploy.

Menerapkan SvelteKit aplikasi untuk Amplify Hosting

Gunakan petunjuk berikut untuk menyebarkan SvelteKit aplikasi ke Amplify Hosting. Anda dapat menggunakan aplikasi Anda sendiri, atau membuat aplikasi pemula. Untuk informasi selengkapnya, lihat Membuat proyek dalam SvelteKit dokumentasi.

Untuk menerapkan SvelteKit aplikasi dengan SSR ke Amplify Hosting, Anda harus menambahkan adaptor ke proyek Anda. Kami tidak memelihara adaptor yang dimiliki Amplify untuk kerangka kerja. SvelteKit Dalam contoh ini, kami menggunakan yang `amplify-adapter` dibuat oleh anggota komunitas. Adaptor tersedia di github. com/gzimbron/amplify-adaptor di GitHub situs web. AWS tidak mempertahankan adaptor ini.

Untuk menerapkan SvelteKit aplikasi ke Amplify Hosting

1. Di komputer lokal Anda, navigasikan ke SvelteKit aplikasi untuk menyebarkan.
2. Untuk menginstal adaptor, buka jendela terminal dan jalankan perintah berikut. Contoh ini menggunakan adaptor komunitas yang tersedia di github. com/gzimbron/amplify-adaptor. Jika Anda menggunakan adaptor komunitas yang berbeda, ganti *amplify-adapter* dengan nama adaptor Anda.

   ```
   npm install amplify-adapter
   ```

3. Di folder project untuk SvelteKit aplikasi Anda, buka `svelte.config.js` file tersebut. Edit file untuk menggunakan `amplify-adapter` atau mengganti *'amplify-adapter'* dengan nama adaptor Anda. File akan terlihat seperti berikut.

   ```
   import adapter from 'amplify-adapter';
   import { vitePreprocess } from '@sveltejs/vite-plugin-svelte';


   /** @type {import('@sveltejs/kit').Config} */
   const config = {
           // Consult https://kit.svelte.dev/docs/integrations#preprocessors
           // for more information about preprocessors
           preprocess: vitePreprocess(),

           kit: {
                   // adapter-auto only supports some environments, see https://
   kit.svelte.dev/docs/adapter-auto for a list.
   ```

```
                // If your environment is not supported, or you settled on a
 specific environment, switch out the adapter.
                // See https://kit.svelte.dev/docs/adapters for more information
 about adapters.
                adapter: adapter()
        }
};

export default config;
```

4. Komit perubahan dan dorong aplikasi ke repositori Git Anda.
5. Sekarang Anda siap untuk menerapkan SvelteKit aplikasi Anda ke Amplify.

   Masuk ke AWS Management Console dan buka konsol Amplify.
6. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
7. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.
8. Di halaman Tambahkan cabang repositori, lakukan langkah berikut:

   a. Pilih nama repositori untuk terhubung.

   b. Pilih nama cabang repositori untuk terhubung.

   c. Pilih Berikutnya.
9. Di halaman Pengaturan aplikasi, cari bagian Pengaturan build. Untuk direktori keluaran Build, masukkan**`build`**.
10. Anda juga harus memperbarui perintah build frontend aplikasi dalam spesifikasi build. Untuk membuka spesifikasi build, pilih Edit file YHTML.
11. Dalam `amplify.yml` file, cari bagian perintah build frontend. Masuk **`- cd build/compute/default/`** dan**`- npm i --production`**.

    File pengaturan build Anda akan terlihat seperti berikut ini.

```
version: 1
frontend:
    phases:
        preBuild:
            commands:
                - 'npm ci --cache .npm --prefer-offline'
        build:
            commands:
```

```
                - 'npm run build'
                - 'cd build/compute/default/'
                - 'npm i --production'

    artifacts:
        baseDirectory: build
        files:
            - '**/*'
    cache:
        paths:
            - '.npm/**/*'
```

12. Pilih Simpan.
13. Jika Anda ingin Amplify dapat mengirimkan log aplikasi ke Amazon CloudWatch Logs, Anda harus mengaktifkannya secara eksplisit di konsol. Buka bagian Pengaturan lanjutan, lalu pilih Aktifkan log aplikasi SSR di bagian penyebaran Server-Side Rendering (SSR).
14. Pilih Berikutnya.
15. Di halaman Tinjauan, pilih Simpan dan deploy.

Menyebarkan aplikasi yang dirender sisi server dengan Amplify Hosting

Anda dapat menggunakan AWS Amplify untuk menyebarkan dan meng-host aplikasi web yang menggunakan rendering sisi server (SSR). Amplify Hosting secara otomatis mendeteksi aplikasi yang dibuat menggunakan framework Next.js dan Anda tidak perlu melakukan konfigurasi manual apa pun di file. AWS Management Console

Amplify juga mendukung framework SSR berbasis Javascript dengan adaptor build open source yang mengubah output build aplikasi menjadi struktur direktori yang diharapkan Amplify Hosting. Misalnya, Anda dapat menerapkan aplikasi yang dibuat dengan Nuxt, Astro, dan SvelteKit framework dengan menginstal adaptor yang tersedia.

Pengguna tingkat lanjut dapat menggunakan spesifikasi penerapan untuk membuat adaptor build atau mengonfigurasi skrip pasca-build.

Anda dapat menerapkan kerangka kerja berikut untuk Amplify Hosting dengan konfigurasi minimal.

Next.js

- Amplify mendukung aplikasi Next.js 15 tanpa perlu adaptor. Untuk memulai, lihat Amplify dukungan untuk Next.js.

Nuxt.js

- Amplify mendukung penerapan aplikasi Nuxt.js dengan adaptor preset. Untuk memulai, lihat Amplify dukungan untuk Nuxt.js.

Astro.js

- Amplify mendukung penerapan aplikasi Astro.js dengan adaptor komunitas. Untuk memulai, lihat Amplify dukungan untuk Astro.js.

SvelteKit

- Amplify mendukung SvelteKit penerapan aplikasi dengan adaptor komunitas. Untuk memulai, lihat Amplify dukungan untuk SvelteKit.

Adaptor sumber terbuka

- Gunakan adaptor sumber terbuka - Untuk petunjuk tentang penggunaan adaptor apa pun yang tidak ada dalam daftar sebelumnya, lihat. Menggunakan adaptor open source untuk kerangka SSR apa pun

- Buat adaptor kerangka kerja - Pembuat kerangka kerja yang ingin mengintegrasikan fitur yang disediakan oleh framework, dapat menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build agar sesuai dengan struktur yang diharapkan Amplify. Untuk informasi selengkapnya, lihat Menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build.
- Mengonfigurasi skrip pasca-build - Anda dapat menggunakan spesifikasi penerapan Amplify Hosting untuk memanipulasi keluaran build sesuai kebutuhan untuk skenario tertentu. Untuk informasi selengkapnya, lihat Menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build. Sebagai contoh, lihat Menyebarkan server Express menggunakan manifes penerapan.

Topik

Amplify dukungan untuk Next.js

Amplify mendukung penerapan dan hosting untuk aplikasi web yang dirender sisi server (SSR) yang dibuat menggunakan Next.js. Next.js adalah kerangka kerja React untuk mengembangkan SPAs dengan JavaScript. Anda dapat menerapkan aplikasi yang dibangun dengan versi Next.js hingga Next.js 15, dengan fitur seperti optimasi gambar dan middleware.

Pengembang dapat menggunakan Next.js untuk menggabungkan pembuatan situs statis (SSG), dan SSR dalam satu proyek. Halaman SSG dirender sebelumnya pada waktu build, dan halaman SSR dirender sebelumnya pada waktu permintaan.

Prerendering dapat meningkatkan performa dan optimasi mesin pencari. Karena Next.js melakukan prerendering pada semua halaman di server, konten HTML setiap halaman siap ketika mencapai

peramban klien. Konten ini juga dapat memuat lebih cepat. Waktu unggah yang lebih cepat meningkatkan pengalaman pengguna akhir saat menggunakan situs web dan berdampak positif pada peringkat SEO situs tersebut. Prerendering juga meningkatkan SEO dengan memungkinkan bot mesin pencari untuk menemukan dan merayapi konten HTML situs web dengan mudah.

Next.js menyediakan dukungan analitik bawaan untuk mengukur berbagai metrik kinerja, seperti Time to first byte (TTFB) dan First contentful paint (FCP). Untuk informasi lebih lanjut tentang Next.js, lihat Memulai di situs web Next.js.

Dukungan fitur Next.js

Amplify Hosting compute sepenuhnya mengelola rendering sisi server (SSR) untuk aplikasi yang dibangun dengan Next.js versi 12 hingga 15.

Jika Anda menerapkan aplikasi Next.js ke Amplify sebelum rilis komputasi Amplify Hosting pada November 2022, aplikasi Anda menggunakan penyedia SSR Amplify sebelumnya, Classic (khusus Next.js 11). Amplify Hosting compute tidak mendukung aplikasi yang dibuat menggunakan Next.js versi 11 atau yang lebih lama. Kami sangat menyarankan agar Anda memigrasikan aplikasi Next.js 11 Anda ke penyedia SSR terkelola komputasi Amplify Hosting.

Daftar berikut menjelaskan fitur spesifik yang didukung oleh penyedia SSR komputasi Amplify Hosting.

Fitur yang didukung

- Halaman yang dirender sisi server (SSR)
- Halaman statis
- Rute API
- Rute dinamis
- Tangkap semua rute
- SSG (Generasi statis)
- Regenerasi Statis Inkremental (ISR)
- Perutean sub-jalur internasional (i18n)
- Perutean domain internasional (i18n)
- Deteksi lokal otomatis yang diinternasionalisasi (i18n)
- Middleware
- Variabel-variabel lingkungan

- Optimalisasi gambar
- Next.js 13 direktori aplikasi

Fitur yang tidak didukung

- Rute API Edge (Middleware tepi tidak didukung)
- Regenerasi Statis Inkremental Sesuai Permintaan (ISR)
- Next.js Streaming
- Menjalankan middleware pada aset statis dan gambar yang dioptimalkan
- Mengeksekusi kode setelah respons dengan `unstable_after` (Fitur eksperimental dirilis dengan Next.js 15)

Gambar Next.js

Ukuran output maksimum gambar tidak boleh melebihi 4,3 MB. Anda dapat menyimpan file gambar yang lebih besar di suatu tempat dan menggunakan komponen Gambar Next.js untuk mengubah ukuran dan mengoptimalkannya ke dalam format Webp atau AVIF dan kemudian menyajikannya sebagai ukuran yang lebih kecil.

Perhatikan bahwa dokumentasi Next.js menyarankan Anda untuk menginstal modul pemrosesan gambar Sharp agar pengoptimalan gambar berfungsi dengan benar dalam produksi. Namun, ini tidak diperlukan untuk penerapan Amplify. Amplify secara otomatis menyebarkan Sharp untuk Anda.

Menerapkan aplikasi SSR Next.js untuk Amplify

Secara default, Amplify menyebarkan aplikasi SSR baru menggunakan layanan komputasi Amplify Hosting dengan dukungan untuk Next.js versi 12 hingga 15. Amplify Hosting compute sepenuhnya mengelola sumber daya yang diperlukan untuk menerapkan aplikasi SSR. Aplikasi SSR di akun Amplify yang Anda terapkan sebelum 17 November 2022 menggunakan penyedia SSR Klasik (khusus Next.js 11).

Kami sangat menyarankan Anda memigrasikan aplikasi menggunakan SSR Klasik (hanya Next.js 11) ke penyedia SSR komputasi Amplify Hosting. Amplify tidak melakukan migrasi otomatis untuk Anda. Anda harus memigrasikan aplikasi secara manual dan kemudian memulai build baru untuk menyelesaikan pembaruan. Untuk petunjuk, lihat Migrasi aplikasi SSR Next.js 11 ke komputasi Amplify Hosting.

Gunakan petunjuk berikut untuk menerapkan aplikasi SSR Next.js baru.

Untuk menerapkan aplikasi SSR ke Amplify menggunakan penyedia SSR komputasi Amplify Hosting

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.
4. Di halaman Tambahkan cabang repositori, lakukan langkah berikut:
   a. Dalam daftar repositori yang baru diperbarui, pilih nama repositori yang akan dihubungkan.
   b. Dalam daftar Branch, pilih nama cabang repositori untuk terhubung.
   c. Pilih Berikutnya.
5. Aplikasi memerlukan peran layanan IAM yang diasumsikan Amplify saat memanggil layanan lain atas nama Anda. Anda dapat mengizinkan komputasi Amplify Hosting untuk secara otomatis membuat peran layanan untuk Anda atau Anda dapat menentukan peran yang telah Anda buat.
   - Untuk mengizinkan Amplify membuat peran secara otomatis dan melampirkannya ke aplikasi Anda:
     - Pilih Buat dan gunakan peran layanan baru.
   - Untuk melampirkan peran layanan yang sebelumnya Anda buat:
     a. Pilih Gunakan peran layanan yang ada.
     b. Pilih peran yang akan digunakan dari daftar.
6. Pilih Berikutnya.
7. Di halaman Tinjauan, pilih Simpan dan deploy.

Pengaturan file Package.json

Saat Anda menerapkan aplikasi Next.js, Amplify akan memeriksa skrip build aplikasi dalam `package.json` file untuk menentukan jenis aplikasi.

Berikut ini adalah contoh skrip build untuk aplikasi Next.js. Skrip build `"next build"` menunjukkan bahwa aplikasi mendukung halaman SSG dan SSR. Skrip build ini juga digunakan untuk aplikasi SSG Next.js 14 atau yang lebih baru.

```
"scripts": {
```

```
  "dev": "next dev",
  "build": "next build",
  "start": "next start"
},
```

Berikut ini adalah contoh skrip build untuk Next.js 13 atau aplikasi SSG sebelumnya. Skrip build `"next build && next export"` menunjukkan bahwa aplikasi mendukung halaman SSG saja.

```
"scripts": {
  "dev": "next dev",
  "build": "next build && next export",
  "start": "next start"
},
```

Amplify pengaturan build untuk aplikasi SSR Next.js

Setelah memeriksa `package.json` file aplikasi Anda, Amplify memeriksa setelan build untuk aplikasi. Anda dapat menyimpan pengaturan build di konsol Amplify atau di file `amplify.yml` di root repositori Anda. Untuk informasi lebih lanjut, lihat Mengonfigurasi pengaturan build untuk aplikasi Amplify.

Jika Amplify mendeteksi bahwa Anda men-deploy aplikasi Next.js SSR, dan tidak ada file `amplify.yml`, Amplify akan membuat buildspec untuk aplikasi dan mengatur `baseDirectory` ke `.next`. Jika Anda men-deploy aplikasi berisi file `amplify.yml`, pengaturan build di file akan menimpa pengaturan build di konsol. Oleh karena itu, Anda harus secara manual mengatur `baseDirectory` ke `.next` di file.

Berikut contoh pengaturan build untuk aplikasi dengan `baseDirectory` diatur ke `.next`. Artinya, artefak build ditujukan untuk aplikasi Next.js yang mendukung halaman SSG dan SSR.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
```

```
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Amplify pengaturan build untuk Next.js 13 atau aplikasi SSG sebelumnya

Jika Amplify mendeteksi bahwa Anda sedang menerapkan Next.js 13 atau aplikasi SSG sebelumnya, Amplify akan menghasilkan spesifikasi build untuk aplikasi dan disetel ke. `baseDirectory` `out` Jika men-deploy aplikasi berisi file `amplify.yml`, Anda harus secara manual mengatur `baseDirectory` ke `out` di file. `out`Direktori adalah folder default yang dibuat Next.js untuk menyimpan aset statis yang diekspor. Saat mengonfigurasi setelan spesifikasi build aplikasi, ubah nama `baseDirectory` folder agar sesuai dengan konfigurasi aplikasi.

Berikut ini adalah contoh setelan build untuk aplikasi yang `baseDirectory` disetel `out` untuk menunjukkan bahwa artefak build adalah untuk Next.js 13 atau aplikasi sebelumnya yang hanya mendukung halaman SSG.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: out
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Amplify setelan build untuk aplikasi SSG Next.js 14 atau yang lebih baru

Di Next.js versi 14, `next export` perintah tidak digunakan lagi dan diganti dengan `output: 'export'` `next.config.js` file untuk mengaktifkan ekspor statis. Jika Anda menerapkan aplikasi Next.js 14 SSG saja di konsol, Amplify akan menghasilkan buildspec untuk aplikasi dan disetel ke.

`baseDirectory` `.next` Jika men-deploy aplikasi berisi file `amplify.yml`, Anda harus secara manual mengatur `baseDirectory` ke `.next` di file. Ini adalah `baseDirectory` pengaturan yang sama yang Amplify gunakan untuk `WEB_COMPUTE` aplikasi Next.js yang mendukung halaman SSG dan SSR.

Berikut ini adalah contoh pengaturan build untuk aplikasi Next.js 14 SSG saja dengan `baseDirectory` set ke`.next`.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Migrasi aplikasi SSR Next.js 11 ke komputasi Amplify Hosting

Saat Anda menerapkan aplikasi Next.js baru, secara default Amplify menggunakan versi Next.js terbaru yang didukung. Saat ini, penyedia SSR komputasi Amplify Hosting mendukung Next.js versi 15.

Konsol Amplify mendeteksi aplikasi di akun Anda yang digunakan sebelum rilis November 2022 dari layanan komputasi Amplify Hosting dengan dukungan penuh untuk Next.js versi 12 hingga 15. Konsol menampilkan spanduk informasi yang mengidentifikasi aplikasi dengan cabang yang digunakan menggunakan penyedia SSR Amplify sebelumnya, Classic (hanya Next.js 11). Kami sangat menyarankan Anda memigrasikan aplikasi Anda ke penyedia SSR komputasi Amplify Hosting.

Jika Anda memperbarui aplikasi Next.js 11 yang dihosting ke Next.js 12 atau yang lebih baru, Anda mungkin mendapatkan `"target" property is no longer supported` kesalahan saat penerapan dipicu. Dalam hal ini, Anda harus bermigrasi ke komputasi Amplify Hosting.

Anda harus memigrasikan aplikasi secara manual dan semua cabang produksinya secara bersamaan. Aplikasi tidak dapat berisi cabang Classic (Next.js 11 saja) dan Next.js 12 atau yang lebih baru.

Gunakan petunjuk berikut untuk memigrasikan aplikasi ke penyedia SSR komputasi Amplify Hosting.

Untuk memigrasikan aplikasi ke penyedia SSR komputasi Amplify Hosting

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi Next.js yang ingin Anda migrasikan.

   > **Note**
   >
   > Sebelum memigrasikan aplikasi di konsol Amplify, Anda harus memperbarui file package.json aplikasi terlebih dahulu untuk menggunakan Next.js versi 12 atau yang lebih baru.

3. Di panel navigasi, pilih Pengaturan aplikasi, Umum.
4. Di beranda aplikasi, konsol menampilkan spanduk jika aplikasi memiliki cabang yang digunakan menggunakan penyedia SSR Klasik (hanya Next.js 11). Pada banner, pilih Migrate.
5. Di jendela konfirmasi migrasi, pilih tiga pernyataan dan pilih Migrasi.
6. Amplify akan membangun dan menerapkan ulang aplikasi Anda untuk menyelesaikan migrasi.

Mengembalikan migrasi SSR

Saat Anda menerapkan aplikasi Next.js, Amplify Hosting mendeteksi setelan di aplikasi Anda dan menetapkan nilai platform internal untuk aplikasi tersebut. Ada tiga nilai platform yang valid. Aplikasi SSG diatur ke nilai `WEB` platform. Aplikasi SSR yang menggunakan Next.js versi 11 diatur ke nilai `WEB_DYNAMIC` platform. Aplikasi SSR Next.js 12 atau yang lebih baru disetel ke nilai `WEB_COMPUTE` platform.

Saat Anda memigrasikan aplikasi menggunakan petunjuk di bagian sebelumnya, Amplify mengubah nilai platform aplikasi `WEB_DYNAMIC` Anda dari ke. `WEB_COMPUTE` Setelah migrasi ke Amplify Hosting komputasi selesai, Anda tidak dapat mengembalikan migrasi di konsol. Untuk mengembalikan migrasi, Anda harus menggunakan file AWS Command Line Interface untuk mengubah platform aplikasi kembali`WEB_DYNAMIC`. Buka jendela terminal dan masukkan perintah berikut, perbarui ID aplikasi dan Wilayah dengan informasi unik Anda.

```
aws amplify update-app --app-id abcd1234 --platform WEB_DYNAMIC --region us-west-2
```

Menambahkan fungsionalitas SSR ke aplikasi Next.js statis

Anda dapat menambahkan fungsionalitas SSR ke aplikasi Next.js statis (SSG) yang ada yang digunakan dengan Amplify. Sebelum Anda memulai proses konversi aplikasi SSG Anda ke SSR, perbarui aplikasi untuk menggunakan Next.js versi 12 atau yang lebih baru dan tambahkan fungsionalitas SSR. Maka Anda perlu melakukan langkah-langkah berikut.

1. Gunakan AWS Command Line Interface untuk mengubah jenis platform aplikasi.
2. Tambahkan peran layanan ke aplikasi.
3. Perbarui direktori keluaran di setelan build aplikasi.
4. Perbarui `package.json` file aplikasi untuk menunjukkan bahwa aplikasi menggunakan SSR.

Memperbarui platform

Ada tiga nilai yang valid untuk tipe platform. Aplikasi SSG diatur ke jenis `WEB` platform. Aplikasi SSR yang menggunakan Next.js versi 11 diatur ke jenis `WEB_DYNAMIC` platform. Untuk aplikasi yang diterapkan ke Next.js 12 atau yang lebih baru menggunakan SSR yang dikelola oleh komputasi Amplify Hosting, jenis platform disetel ke. `WEB_COMPUTE`

Saat Anda menerapkan aplikasi sebagai aplikasi SSG, Amplify menyetel jenis platform ke. `WEB` Gunakan AWS CLI untuk mengubah platform aplikasi Anda`WEB_COMPUTE`. Buka jendela terminal dan masukkan perintah berikut, perbarui teks berwarna merah dengan id aplikasi dan Wilayah unik Anda.

```
aws amplify update-app --app-id abcd1234 --platform WEB_COMPUTE --region us-west-2
```

Menambahkan peran layanan

Peran layanan adalah peran AWS Identity and Access Management (IAM) yang diasumsikan Amplify saat memanggil layanan lain atas nama Anda. Ikuti langkah-langkah berikut untuk menambahkan peran layanan ke aplikasi SSG yang sudah digunakan dengan Amplify.

Cara Menambahkan peran layanan

1. Masuk ke AWS Management Console dan buka konsol Amplify.

2. Jika Anda belum membuat peran layanan di akun Amplify, lihat Menambahkan peran layanan untuk menyelesaikan langkah prasyarat ini.
3. Pilih aplikasi Next.js statis tempat peran layanan akan ditambahkan.
4. Di panel navigasi, pilih Pengaturan aplikasi, Umum.
5. Di halaman Detail aplikasi, pilih Edit
6. Untuk Peran layanan, pilih nama peran layanan yang ada atau nama peran layanan yang Anda buat di langkah 2.
7. Pilih Simpan.

Memperbarui setelan build

Sebelum men-deploy ulang aplikasi dengan fungsionalitas SSR, Anda harus memperbarui pengaturan build untuk aplikasi guna mengatur direktori output ke `.next`. Anda dapat mengedit pengaturan build di konsol Amplify atau di file `amplify.yml` yang disimpan di repo Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi pengaturan build untuk aplikasi Amplify.

Berikut contoh pengaturan build untuk aplikasi dengan `baseDirectory` diatur ke `.next`.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Memperbarui file package.json

Setelah Anda menambahkan peran layanan dan memperbarui pengaturan build, perbarui file `package.json` aplikasi. Seperti dalam contoh berikut, atur skrip build ke `"next build"` untuk menunjukkan bahwa aplikasi Next.js mendukung halaman SSG dan SSR.

```
"scripts": {
  "dev": "next dev",
  "build": "next build",
  "start": "next start"
},
```

Amplify mendeteksi perubahan pada file `package.json` di repo Anda dan men-deploy ulang aplikasi dengan fungsionalitas SSR.

Membuat variabel lingkungan dapat diakses oleh runtime sisi server

Amplify Hosting mendukung penambahan variabel lingkungan ke build aplikasi Anda dengan menyetelnya dalam konfigurasi project di konsol Amplify.

Namun, komponen server Next.js tidak memiliki akses ke variabel lingkungan tersebut secara default. Perilaku ini disengaja untuk melindungi setiap rahasia yang disimpan dalam variabel lingkungan yang digunakan aplikasi Anda selama fase build.

Untuk membuat variabel lingkungan tertentu dapat diakses oleh Next.js, Anda dapat memodifikasi file spesifikasi build Amplify untuk mengaturnya dalam file lingkungan yang dikenali Next.js. Ini memungkinkan Amplify memuat variabel lingkungan ini sebelum membangun aplikasi.

> **Important**
>
> Kami sangat menyarankan agar Anda tidak menyimpan kredensi, rahasia, atau informasi sensitif apa pun dalam variabel lingkungan Anda karena pengguna mana pun yang memiliki akses ke artefak penerapan dapat membacanya.
> Untuk memberikan akses fungsi komputasi SSR Anda ke AWS sumber daya, sebaiknya gunakan peran IAM.

Contoh spesifikasi build berikut menunjukkan cara menambahkan variabel lingkungan di bagian perintah build.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - env | grep -e API_BASE_URL >> .env.production
        - env | grep -e NEXT_PUBLIC_ >> .env.production
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
      - .next/cache/**/*
```

Dalam contoh ini, bagian perintah build menyertakan dua perintah yang menulis variabel lingkungan ke `.env.production` file sebelum build aplikasi berjalan. Amplify Hosting memungkinkan aplikasi Anda mengakses variabel-variabel ini saat aplikasi menerima lalu lintas.

Baris berikut dari bagian perintah build pada contoh sebelumnya menunjukkan cara mengambil variabel tertentu dari lingkungan build dan menambahkannya ke file. `.env.production`

```
- env | grep -e API_BASE_URL -e APP_ENV >> .env.production
```

Jika variabel ada di lingkungan build Anda, `.env.production` file akan berisi variabel lingkungan berikut.

```
API_BASE_URL=localhost
APP_ENV=dev
```

Baris berikut dari bagian perintah build pada contoh sebelumnya menunjukkan cara menambahkan variabel lingkungan dengan awalan tertentu ke file. `.env.production` Dalam contoh ini, semua variabel dengan awalan `NEXT_PUBLIC_` ditambahkan.

```
- env | grep -e NEXT_PUBLIC_ >> .env.production
```

Jika beberapa variabel dengan `NEXT_PUBLIC_` awalan ada di lingkungan build, `.env.production` file akan terlihat mirip dengan berikut ini.

```
NEXT_PUBLIC_ANALYTICS_ID=abcdefghijk
NEXT_PUBLIC_GRAPHQL_ENDPOINT=uowelalsmlsadf
NEXT_PUBLIC_FEATURE_FLAG=true
```

Variabel lingkungan SSR untuk monorepos

Jika Anda menerapkan aplikasi SSR di monorepo dan ingin membuat variabel lingkungan tertentu dapat diakses oleh Next.js, Anda harus mengawali `.env.production` file dengan root aplikasi Anda. Contoh spesifikasi build berikut untuk aplikasi Next.js dalam monorepo Nx menunjukkan cara menambahkan variabel lingkungan di bagian perintah build.

```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
          commands:
            - npm ci
        build:
          commands:
            - env | grep -e API_BASE_URL -e APP_ENV >> apps/app/.env.production
            - env | grep -e NEXT_PUBLIC_ >> apps/app/.env.production
            - npx nx build app
      artifacts:
        baseDirectory: dist/apps/app/.next
        files:
          - '**/*'
      cache:
        paths:
          - node_modules/**/*
      buildPath: /
    appRoot: apps/app
```

Baris berikut dari bagian perintah build pada contoh sebelumnya menunjukkan cara mengambil variabel tertentu dari lingkungan build dan menambahkannya ke `.env.production` file untuk aplikasi dalam monorepo dengan root aplikasi. `apps/app`

```
- env | grep -e API_BASE_URL -e APP_ENV >> apps/app/.env.production
```

```
- env | grep -e NEXT_PUBLIC_ >> apps/app/.env.production
```

Menerapkan aplikasi Next.js di monorepo

Amplify mendukung aplikasi dalam monorepos generik serta aplikasi di monorepos yang dibuat menggunakan ruang kerja npm, ruang kerja pnpm, ruang kerja Yarn, Nx, dan Turborepo. Saat menerapkan aplikasi, Amplify secara otomatis mendeteksi framework build monorepo yang Anda gunakan. Amplify secara otomatis menerapkan pengaturan build untuk aplikasi di ruang kerja npm, ruang kerja Yarn, atau Nx. Aplikasi Turborepo dan pnpm memerlukan konfigurasi tambahan. Untuk informasi selengkapnya, lihat Mengonfigurasi pengaturan build monorepo.

Untuk contoh Nx mendetail, lihat Bagikan kode antara aplikasi Next.js dengan Nx di postingan blog AWS Amplify Hosting.

Amplify dukungan untuk Nuxt.js

Nuxt adalah kerangka kerja untuk membuat aplikasi web full stack dengan Vue.js.

Adaptor

Anda dapat menerapkan aplikasi Nuxt.js ke Amplify menggunakan adaptor preset dengan konfigurasi nol. Untuk informasi selengkapnya tentang adaptor, lihat dokumentasi Nuxt.

Tutorial

Untuk mempelajari cara menerapkan aplikasi Nuxt.js ke Amplify, lihat. Menerapkan aplikasi Nuxt.js untuk Amplify Hosting

Demo

Untuk demonstrasi video, lihat Hosting Nuxt Dengan Konfigurasi NOL Dalam Menit (Dengan AWS) aktif. YouTube

Amplify dukungan untuk Astro.js

Astro adalah kerangka kerja web untuk membuat aplikasi web berbasis konten.

Adaptor

Anda dapat menerapkan aplikasi Astro.js ke Amplify menggunakan adaptor komunitas. Kami tidak memelihara adaptor milik Amplify untuk kerangka kerja Astro. Namun, adaptor tersedia di github. com/alexnguyennz/astro-aws-amplify di situs web. GitHub Adaptor ini dibuat oleh anggota komunitas dan tidak dikelola oleh AWS.

Tutorial

Untuk mempelajari cara menerapkan aplikasi Astro ke Amplify, lihat. Menerapkan aplikasi Astro.js untuk Amplify Hosting

Demo

Untuk demonstrasi video, lihat Cara menerapkan Situs Web Astro ke AWS saluran Amazon Web Services. YouTube

Amplify dukungan untuk SvelteKit

SvelteKit adalah kerangka kerja untuk membuat aplikasi web full stack dengan Svelte.

Adaptor

Anda dapat menerapkan SvelteKit aplikasi ke Amplify menggunakan adaptor komunitas. Kami tidak memelihara adaptor yang dimiliki Amplify untuk kerangka kerja. SvelteKit Namun, adaptor tersedia di github. com/gzimbron/amplify-adaptor di GitHub situs web. Adaptor ini dibuat oleh anggota komunitas dan tidak dikelola oleh AWS.

Tutorial

Untuk mempelajari cara menerapkan SvelteKit aplikasi ke Amplify, lihat. Menerapkan SvelteKit aplikasi untuk Amplify Hosting

Demo

Untuk demonstrasi video, lihat Cara menerapkan SvelteKit situs web (dengan API) AWS di saluran Amazon Web Services YouTube .

Menerapkan aplikasi SSR untuk Amplify

Anda dapat menggunakan petunjuk ini untuk menerapkan aplikasi yang dibuat dengan kerangka kerja apa pun dengan bundel penerapan yang sesuai dengan keluaran build yang diharapkan Amplify. Jika Anda menerapkan aplikasi Next.js, adaptor tidak diperlukan.

Jika Anda menerapkan aplikasi SSR yang menggunakan adaptor kerangka kerja, Anda harus menginstal dan mengonfigurasi adaptor terlebih dahulu. Untuk petunjuk, lihat Menggunakan adaptor open source untuk kerangka SSR apa pun.

Untuk menerapkan aplikasi SSR ke Amplify Hosting

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.
4. Pada halaman Add repository branch lakukan hal berikut:

   a. Pilih nama repositori untuk terhubung.

   b. Pilih nama cabang repositori untuk terhubung.

   c. Pilih Berikutnya.
5. Pada halaman Pengaturan aplikasi, Amplify secara otomatis mendeteksi aplikasi SSR Next.js.

   Jika Anda menerapkan aplikasi SSR yang menggunakan adaptor untuk kerangka kerja lain, Anda harus secara eksplisit mengaktifkan Amazon Logs. CloudWatch Buka bagian Pengaturan lanjutan, lalu pilih Aktifkan log aplikasi SSR di bagian penyebaran Server-Side Rendering (SSR).
6. Aplikasi ini memerlukan peran layanan IAM yang Amplify asumsikan untuk mengirimkan log ke Anda. Akun AWS

   Prosedur untuk menambahkan peran layanan bervariasi tergantung pada apakah Anda ingin membuat peran baru atau menggunakan peran yang sudah ada.

   - Untuk membuat peran baru:

     - Pilih Buat dan gunakan peran layanan baru.
   - Untuk menggunakan peran yang ada:

     a. Pilih Gunakan peran yang ada.

b. Dalam daftar peran layanan, pilih peran yang akan digunakan.

7. Pilih Berikutnya.
8. Di halaman Tinjauan, pilih Simpan dan deploy.

Fitur yang didukung SSR

Bagian ini memberikan informasi tentang dukungan Amplify untuk fitur SSR.

Amplify menyediakan dukungan versi Node.js agar sesuai dengan versi Node.js yang digunakan untuk membangun aplikasi Anda.

Amplify menyediakan fitur pengoptimalan gambar bawaan yang mendukung semua aplikasi SSR. Jika Anda tidak ingin menggunakan fitur pengoptimalan gambar default, Anda dapat menerapkan pemuat pengoptimalan gambar khusus.

Topik

- Dukungan versi Node.js untuk aplikasi Next.js
- Pengoptimalan gambar untuk aplikasi SSR
- CloudWatch Log Amazon untuk aplikasi SSR
- Amplify dukungan SSR Next.js 11

Dukungan versi Node.js untuk aplikasi Next.js

Saat Amplify membangun dan menerapkan aplikasi komputasi Next.js, Amplify akan menggunakan aplikasi komputasi Node.js versi runtime yang cocok dengan versi utama Node.js yang digunakan untuk membangun aplikasi.

Anda dapat menentukan Node.js versi yang akan digunakan dalam fitur penggantian paket Live di konsol Amplify. Untuk informasi selengkapnya tentang mengonfigurasi pembaruan paket langsung, lihatMenggunakan versi paket dan dependensi tertentu dalam image build. Anda juga dapat menentukan Node.js versi menggunakan mekanisme lain, seperti nvm perintah. Jika Anda tidak menentukan versi, Amplify default untuk menggunakan versi saat ini yang digunakan oleh container build Amplify.

Pengoptimalan gambar untuk aplikasi SSR

Amplify Hosting menyediakan fitur pengoptimalan gambar bawaan yang mendukung semua aplikasi SSR. Dengan optimasi gambar Amplify, Anda dapat memberikan gambar berkualitas tinggi dalam format, dimensi, dan resolusi yang tepat untuk perangkat yang mengaksesnya, sambil mempertahankan ukuran file sekecil mungkin.

Saat ini, Anda dapat menggunakan komponen Gambar Next.js untuk mengoptimalkan gambar sesuai permintaan atau Anda dapat menerapkan pemuat gambar khusus. Jika Anda menggunakan Next.js 13 atau yang lebih baru, Anda tidak perlu mengambil tindakan lebih lanjut untuk menggunakan fitur pengoptimalan gambar Amplify. Jika Anda menerapkan pemuat kustom, lihat berikut ini Menggunakan topik pemuat gambar kustom.

Menggunakan pemuat gambar khusus

Jika Anda menggunakan pemuat gambar khusus, Amplify mendeteksi loader di `next.config.js` file aplikasi Anda dan tidak menggunakan fitur pengoptimalan gambar bawaan. Untuk informasi selengkapnya tentang pemuat kustom yang didukung Next.js, lihat dokumentasi gambar Next.js.

CloudWatch Log Amazon untuk aplikasi SSR

Amplify mengirimkan informasi tentang runtime SSR Anda ke CloudWatch Amazon Logs di Anda. Akun AWS Saat Anda menerapkan aplikasi SSR, aplikasi memerlukan peran layanan IAM yang diasumsikan Amplify saat memanggil layanan lain atas nama Anda. Anda dapat mengizinkan komputasi Amplify Hosting untuk secara otomatis membuat peran layanan untuk Anda atau Anda dapat menentukan peran yang telah Anda buat.

Jika Anda memilih untuk mengizinkan Amplify membuat peran IAM untuk Anda, peran tersebut sudah memiliki izin untuk membuat Log. CloudWatch Jika membuat peran IAM sendiri, Anda perlu menambahkan izin berikut ke kebijakan agar Amplify dapat mengakses Log Amazon. CloudWatch

```
logs:CreateLogStream
logs:CreateLogGroup
logs:DescribeLogGroups
logs:PutLogEvents
```

Untuk informasi selengkapnya tentang peran layanan, lihat Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend.

Amplify dukungan SSR Next.js 11

Jika Anda menerapkan aplikasi Next.js ke Amplify sebelum rilis komputasi Amplify Hosting pada 17 November 2022, aplikasi Anda menggunakan penyedia SSR Amplify sebelumnya, Classic (khusus Next.js 11). Dokumentasi di bagian ini hanya berlaku untuk aplikasi yang digunakan menggunakan penyedia SSR Klasik (hanya Next.js 11).

> **Note**
>
> Kami sangat menyarankan agar Anda memigrasikan aplikasi Next.js 11 Anda ke penyedia SSR terkelola komputasi Amplify Hosting. Untuk informasi selengkapnya, lihat Migrasi aplikasi SSR Next.js 11 ke komputasi Amplify Hosting.

Daftar berikut menjelaskan fitur spesifik yang didukung oleh penyedia SSR Amplify Classic (hanya Next.js 11).

Fitur yang didukung

- Halaman yang dirender sisi server (SSR)
- Halaman statis
- Rute API
- Rute dinamis
- Tangkap semua rute
- SSG (Generasi statis)
- Regenerasi Statis Inkremental (ISR)
- Perutean sub-jalur internasional (i18n)
- Variabel-variabel lingkungan

Fitur yang tidak didukung

- Optimalisasi gambar
- Regenerasi Statis Inkremental Sesuai Permintaan (ISR)
- Perutean domain internasional (i18n)
- Deteksi lokal otomatis yang diinternasionalisasi (i18n)

- Middleware
- Middleware Tepi
- Rute API Edge

Harga untuk aplikasi SSR Next.js 11

Saat menerapkan aplikasi SSR Next.js 11, Amplify membuat resource backend tambahan di akun Anda, termasuk: AWS

- Bucket Amazon Simple Storage Service (Amazon S3) yang menyimpan sumber daya untuk aset statis aplikasi Anda. Untuk informasi seputar harga Amazon S3, lihat Harga Amazon S3.
- CloudFront Distribusi Amazon untuk melayani aplikasi. Untuk informasi tentang CloudFront tagihan, lihat CloudFront Harga Amazon.
- Empat fungsi Lambda @Edge untuk menyesuaikan konten yang CloudFront dikirimkan.

AWS Identity and Access Management izin untuk aplikasi SSR Next.js 11

Amplify memerlukan izin AWS Identity and Access Management (IAM) untuk menerapkan aplikasi SSR. Untuk aplikasi SSR, Amplify menyebarkan sumber daya seperti bucket Amazon S3, distribusi, CloudFront Lambda@Edge fungsi, antrian Amazon SQS (jika menggunakan ISR) dan peran IAM. Tanpa izin minimum yang diperlukan, Anda akan mendapatkan `Access Denied` kesalahan saat mencoba menerapkan aplikasi SSR Anda. Untuk memberikan Amplify dengan izin yang diperlukan, Anda harus menentukan peran layanan.

Untuk membuat peran layanan IAM yang Amplify asumsikan saat memanggil layanan lain atas nama Anda, lihat. Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend Instruksi ini menunjukkan cara membuat peran yang melekat pada kebijakan `AdministratorAccess-Amplify` terkelola.

Kebijakan `AdministratorAccess-Amplify` terkelola menyediakan akses ke beberapa AWS layanan, termasuk tindakan IAM. dan harus dianggap sekuat kebijakan. `AdministratorAccess` Kebijakan ini memberikan lebih banyak izin daripada yang diperlukan untuk menerapkan aplikasi SSR Anda.

Disarankan agar Anda mengikuti praktik terbaik pemberian hak istimewa paling sedikit dan mengurangi izin yang diberikan untuk peran layanan. Alih-alih memberikan izin akses administrator

ke peran layanan Anda, Anda dapat membuat kebijakan IAM terkelola pelanggan Anda sendiri yang hanya memberikan izin yang diperlukan untuk menerapkan aplikasi SSR Anda. Lihat, Membuat kebijakan IAM di Panduan Pengguna IAM untuk petunjuk tentang cara membuat kebijakan terkelola pelanggan.

Jika Anda membuat kebijakan sendiri, lihat daftar izin minimum yang diperlukan untuk menerapkan aplikasi SSR berikut.

```
acm:DescribeCertificate
acm:DescribeCertificate
acm:ListCertificates
acm:RequestCertificate
cloudfront:CreateCloudFrontOriginAccessIdentity
cloudfront:CreateDistribution
cloudfront:CreateInvalidation
cloudfront:GetDistribution
cloudfront:GetDistributionConfig
cloudfront:ListCloudFrontOriginAccessIdentities
cloudfront:ListDistributions
cloudfront:ListDistributionsByLambdaFunction
cloudfront:ListDistributionsByWebACLId
cloudfront:ListFieldLevelEncryptionConfigs
cloudfront:ListFieldLevelEncryptionProfiles
cloudfront:ListInvalidations
cloudfront:ListPublicKeys
cloudfront:ListStreamingDistributions
cloudfront:UpdateDistribution
cloudfront:TagResource
cloudfront:UntagResource
cloudfront:ListTagsForResource
iam:AttachRolePolicy
iam:CreateRole
iam:CreateServiceLinkedRole
iam:GetRole
iam:PutRolePolicy
iam:PassRole
lambda:CreateFunction
lambda:EnableReplication
lambda:DeleteFunction
lambda:GetFunction
lambda:GetFunctionConfiguration
lambda:PublishVersion
lambda:UpdateFunctionCode
```

```
lambda:UpdateFunctionConfiguration
lambda:ListTags
lambda:TagResource
lambda:UntagResource
route53:ChangeResourceRecordSets
route53:ListHostedZonesByName
route53:ListResourceRecordSets
s3:CreateBucket
s3:GetAccelerateConfiguration
s3:GetObject
s3:ListBucket
s3:PutAccelerateConfiguration
s3:PutBucketPolicy
s3:PutObject
s3:PutBucketTagging
s3:GetBucketTagging
lambda:ListEventSourceMappings
lambda:CreateEventSourceMapping
iam:UpdateAssumeRolePolicy
iam:DeleteRolePolicy
sqs:CreateQueue           // SQS only needed if using ISR feature
sqs:DeleteQueue
sqs:GetQueueAttributes
sqs:SetQueueAttributes
amplify:GetApp
amplify:GetBranch
amplify:UpdateApp
amplify:UpdateBranch
```

Pemecahan masalah Next.js 11 penerapan SSR

Jika Anda mengalami masalah tak terduga saat menerapkan aplikasi SSR Klasik (hanya Next.js 11) dengan Amplify, tinjau topik pemecahan masalah berikut.

Topik

- Direktori keluaran aplikasi saya diganti
- Saya mendapatkan kesalahan 404 setelah menerapkan situs SSR saya
- Aplikasi saya tidak memiliki aturan penulisan ulang untuk distribusi CloudFront SSR
- Aplikasi saya terlalu besar untuk diterapkan
- Build saya gagal dengan kesalahan kehabisan memori
- Aplikasi saya memiliki cabang SSR dan SSG

- Aplikasi saya menyimpan file statis dalam folder dengan jalur yang dicadangkan
- Aplikasi saya telah mencapai CloudFront batas
- Fungsi Lambda @Edge dibuat di Wilayah AS Timur (Virginia N.)
- Aplikasi Next.js saya menggunakan fitur yang tidak didukung
- Gambar di aplikasi Next.js saya tidak dimuat
- Wilayah yang Tidak Didukung

Direktori keluaran aplikasi saya diganti

Direktori output untuk aplikasi Next.js yang di-deploy dengan Amplify harus diatur ke `.next`. Jika direktori output aplikasi Anda ditimpa, periksa file `next.config.js`. Untuk mengatur secara default direktori output build ke `.next`, hapus baris berikut dari file:

```
distDir: 'build'
```

Verifikasi bahwa direktori output diatur ke `.next` di pengaturan build Anda. Untuk informasi seputar melihat pengaturan build aplikasi Anda, lihat Mengonfigurasi pengaturan build untuk aplikasi Amplify.

Berikut contoh pengaturan build untuk aplikasi dengan `baseDirectory` diatur ke `.next`.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

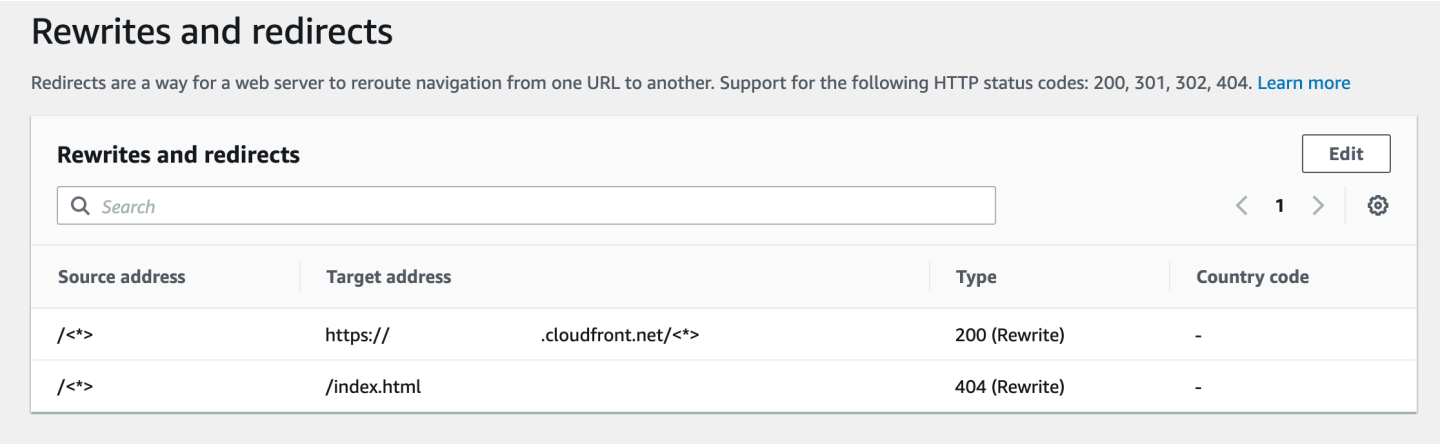
Saya mendapatkan kesalahan 404 setelah menerapkan situs SSR saya

Jika Anda menerima pesan kesalahan 404 setelah men-deploy situs, masalah tersebut dapat terjadi karena direktori output Anda ditimpa. Untuk memeriksa file `next.config.js` dan memverifikasi direktori output build yang benar dalam spesifikasi build aplikasi Anda, ikuti langkah-langkah di topik sebelumnya, Direktori keluaran aplikasi saya diganti.

Aplikasi saya tidak memiliki aturan penulisan ulang untuk distribusi CloudFront SSR

Saat Anda menerapkan aplikasi SSR, Amplify membuat aturan penulisan ulang untuk distribusi SSR Anda. CloudFront Jika Anda tidak dapat mengakses aplikasi di browser web, verifikasi bahwa aturan CloudFront penulisan ulang ada untuk aplikasi Anda di konsol Amplify. Jika aturan tersebut tidak ada, Anda dapat menambahkannya secara manual atau men-deploy ulang aplikasi.

Untuk melihat atau mengedit aturan penulisan ulang dan pengalihan aplikasi di konsol Amplify, pilih Pengaturan aplikasi di panel navigasi, lalu pilih Penulisan ulang dan pengalihan. Tangkapan layar berikut menampilkan contoh aturan penulisan ulang yang dibuat Amplify untuk Anda ketika Anda men-deploy aplikasi SSR. Perhatikan bahwa dalam contoh ini, aturan CloudFront penulisan ulang ada.



Aplikasi saya terlalu besar untuk diterapkan

Amplify membatasi ukuran penyebaran SSR hingga 50 MB. Jika Anda menerima pesan kesalahan `RequestEntityTooLargeException` saat mencoba men-deploy aplikasi Next.js SSR ke Amplify, artinya ukuran aplikasi yang akan di-deploy terlalu besar. Untuk mengatasinya, Anda dapat menambahkan kode pembersihan cache ke file `next.config.js`.

Berikut contoh kode di file `next.config.js` untuk melakukan pembersihan cache.

```
module.exports = {
```

```
    webpack: (config, { buildId, dev, isServer, defaultLoaders, webpack }) => {
        config.optimization.splitChunks.cacheGroups = { }
        config.optimization.minimize = true;
        return config
      },
}
```

Build saya gagal dengan kesalahan kehabisan memori

Next.js memungkinkan Anda untuk menyimpan artefak build cache untuk meningkatkan kinerja pada build berikutnya. Selain itu, AWS CodeBuild container Amplify mengompres dan mengunggah cache ini ke Amazon S3, atas nama Anda, untuk meningkatkan performa build berikutnya. Ini dapat menyebabkan build Anda gagal dengan kesalahan kehabisan memori.

Lakukan tindakan berikut untuk mencegah aplikasi Anda melebihi batas memori selama fase build. Pertama, hapus `.next/cache/**/*` dari bagian cache.paths dari pengaturan build Anda. Selanjutnya, hapus variabel `NODE_OPTIONS` lingkungan dari file setelan build Anda. Sebagai gantinya, atur variabel `NODE_OPTIONS` lingkungan di konsol Amplify untuk menentukan batas memori maksimum Node. Untuk informasi selengkapnya tentang menyetel variabel lingkungan menggunakan konsol Amplify, lihat. Mengatur variabel lingkungan

Setelah melakukan perubahan ini, coba build Anda lagi. Jika berhasil, tambahkan `.next/cache/**/*` kembali ke bagian cache.paths dari file pengaturan build Anda.

Untuk informasi selengkapnya tentang konfigurasi cache Next.js guna meningkatkan performa build, lihat AWS CodeBuild di situs web Next.js.

Aplikasi saya memiliki cabang SSR dan SSG

Anda tidak dapat men-deploy aplikasi yang memiliki cabang SSR dan SSG. Untuk men-deploy cabang SSR dan SSG, Anda harus men-deploy aplikasi yang menggunakan cabang SSR saja dan aplikasi lain yang menggunakan cabang SSG saja.

Aplikasi saya menyimpan file statis dalam folder dengan jalur yang dicadangkan

Next.js dapat melayani file statis dari folder bernama `public` yang disimpan di direktori root proyek. Ketika Anda men-deploy dan meng-host aplikasi Next.js dengan Amplify, proyek Anda tidak dapat menyertakan folder dengan path `public/static`. Amplify mengatur agar path `public/static` digunakan saat mendistribusikan aplikasi. Jika aplikasi Anda mencakup path ini, Anda harus mengubah nama folder `static` sebelum men-deploy dengan Amplify.

Aplikasi saya telah mencapai CloudFront batas

CloudFront kuota layanan membatasi AWS akun Anda hingga 25 distribusi dengan fungsi Lambda @Edge terlampir. Jika melebihi kuota ini, Anda dapat menghapus CloudFront distribusi yang tidak terpakai dari akun Anda atau meminta peningkatan kuota. Untuk informasi selengkapnya, lihat Meminta peningkatan kuota di Panduan Pengguna Service Quotas.

Fungsi Lambda @Edge dibuat di Wilayah AS Timur (Virginia N.)

Saat Anda menerapkan aplikasi Next.js, Amplify membuat fungsi Lambda @Edge untuk menyesuaikan konten yang dikirimkan. CloudFront Fungsi Lambda @Edge dibuat di Wilayah AS Timur (Virginia N.), bukan Wilayah tempat aplikasi Anda digunakan. Ini adalah pembatasan Lambda @Edge. Untuk informasi selengkapnya tentang fungsi Lambda @Edge, lihat Pembatasan fungsi edge di Panduan CloudFront Pengembang Amazon.

Aplikasi Next.js saya menggunakan fitur yang tidak didukung

Aplikasi yang digunakan dengan Amplify mendukung versi utama Next.js hingga versi 11. Untuk daftar detail fitur Next.js yang didukung dan tidak didukung oleh Amplify, lihat. supported features

Ketika Anda men-deploy aplikasi Next.js baru, Amplify menggunakan versi Next.js terbaru yang didukung secara default. Jika Anda memiliki aplikasi Next.js yang sudah digunakan untuk Amplify dengan versi Next.js yang lebih lama, Anda dapat memigrasikan aplikasi ke penyedia SSR komputasi Amplify Hosting. Untuk petunjuk, lihat Migrasi aplikasi SSR Next.js 11 ke komputasi Amplify Hosting.

Gambar di aplikasi Next.js saya tidak dimuat

Jika Anda menambahkan gambar ke aplikasi Next.js menggunakan `next/image` komponen, ukuran gambar tidak boleh melebihi 1 MB. Saat Anda menerapkan aplikasi ke Amplify, gambar yang lebih besar dari 1 MB akan menampilkan kesalahan 503. Ini disebabkan oleh batas Lambda @Edge yang membatasi ukuran respons yang dihasilkan oleh fungsi Lambda, termasuk header dan badan, hingga 1 MB.

Batas 1 MB berlaku untuk artefak lain di aplikasi Anda, seperti file PDF dan dokumen.

Wilayah yang Tidak Didukung

Amplify tidak mendukung penerapan aplikasi SSR Klasik (khusus Next.js 11) di setiap wilayah AWS tempat Amplify tersedia. SSR klasik (hanya Next.js 11) tidak didukung di Wilayah berikut: Eropa (Milan) eu-south-1, Timur Tengah (Bahrain) me-south-1, dan Asia Pasifik (Hong Kong) ap-east-1.

Harga untuk aplikasi SSR

Saat Anda menerapkan aplikasi SSR, Amplify Hosting compute mengelola sumber daya yang diperlukan untuk menerapkan aplikasi SSR untuk Anda. Untuk informasi tentang biaya komputasi Amplify Hosting, lihat Harga.AWS Amplify

Memecahkan masalah penerapan SSR

Jika Anda mengalami masalah tak terduga saat menerapkan aplikasi SSR dengan komputasi Amplify Hosting, lihat di Memecahkan masalah aplikasi sisi server bagian pemecahan masalah Amplify.

Lanjutan: Adaptor sumber terbuka

Penulis kerangka kerja dapat menggunakan spesifikasi penerapan berbasis sistem file untuk mengembangkan adaptor build open source yang disesuaikan untuk kerangka kerja spesifik mereka. Adaptor ini akan mengubah output build aplikasi menjadi bundel penerapan yang sesuai dengan struktur direktori yang diharapkan Amplify Hosting. Bundel penerapan ini akan mencakup semua file dan aset yang diperlukan untuk meng-host aplikasi, termasuk konfigurasi runtime, seperti aturan perutean.

Jika Anda tidak menggunakan kerangka kerja, Anda dapat mengembangkan solusi Anda sendiri untuk menghasilkan output build yang diharapkan Amplify.

Topik

- Menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build
- Menyebarkan server Express menggunakan manifes penerapan
- Integrasi optimasi gambar untuk penulis kerangka kerja
- Menggunakan adaptor open source untuk kerangka SSR apa pun

Menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build

Spesifikasi penerapan Amplify Hosting adalah spesifikasi berbasis sistem file yang mendefinisikan struktur direktori yang memfasilitasi penerapan ke Amplify Hosting. Kerangka kerja dapat menghasilkan struktur direktori yang diharapkan ini sebagai output dari perintah build-nya,

memungkinkan kerangka kerja untuk memanfaatkan primitif layanan Amplify Hosting. Amplify Hosting memahami struktur bundel penerapan dan menerapkannya sesuai dengan itu.

Untuk demonstrasi video yang menjelaskan cara menggunakan spesifikasi penerapan, lihat Cara meng-host situs web apa pun yang menggunakan saluran AWS Amplify Amazon Web Services YouTube .

Berikut ini adalah contoh struktur folder yang Amplify harapkan untuk bundel penerapan. Pada tingkat tinggi, ia memiliki folder bernama`static`, folder bernama `compute` dan file manifes penyebaran bernama`deploy-manifest.json`.

```
.amplify-hosting/
### compute/
#   ### default/
#       ### chunks/
#       #   ### app/
#       #       ### _nuxt/
#       #       #   ### index-xxx.mjs
#       #       #   ### index-styles.xxx.js
#       #       ### server.mjs
#       ### node_modules/
#       ### server.js
### static/
#   ### css/
#   #   ### nuxt-google-fonts.css
#   ### fonts/
#   #   ### font.woff2
#   ### _nuxt/
#   #   ### builds/
#   #   #   ### latest.json
#   #   ### entry.xxx.js
#   ### favicon.ico
#   ### robots.txt
### deploy-manifest.json
```

Amplify dukungan primitif SSR

Spesifikasi penerapan Amplify Hosting mendefinisikan kontrak yang memetakan secara dekat ke primitif berikut.

Aset statis

Menyediakan kerangka kerja dengan kemampuan untuk meng-host file statis.

Hitung

Menyediakan kerangka kerja dengan kemampuan untuk menjalankan server HTTP Node.js pada port 3000.

Optimalisasi gambar

Menyediakan kerangka kerja dengan layanan untuk mengoptimalkan gambar saat runtime.

Aturan perutean

Menyediakan kerangka kerja dengan mekanisme untuk memetakan jalur permintaan masuk ke target tertentu.

Bagian .amplify-hosting/static direktori

Anda harus menempatkan semua file statis yang dapat diakses publik yang dimaksudkan untuk disajikan dari URL aplikasi di `.amplify-hosting/static` direktori. File di dalam direktori ini disajikan melalui aset statis primitif.

File statis dapat diakses di root (/) URL aplikasi tanpa perubahan apa pun pada konten, nama file, atau ekstensinya. Selain itu, subdirektori dipertahankan dalam struktur URL dan muncul sebelum nama file. Sebagai contoh, `.amplify-hosting/static/favicon.ico` akan dilayani dari `https://myAppId.amplify-hostingapp.com/favicon.ico` dan `.amplify-hosting/static/_nuxt/main.js` akan dilayani dari `https://myAppId.amplify-hostingapp.com/_nuxt/main.js`

Jika kerangka kerja mendukung kemampuan untuk memodifikasi jalur dasar aplikasi, itu harus menambahkan jalur dasar ke aset statis di dalam direktori. `.amplify-hosting/static` Misalnya, jika jalur dasarnya`/folder1/folder2`, maka output build untuk aset statis yang dipanggil `main.css` akan menjadi`.amplify-hosting/static/folder1/folder2/main.css`.

Bagian .amplify-hosting/compute direktori

Sumber daya komputasi tunggal diwakili oleh subdirektori tunggal bernama `default` terkandung dalam direktori. `.amplify-hosting/compute` Jalannya adalah`.amplify-hosting/compute/default`. Sumber daya komputasi ini memetakan ke primitif komputasi Amplify Hosting.

Isi `default` subdirektori harus sesuai dengan aturan berikut.

- File harus ada di root `default` subdirektori, untuk berfungsi sebagai titik masuk ke sumber daya komputasi.
- File titik masuk harus berupa modul Node.js dan harus memulai server HTTP yang mendengarkan pada port 3000.
- Anda dapat menempatkan file lain di `default` subdirektori dan mereferensikannya dari kode di file titik masuk.
- Isi subdirektori harus mandiri. Kode dalam modul titik masuk tidak dapat mereferensikan modul apa pun di luar subdirektori. Perhatikan bahwa kerangka kerja dapat menggabungkan server HTTP mereka dengan cara apa pun yang mereka inginkan. Jika proses komputasi dapat dimulai dengan `node server.js` perintah, di mana `server.js is` adalah nama file entri, dari dalam subdirektori, Amplify mempertimbangkan struktur direktori agar sesuai dengan spesifikasi penyebaran.

Amplify Hosting bundel dan gunakan semua file di dalam `default` subdirektori ke sumber daya komputasi yang disediakan. Setiap sumber daya komputasi dialokasikan 512 MB penyimpanan sementara. Penyimpanan ini tidak dibagi antara instance eksekusi, tetapi dibagi di antara pemanggilan berikutnya dalam instance eksekusi yang sama. Contoh eksekusi dibatasi hingga waktu eksekusi maksimum 15 menit, dan satu-satunya jalur yang dapat ditulis dalam instance eksekusi adalah direktori. `/tmp` Ukuran terkompresi dari setiap bundel sumber daya komputasi tidak dapat melebihi 220 MB. Misalnya, `.amplify/compute/default` subdirektori tidak dapat melebihi 220 MB saat dikompresi.

Bagian .amplify-hosting/deploy-manifest.json file

Gunakan `deploy-manifest.json` file untuk menyimpan detail konfigurasi dan metadata untuk penerapan. Minimal, `deploy-manifest.json` file harus menyertakan `version` atribut, `routes` atribut dengan rute catch-all yang ditentukan, dan `framework` atribut dengan metadata kerangka ditentukan.

Definisi objek berikut menunjukkan konfigurasi untuk manifes penerapan.

```
type DeployManifest = {
  version: 1;
  routes: Route[];
  computeResources?: ComputeResource[];
  imageSettings?: ImageSettings;
  framework: FrameworkMetadata;
};
```

Topik berikut menjelaskan detail dan penggunaan untuk setiap atribut dalam manifes penerapan.

Menggunakan atribut versi

`version`Atribut mendefinisikan versi spesifikasi penerapan yang Anda terapkan. Saat ini, satu-satunya versi untuk spesifikasi penerapan Amplify Hosting adalah versi 1. Contoh JSON berikut menunjukkan penggunaan untuk atribut. `version`

```
"version": 1
```

Menggunakan atribut routes

`routes`Atribut ini memungkinkan kerangka kerja untuk memanfaatkan aturan perutean Amplify Hosting primitif. Aturan perutean menyediakan mekanisme untuk merutekan jalur permintaan masuk ke target tertentu dalam bundel penerapan. Aturan perutean hanya menentukan tujuan permintaan yang masuk dan diterapkan setelah permintaan diubah oleh aturan penulisan ulang dan pengalihan. Untuk informasi selengkapnya tentang cara Amplify Hosting menangani penulisan ulang dan pengalihan, lihat. Menyiapkan pengalihan dan penulisan ulang untuk aplikasi Amplify

Aturan perutean tidak menulis ulang atau mengubah permintaan. Jika permintaan masuk cocok dengan pola jalur untuk rute, permintaan akan dirutekan apa adanya ke target rute.

Aturan routing yang ditentukan dalam `routes` array harus sesuai dengan aturan berikut.

- Rute catch-all harus ditentukan. Rute catch-all memiliki `/*` pola yang cocok dengan semua permintaan yang masuk.
- `routes`Array dapat berisi maksimal 25 item.
- Anda harus menentukan `Static` rute atau `Compute` rute.
- Jika Anda menentukan `Static` rute, `.amplify-hosting/static` direktori harus ada.
- Jika Anda menentukan `Compute` rute, `.amplify-hosting/compute` direktori harus ada.
- Jika Anda menentukan `ImageOptimization` rute, Anda juga harus menentukan `Compute` rute. Ini diperlukan karena optimasi gambar belum didukung untuk aplikasi statis murni.

Definisi objek berikut menunjukkan konfigurasi untuk `Route` objek.

```
type Route = {
  path: string;
  target: Target;
  fallback?: Target;
```

```
}
```

Tabel berikut menjelaskan properti `Route` objek.

Kunci	Tipe	Diperlukan	Deskripsi
path	Tali	Ya	Mendefinisikan pola yang cocok dengan jalur permintaan masuk (tidak termasuk querystring). Panjang jalur maksimum adalah 255 karakter. Jalur harus dimulai dengan garis miring `/` ke depan. Sebuah jalur dapat berisi salah satu karakter berikut: [A-Z], [a-z], [0-9], [_-.*$/~'"@: +]. Untuk pencocokan pola, hanya karakter wildcard berikut yang didukung: • `*`(cocok dengan 0 karakter atau lebih) • `/*`Pola ini disebut pola catch-all dan akan cocok dengan semua permintaan yang masuk.

Kunci	Tipe	Diperlukan	Deskripsi
target	Target	Ya	Objek yang mendefinisikan target untuk merutekan permintaan yang cocok. Jika `Compute` rute ditentukan, yang sesuai `ComputeResource` harus ada. Jika `ImageOptimization` rute ditentukan, juga `imageSettings` harus ditentukan.

Kunci	Tipe	Diperlukan	Deskripsi
mundur	Target	Tidak	Objek yang mendefinisikan target untuk mundur jika target asli mengembalikan kesalahan 404. `target`Jenis dan `fallback` jenisnya tidak bisa sama untuk rute yang ditentukan. Misalnya, fallback from `Static` to tidak `Static` diperbolehkan. Fallback hanya didukung untuk permintaan GET yang tidak memiliki badan. Jika ada badan dalam permintaan, itu akan dijatuhkan selama fallback.

Definisi objek berikut menunjukkan konfigurasi untuk `Target` objek.

```
type Target = {
  kind: TargetKind;
  src?: string;
  cacheControl?: string;
}
```

Tabel berikut menjelaskan properti `Target` objek.

Kunci	Tipe	Diperlukan	Deskripsi
jenis	Targetkind	Ya	An `enum` yang mendefinisikan tipe target. Nilai yang valid adalah `Static`, `Compute`, dan `ImageOptimization` .
src	String	Ya untuk `Compute` Tidak untuk primitif lainnya	String yang menentukan nama subdirektori dalam bundel penyebaran yang berisi kode executable primitif. Valid dan diperlukan hanya untuk primitif Compute. Nilai harus menunjuk ke salah satu sumber daya komputasi yang ada dalam bundel penerapan. Saat ini, satu-satunya nilai yang didukung untuk bidang ini adalah`default`.
CacheControl	String	Tidak	String yang menentukan nilai header Cache-Control untuk diterapkan pada respons. Hanya berlaku untuk

Kunci	Tipe	Diperlukan	Deskripsi
			Statis dan ImageOptimization primitif. Nilai yang ditentukan diganti oleh header khusus. Untuk informasi selengkapnya tentang header pelanggan Amplify Hosting, lihat. Menyetel header khusus untuk aplikasi Amplify **Note** Header Cache-Control ini hanya diterapkan pada respons yang berhasil dengan kode status disetel ke 200 (OK).

Definisi objek berikut menunjukkan penggunaan untuk `TargetKind` enumerasi.

```
enum TargetKind {
  Static = "Static",
  Compute = "Compute",
  ImageOptimization = "ImageOptimization"
}
```

Daftar berikut menentukan nilai yang valid untuk `TargetKind` enum.

Statis

Rute permintaan ke aset statis primitif.

Hitung

Permintaan rute ke primitif komputasi.

ImageOptimization

Permintaan rute ke primitif optimasi gambar.

Contoh JSON berikut menunjukkan penggunaan untuk `routes` atribut dengan beberapa aturan routing ditentukan.

```
"routes": [
    {
      "path": "/_nuxt/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/_nuxt/builds/meta/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
```

```
    {
      "path": "/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ]
```

Untuk informasi selengkapnya tentang menentukan aturan perutean dalam manifes penerapan, lihat Praktik terbaik untuk mengonfigurasi aturan perutean

Menggunakan atribut ComputereSources

`computeResources`Atribut memungkinkan kerangka kerja untuk menyediakan metadata tentang sumber daya komputasi yang disediakan. Setiap sumber daya komputasi harus memiliki rute yang sesuai yang terkait dengannya.

Definisi objek berikut menunjukkan penggunaan untuk `ComputeResource` objek.

```
type ComputeResource = {
  name: string;
  runtime: ComputeRuntime;
  entrypoint: string;
};

type ComputeRuntime = 'nodejs16.x' | 'nodejs18.x' | 'nodejs20.x';
```

Tabel berikut menjelaskan properti `ComputeResource` objek.

Kunci	Tipe	Diperlukan	Deskripsi
nama	Tali	Ya	Menentukan nama sumber daya komputasi. Nama harus cocok dengan nama subdirektori di dalam. `.amplify-hosting/compute directory` Untuk versi 1 dari spesifikasi penerapan , satu-satunya nilai yang valid adalah`default`.
runtime	ComputeRuntime	Ya	Mendefinisikan runtime untuk sumber daya komputasi yang disediakan. Nilai yang valid adalah `nodejs16.x` , `nodejs18.x` , dan `nodejs20.x` .
titik masuk	Tali	Ya	Menentukan nama file awal yang kode akan berjalan dari sumber daya komputasi tertentu. File harus ada di dalam subdirektori yang mewakili sumber daya komputasi.

Jika Anda memiliki struktur direktori yang terlihat seperti berikut ini.

```
.amplify-hosting
|---compute
|   |---default
|       |---index.js
```

JSON untuk `computeResource` atribut akan terlihat seperti berikut ini.

```
"computeResources": [
    {
      "name": "default",
      "runtime": "nodejs16.x",
      "entrypoint": "index.js",
    }
  ]
```

Menggunakan atribut ImageSettings

`imageSettings`Atribut ini memungkinkan kerangka kerja untuk menyesuaikan perilaku primitif pengoptimalan gambar, yang menyediakan optimasi gambar sesuai permintaan saat runtime.

Definisi objek berikut menunjukkan penggunaan untuk `ImageSettings` objek.

```
type ImageSettings = {
  sizes: number[];
  domains: string[];
  remotePatterns: RemotePattern[];
  formats: ImageFormat[];
  minumumCacheTTL: number;
  dangerouslyAllowSVG: boolean;
};

type ImageFormat = 'image/avif' | 'image/webp' | 'image/png' | 'image/jpeg';
```

Tabel berikut menjelaskan properti `ImageSettings` objek.

Kunci	Tipe	Diperlukan	Deskripsi
ukuran	Nomor []	Ya	Array lebar gambar yang didukung.

Kunci	Tipe	Diperlukan	Deskripsi
wilayah	Tali []	Ya	Array domain eksternal yang diizinkan yang dapat menggunak an optimasi gambar. Biarkan array kosong untuk mengizink an hanya domain penerapan untuk menggunakan optimasi gambar.
RemotePatterns	RemotePattern[]	Ya	Array pola eksternal yang diizinkan yang dapat menggunak an optimasi gambar. Mirip dengan domain, tetapi memberikan kontrol lebih dengan ekspresi reguler (regex).
format	ImageFormat[]	Ya	Array format gambar keluaran yang diizinkan.
MinimumCachettl	Bilangan	Ya	Durasi cache dalam hitungan detik untuk gambar yang dioptimalkan.

Kunci	Tipe	Diperlukan	Deskripsi
BerbahayaAllowSVG	Boolean	Ya	Memungkinkan gambar URLs masukan SVG. Ini dinonaktifkan secara default untuk tujuan keamanan.

Definisi objek berikut menunjukkan penggunaan untuk `RemotePattern` objek.

```
type RemotePattern = {
  protocol?: 'https';
  hostname: string;
  port?: string;
  pathname?: string;
}
```

Tabel berikut menjelaskan properti `RemotePattern` objek.

Kunci	Tipe	Diperlukan	Deskripsi
protokol	String	Tidak	Protokol pola jarak jauh yang diizinkan . Satu-satunya nilai yang valid adalah `https`.
hostname	Tali	Ya	Nama host dari pola jarak jauh yang diizinkan. Anda dapat menentukan literal atau wildcard. Satu `*` cocok dengan satu subdomain. Sebuah `**` ganda cocok

Kunci	Tipe	Diperlukan	Deskripsi
			dengan sejumlah subdomain. Amplify tidak mengizinkan wildcard selimut di mana hanya `**` yang ditentukan.
port	String	Tidak	Port dari pola jarak jauh yang diizinkan.
nama jalur	String	Tidak	Nama jalur dari pola jarak jauh yang diizinkan.

Contoh berikut menunjukkan `imageSettings` atribut.

```
"imageSettings": {
    "sizes": [
      100,
      200
    ],
    "domains": [
      "example.com"
    ],
    "remotePatterns": [
      {
        "protocol": "https",
        "hostname": "example.com",
        "port": "",
        "pathname": "/**",
      }
    ],
    "formats": [
      "image/webp"
    ],
    "minumumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  }
```

Menggunakan atribut framework

Gunakan `framework` atribut untuk menentukan kerangka metadata.

Definisi objek berikut menunjukkan konfigurasi untuk `FrameworkMetadata` objek.

```
type FrameworkMetadata = {
  name: string;
  version: string;
}
```

Tabel berikut menjelaskan properti `FrameworkMetadata` objek.

Kunci	Tipe	Diperlukan	Deskripsi
nama	Tali	Ya	Nama kerangka kerja.
versi	Tali	Ya	Versi kerangka kerja. Itu harus berupa string versi semantik (semver) yang valid.

Praktik terbaik untuk mengonfigurasi aturan perutean

Aturan perutean menyediakan mekanisme untuk merutekan jalur permintaan masuk ke target tertentu dalam bundel penerapan. Dalam bundel penerapan, pembuat kerangka kerja dapat memancarkan file ke output build yang diterapkan ke salah satu target berikut:

- Aset statis primitif — File terkandung dalam `.amplify-hosting/static` direktori.
- Compute primitive — File yang terkandung dalam direktori. `.amplify-hosting/compute/default`

Penulis kerangka kerja juga menyediakan serangkaian aturan perutean dalam file manifes penyebaran. Setiap aturan dalam array dicocokkan dengan permintaan yang masuk dalam urutan traversal berurutan, hingga ada kecocokan. Ketika ada aturan yang cocok, permintaan dirutekan ke target yang ditentukan dalam aturan pencocokan. Secara opsional, target fallback dapat ditentukan

untuk setiap aturan. Jika target asli mengembalikan kesalahan 404, permintaan dirutekan ke target fallback.

Spesifikasi penerapan mensyaratkan aturan terakhir dalam urutan traversal menjadi aturan catch-all. Aturan catch-all ditentukan dengan jalur. `/*` Jika permintaan masuk tidak cocok dengan rute sebelumnya dalam larik aturan perutean, permintaan akan dirutekan ke target aturan catch-all.

Untuk kerangka kerja SSR seperti Nuxt.js, target aturan catch-all harus primitif komputasi. Ini karena aplikasi SSR memiliki halaman yang dirender sisi server dengan rute yang tidak dapat diprediksi pada waktu pembuatan. Misalnya, jika Nuxt.js aplikasi memiliki halaman di `/blog/[slug]` mana `[slug]` adalah parameter rute dinamis. Target aturan catch-all adalah satu-satunya cara untuk merutekan permintaan ke halaman ini.

Sebaliknya, pola jalur tertentu dapat digunakan untuk menargetkan rute yang diketahui pada waktu pembuatan. Misalnya, Nuxt.js melayani aset statis dari `/_nuxt` jalur. Ini berarti bahwa `/_nuxt/*` jalur dapat ditargetkan oleh aturan perutean tertentu yang merutekan permintaan ke aset statis primitif.

Perutean folder publik

Sebagian besar kerangka kerja SSR menyediakan kemampuan untuk melayani aset statis yang dapat berubah dari folder. `public` File seperti `favicon.ico` dan `robots.txt` biasanya disimpan di dalam `public` folder dan disajikan dari URL root aplikasi. Misalnya, `favicon.ico` file dilayani dari`https://example.com/favicon.ico`. Perhatikan bahwa tidak ada pola jalur yang dapat diprediksi untuk file-file ini. Mereka hampir seluruhnya ditentukan oleh nama file. Satu-satunya cara untuk menargetkan file di dalam `public` folder adalah dengan menggunakan rute catch-all. Namun, target rute catch-all harus primitif komputasi.

Kami merekomendasikan salah satu pendekatan berikut untuk mengelola `public` folder Anda.

1. Gunakan pola jalur untuk menargetkan jalur permintaan yang berisi ekstensi file. Misalnya, Anda dapat menggunakan `/*.*` untuk menargetkan semua jalur permintaan yang berisi ekstensi file.

   Perhatikan bahwa pendekatan ini bisa tidak dapat diandalkan. Misalnya, jika ada file tanpa ekstensi file di dalam `public` folder, mereka tidak ditargetkan oleh aturan ini. Masalah lain yang harus diperhatikan dengan pendekatan ini adalah bahwa aplikasi dapat memiliki halaman dengan periode dalam nama mereka. Misalnya, halaman di `/blog/2021/01/01/hello.world` akan ditargetkan oleh `/*.*` aturan. Ini tidak ideal karena halaman tersebut bukan aset statis. Namun, Anda dapat menambahkan target fallback ke aturan ini untuk memastikan bahwa ketika ada kesalahan 404 dari primitif statis, permintaan akan kembali ke primitif komputasi.

```
{
    "path": "/*.*",
    "target": {
        "kind": "Static"
    },
    "fallback": {
        "kind": "Compute",
        "src": "default"
    }
}
```

2. Identifikasi file di `public` folder pada waktu pembuatan dan keluarkan aturan perutean untuk setiap file. Pendekatan ini tidak dapat diskalakan karena ada batas 25 aturan yang diberlakukan oleh spesifikasi penerapan.

```
{
    "path": "/favicon.ico",
    "target": {
        "kind": "Static"
    }
},
{
    "path": "/robots.txt",
    "target": {
        "kind": "Static"
    }
}
```

3. Sarankan agar pengguna kerangka kerja Anda menyimpan semua aset statis yang dapat berubah di dalam sub-folder di dalam folder. `public`

   Dalam contoh berikut, pengguna dapat menyimpan semua aset statis yang bisa berubah di dalam `public/assets` folder. Kemudian, aturan perutean dengan pola jalur `/assets/*` dapat digunakan untuk menargetkan semua aset statis yang dapat berubah di dalam folder`public/assets`.

```
{
    "path": "/assets/*",
    "target": {
        "kind": "Static"
    }
```

```
}
```

4. Tentukan fallback statis untuk rute catch-all. Pendekatan ini memiliki kelemahan yang dijelaskan secara lebih rinci di Tangkap semua perutean fallback bagian selanjutnya.

Tangkap semua perutean fallback

Untuk kerangka kerja SSR seperti Nuxt.js, di mana rute catch-all ditentukan untuk target primitif komputasi, penulis kerangka kerja mungkin mempertimbangkan untuk menentukan fallback statis untuk rute catch-all untuk menyelesaikan masalah perutean folder. `public` Namun, jenis aturan perutean ini merusak 404 halaman yang dirender sisi server. Misalnya, jika pengguna akhir mengunjungi halaman yang tidak ada, aplikasi akan merender halaman 404 dengan kode status 404. Namun, jika rute catch-all memiliki fallback statis, halaman 404 tidak akan dirender. Sebaliknya, permintaan kembali ke primitif statis dan masih berakhir dengan kode status 404, tetapi halaman 404 tidak dirender.

```
{
    "path": "/*",
    "target": {
        "kind": "Compute",
        "src": "default"
    },
    "fallback": {
        "kind": "Static"
    }
}
```

Perutean jalur dasar

Kerangka kerja yang menawarkan kemampuan untuk memodifikasi jalur dasar aplikasi diharapkan untuk menambahkan jalur dasar ke aset statis di dalam direktori. `.amplify-hosting/static` Misalnya, jika jalur dasarnya`/folder1/folder2`, maka output build untuk aset statis yang disebut main.css akan menjadi`.amplify-hosting/static/folder1/folder2/main.css`.

Ini berarti bahwa aturan routing juga perlu diperbarui untuk mencerminkan jalur dasar. Misalnya, jika jalur dasarnya`/folder1/folder2`, maka aturan perutean untuk aset statis di `public` folder akan terlihat seperti berikut ini.

```
{
    "path": "/folder1/folder2/*.*",
```

```
    "target": {
        "kind": "Static"
    }
}
```

Demikian pula, rute sisi server juga perlu memiliki jalur dasar yang ditambahkan ke mereka. Misalnya, jika jalur dasarnya`/folder1/folder2`, maka aturan perutean untuk `/api` rute akan terlihat seperti berikut.

```
{
    "path": "/folder1/folder2/api/*",
    "target": {
        "kind": "Compute",
        "src": "default"
    }
}
```

Namun, jalur dasar tidak boleh dilanjutkan ke rute catch-all. Misalnya, jika jalur dasarnya`/folder1/folder2`, maka rute catch-all akan tetap seperti berikut ini.

```
{
    "path": "/*",
    "target": {
        "kind": "Compute",
        "src": "default"
    }
}
```

Contoh rute Nuxt.js

Berikut ini adalah `deploy-manifest.json` file contoh untuk aplikasi Nuxt yang menunjukkan cara menentukan aturan routing.

```
{
  "version": 1,
  "routes": [
    {
      "path": "/_nuxt/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
```

```
    },
    {
      "path": "/_nuxt/builds/meta/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
```

```
      "runtime": "nodejs18.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

Berikut ini adalah `deploy-manifest.json` file contoh untuk Nuxt yang menunjukkan cara menentukan aturan routing termasuk jalur dasar.

```
{
  "version": 1,
  "routes": [
    {
      "path": "/base-path/_nuxt/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/base-path/_nuxt/builds/meta/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
```

```
    {
      "path": "/base-path/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
      "runtime": "nodejs18.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

Untuk informasi selengkapnya tentang penggunaan `routes` atribut, lihatMenggunakan atribut routes.

Menyebarkan server Express menggunakan manifes penerapan

Contoh ini menjelaskan cara menerapkan server Express dasar menggunakan spesifikasi penerapan Amplify Hosting. Anda dapat memanfaatkan manifes penerapan yang disediakan untuk menentukan perutean, sumber daya komputasi, dan konfigurasi lainnya.

Siapkan server Express secara lokal sebelum menerapkan ke Amplify Hosting

1. Buat direktori baru untuk proyek Anda dan instal Express dan TypeScript.

```
mkdir express-app
cd express-app

# The following command will prompt you for information about your project
npm init

# Install express, typescript and types
npm install express --save
npm install typescript ts-node @types/node @types/express --save-dev
```

2. Tambahkan `tsconfig.json` file ke root proyek Anda dengan konten berikut.

```
{
  "compilerOptions": {
    "target": "es6",
    "module": "commonjs",
    "outDir": "./dist",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true
  },
  "include": ["src/**/*.ts"],
  "exclude": ["node_modules"]
}
```

3. Buat direktori bernama `src` di root proyek Anda.
4. Buat `index.ts` file di `src` direktori. Ini akan menjadi titik masuk ke aplikasi yang memulai server Express. Server harus dikonfigurasi untuk mendengarkan pada port 3000.

```
// src/index.ts
import express from 'express';

const app: express.Application = express();
const port = 3000;

app.use(express.text());

app.listen(port, () => {
  console.log(`server is listening on ${port}`);
});
```

```
// Homepage
app.get('/', (req: express.Request, res: express.Response) => {
  res.status(200).send("Hello World!");
});

// GET
app.get('/get', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-get-header", "get-header-value").send("get-response-
from-compute");
});

//POST
app.post('/post', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-post-header", "post-header-
value").send(req.body.toString());
});

//PUT
app.put('/put', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-put-header", "put-header-
value").send(req.body.toString());
});

//PATCH
app.patch('/patch', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-patch-header", "patch-header-
value").send(req.body.toString());
});

// Delete
app.delete('/delete', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-delete-header", "delete-header-value").send();
});
```

5. Tambahkan skrip berikut ke `package.json` file Anda.

```
"scripts": {
  "start": "ts-node src/index.ts",
  "build": "tsc",
  "serve": "node dist/index.js"
}
```

6. Buat direktori bernama `public` di root proyek Anda. Kemudian buat file bernama `hello-world.txt` dengan konten berikut.

```
Hello world!
```

7. Tambahkan `.gitignore` file ke root proyek Anda dengan konten berikut.

```
.amplify-hosting
dist
node_modules
```

Siapkan manifes penerapan Amplify

1. Buat file bernama `deploy-manifest.json` di direktori root proyek Anda.
2. Salin dan tempel manifes berikut ke dalam `deploy-manifest.json` file Anda.

```
{
  "version": 1,
  "framework": { "name": "express", "version": "4.18.2" },
  "imageSettings": {
    "sizes": [
      100,
      200,
      1920
    ],
    "domains": [],
    "remotePatterns": [],
    "formats": [],
    "minimumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  },
  "routes": [
    {
      "path": "/_amplify/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/*.*",
```

```
      "target": {
        "kind": "Static",
        "cacheControl": "public, max-age=2"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "runtime": "nodejs18.x",
      "entrypoint": "index.js"
    }
  ]
}
```

Manifes menjelaskan bagaimana Amplify Hosting harus menangani penerapan aplikasi Anda. Pengaturan utama adalah sebagai berikut.

- version — Menunjukkan versi spesifikasi penerapan yang Anda gunakan.
- framework — Sesuaikan ini untuk menentukan Express pengaturan server.
- ImageSettings - Bagian ini opsional untuk Express server kecuali Anda menangani optimasi gambar.
- rute — Ini sangat penting untuk mengarahkan lalu lintas ke bagian kanan aplikasi Anda. `"kind": "Compute"`Rute mengarahkan lalu lintas ke logika server Anda.
- ComputereSources - Gunakan bagian ini untuk menentukan Express runtime server dan titik masuk.

Selanjutnya, siapkan skrip pasca-build yang memindahkan artefak aplikasi yang dibangun ke dalam bundel `.amplify-hosting` penerapan. Struktur direktori selaras dengan spesifikasi penerapan Amplify Hosting.

Siapkan skrip pasca-build

1. Buat direktori bernama `bin` di root proyek Anda.
2. Buat file bernama `postbuild.sh` di `bin` direktori. Tambahkan konten berikut ini ke file `postbuild.sh`.

```
#!/bin/bash

rm -rf ./.amplify-hosting

mkdir -p ./.amplify-hosting/compute

cp -r ./dist ./.amplify-hosting/compute/default
cp -r ./node_modules ./.amplify-hosting/compute/default/node_modules

cp -r public ./.amplify-hosting/static

cp deploy-manifest.json ./.amplify-hosting/deploy-manifest.json
```

3. Tambahkan `postbuild` skrip ke `package.json` file Anda. File akan terlihat seperti berikut.

```
"scripts": {
  "start": "ts-node src/index.ts",
  "build": "tsc",
  "serve": "node dist/index.js",
  "postbuild": "chmod +x bin/postbuild.sh && ./bin/postbuild.sh"
}
```

4. Jalankan perintah berikut untuk membangun aplikasi Anda.

```
npm run build
```

5. (Opsional) Sesuaikan rute Anda untuk Express. Anda dapat memodifikasi rute dalam manifes penerapan agar sesuai dengan server Express Anda. Misalnya, jika Anda tidak memiliki aset statis di `public` direktori, Anda mungkin hanya memerlukan rute catch-all yang `"path": "/*"` mengarahkan ke Compute. Ini akan tergantung pada pengaturan server Anda.

Struktur direktori akhir Anda akan terlihat seperti berikut ini.

```
express-app/
### .amplify-hosting/
#   ### compute/
#   #   ### default/
#   #       ### node_modules/
#   #       ### index.js
#   ### static/
#   #   ### hello.txt
#   ### deploy-manifest.json
### bin/
#   ### .amplify-hosting/
#   #   ### compute/
#   #   #   ### default/
#   #   ### static/
#   ### postbuild.sh*
### dist/
#   ### index.js
### node_modules/
### public/
#   ### hello.txt
### src/
#   ### index.ts
### deploy-manifest.json
### package.json
### package-lock.json
### tsconfig.json
```

Menyebarkan server Anda

1. Dorong kode Anda ke repositori Git Anda, lalu terapkan aplikasi Anda ke Amplify Hosting.
2. Perbarui setelan build Anda untuk menunjuk `baseDirectory` ke `.amplify-hosting` sebagai berikut. Selama pembuatan, Amplify akan mendeteksi file manifes di `.amplify-hosting` direktori dan menerapkan server Express Anda seperti yang dikonfigurasi.

   ```
   version: 1
   frontend:
     phases:
       preBuild:
         commands:
           - nvm use 18
   ```

```
        - npm install
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .amplify-hosting
    files:
      - '**/*'
```

3. Untuk memverifikasi bahwa penerapan Anda berhasil dan server Anda berjalan dengan benar, kunjungi aplikasi Anda di URL default yang disediakan oleh Amplify Hosting.

Integrasi optimasi gambar untuk penulis kerangka kerja

Penulis kerangka kerja dapat mengintegrasikan fitur optimasi gambar Amplify dengan menggunakan spesifikasi penerapan Amplify Hosting. Untuk mengaktifkan pengoptimalan gambar, manifes penerapan Anda harus berisi aturan perutean yang menargetkan layanan pengoptimalan gambar. Contoh berikut menunjukkan cara mengkonfigurasi aturan routing.

```
// .amplify-hosting/deploy-manifest.json

{
  "routes": [
    {
      "path": "/images/*",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=31536000, immutable"
      }
    }
  ]
}
```

Untuk informasi selengkapnya tentang mengonfigurasi setelan pengoptimalan gambar menggunakan spesifikasi penerapan, lihat. Menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build

Memahami API Pengoptimalan Gambar

Pengoptimalan gambar dapat dipanggil saat runtime melalui URL domain aplikasi Amplify, di jalur yang ditentukan oleh aturan perutean.

```
GET https://{appDomainName}/{path}?{queryParams}
```

Optimalisasi gambar memberlakukan aturan berikut pada gambar.

- Amplify tidak dapat mengoptimalkan format GIF, APNG, dan SVG atau mengonversinya ke format lain.
- Gambar SVG tidak disajikan kecuali `dangerouslyAllowSVG` pengaturan diaktifkan.
- Lebar atau tinggi gambar sumber tidak boleh melebihi 11 MB atau 9.000 piksel.
- Batas ukuran gambar yang dioptimalkan adalah 4 MB.
- HTTPS adalah satu-satunya protokol yang didukung untuk sumber gambar dengan remote URLs.

Header HTTP

Header HTTP permintaan Terima digunakan untuk menentukan format gambar, dinyatakan sebagai tipe MIME, diizinkan oleh klien (biasanya browser web). Layanan optimasi gambar akan mencoba mengonversi gambar ke format yang ditentukan. Nilai yang ditentukan untuk header ini akan memiliki prioritas yang lebih tinggi daripada parameter kueri format. Misalnya, nilai yang valid untuk header Terima adalah`image/png, image/webp, */*` . Setelan format yang ditentukan dalam manifes penerapan Amplify akan membatasi format ke format yang ada dalam daftar. Bahkan jika header Terima meminta format tertentu, itu akan diabaikan jika formatnya tidak ada dalam daftar izinkan.

Parameter permintaan URI

Tabel berikut menjelaskan parameter permintaan URI untuk optimasi Gambar.

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
url	Tali	Ya	Jalur relatif atau URL absolut ke gambar sumber. Untuk URL jarak jauh, hanya protokol https yang didukung. Nilai harus dikodekan URL.	`?url=https%3A%2F%2Fwww.example.com%2Fbuffalo.png`

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
lebar	Bilangan	Ya	Lebar dalam piksel dari gambar yang dioptimalkan.	`?width=800`
tingginya	Bilangan	Tidak	Ketinggian piksel dari gambar yang dioptimal kan. Jika tidak ditentukan, gambar akan diskalaka n secara otomatis agar sesuai dengan lebarnya.	`?height=600`
cocok	Nilai enum:`cover`,,`cont` `inside` `outside`	Tidak	Bagaimana gambar diubah ukurannya agar sesuai dengan lebar dan tinggi yang ditentukan.	`?width=80 0&height= 600&fit=c over`
posisi	Nilai enum:`center`,,`top` `bottom` `left`	Tidak	Posisi yang akan digunakan saat fit adalah `cover` atau`contain`.	`?fit=cont ain&posit ion=centre`

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
memangkas	Bilangan	Tidak	Memangkas piksel dari semua tepi yang berisi nilai yang mirip dengan warna latar belakang yang ditentukan dari piksel kiri atas.	`?trim=50`
perluas	Objek	Tidak	Menambahkan piksel ke tepi gambar menggunakan warna yang berasal dari piksel tepi terdekat. Formatnya adalah `{top}_{right}_{bottom}_{left}` di mana setiap nilai adalah jumlah piksel yang akan ditambahkan.	`?extend=10_0_5_0`

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
sari	Objek	Tidak	Pangkas gambar ke persegi panjang yang ditentukan dibatasi oleh atas, kiri, lebar dan tinggi. Formatnya adalah {left} _ {top} _ {width} _ {right} di mana setiap nilai adalah jumlah piksel yang akan dipotong.	`?extract=10_0_5_0`
format	String	Tidak	Format output yang diinginkan untuk gambar yang dioptimalkan.	`?format=webp`
kualitas	Bilangan	Tidak	Kualitas gambar, dari 1 hingga 100. Hanya digunakan saat mengonversi format gambar.	`?quality=50`
merotasi	Bilangan	Tidak	Memutar gambar dengan sudut yang ditentukan dalam jumlah derajat.	`?rotate=45`

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
membalik	Boolean	Tidak	Mencerminkan gambar secara vertikal (atas-bawah) pada sumbu x. Ini selalu terjadi sebelum rotasi, jika ada.	`?flip`
gagal	Boolean	Tidak	Mencerminkan gambar secara horizontal (kiri-kanan) pada sumbu y. Ini selalu terjadi sebelum rotasi, jika ada.	`?flop`
mempertajam	Bilangan	Tidak	Penajaman meningkatkan definisi tepi pada gambar. Nilai yang valid adalah antara 0,000001 dan 10.	`?sharpen=1`
median	Bilangan	Tidak	Menerapkan filter median. Ini menghilangkan noise atau menghaluskan tepi gambar.	`?sharpen=3`

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
mengaburkan	Bilangan	Tidak	Menerapkan blur Gaussian dari sigma yang ditentukan. Nilai yang valid adalah 0,3 hingga 1.000.	`?blur=20`
gama	Bilangan	Tidak	Menerapkan koreksi gamma untuk meningkatkan kecerahan yang dirasakan dari gambar yang diubah ukurannya. Nilai harus antara 1.0 dan 3.0.	`?gamma=1`
meniadakan	Boolean	Tidak	Membalikkan warna gambar.	`?negate`
normalisasi	Boolean	Tidak	Meningkatkan kontras gambar dengan meregangkan luminansinya untuk menutupi rentang dinamis penuh.	`?normalize`

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
ambang	Bilangan	Tidak	Mengganti piksel apa pun dalam gambar dengan piksel hitam, jika intensitasnya kurang dari ambang batas yang ditentukan. Atau dengan piksel putih jika lebih besar dari ambang batas. Nilai yang valid adalah antara 0 dan 255.	`?threshold=155`
warna	String	Tidak	Mewarnai gambar menggunakan RGB yang disediakan sambil mempertahankan pencahayaan gambar.	`?tint=#7743CE`
skala abu-abu	Boolean	Tidak	Mengubah gambar menjadi skala abu-abu (hitam dan putih).	`?grayscale`

Kode status respons

Daftar berikut menjelaskan kode status respons untuk optimasi gambar.

Sukses - kode status HTTP 200

Permintaan itu terpenuhi dengan sukses.

BadRequest - Kode status HTTP 400

- Parameter kueri masukan ditentukan secara tidak benar.
- URL jarak jauh tidak terdaftar sebagaimana diizinkan dalam `remotePatterns` pengaturan.
- URL jarak jauh tidak diselesaikan ke gambar.
- Lebar atau tinggi yang diminta tidak tercantum sebagaimana diizinkan dalam `sizes` pengaturan.
- Gambar yang diminta adalah SVG tetapi `dangerouslyAllowSvg` pengaturannya dinonaktifkan.

Tidak Ditemukan - Kode status HTTP 404

Sumber gambar tidak ditemukan.

Konten terlalu besar - kode status HTTP 413

Baik gambar sumber atau gambar yang dioptimalkan melebihi ukuran maksimum yang diizinkan dalam byte.

Memahami caching gambar yang dioptimalkan

Amplify Hosting cache gambar yang dioptimalkan pada CDN kami sehingga permintaan berikutnya ke gambar yang sama, dengan parameter kueri yang sama, disajikan dari cache. Cache Time to live (TTL) dikendalikan oleh `Cache-Control` header. Daftar berikut menjelaskan pilihan Anda untuk menentukan `Cache-Control` header.

- Menggunakan `Cache-Control` kunci dalam aturan routing yang menargetkan optimasi gambar.
- Menggunakan header khusus yang ditentukan dalam aplikasi Amplify.
- Untuk gambar jarak jauh, `Cache-Control` header yang dikembalikan oleh gambar jarak jauh dihormati.

Yang `minimumCacheTTL` ditentukan dalam pengaturan optimasi gambar mendefinisikan batas bawah Cache-Control max-age arahan. Misalnya, jika URL gambar jarak jauh merespons

dengan`Cache-Control s-max-age=10`, tetapi nilainya `minimumCacheTTL` adalah 60, maka 60 digunakan.

Menggunakan adaptor open source untuk kerangka SSR apa pun

Anda dapat menggunakan adaptor build kerangka SSR apa pun yang telah dibuat untuk integrasi dengan Amplify Hosting. Setiap kerangka kerja yang menawarkan adaptor menentukan bagaimana adaptor dikonfigurasi dan terhubung ke proses pembuatannya. Biasanya, Anda akan menginstal adaptor sebagai ketergantungan pengembangan npm.

Setelah Anda membuat aplikasi dengan kerangka kerja, gunakan dokumentasi kerangka kerja untuk mempelajari cara menginstal adaptor Amplify Hosting dan mengonfigurasinya di file konfigurasi aplikasi Anda.

Selanjutnya, buat `amplify.yml` file di direktori root proyek Anda. Dalam `amplify.yml` file, setel `baseDirectory` ke direktori keluaran build aplikasi Anda. Framework menjalankan adaptor selama proses build untuk mengubah output menjadi bundel penerapan Amplify Hosting.

Nama direktori keluaran build bisa apa saja, tetapi `.amplify-hosting` nama file memiliki signifikansi. Amplify pertama-tama mencari direktori yang didefinisikan sebagai. `baseDirectory` Jika ada, Amplify mencari output build di sana. Jika direktori tidak ada, Amplify mencari keluaran build di dalamnya`.amplify-hosting`, meskipun belum ditentukan oleh pelanggan.

Berikut ini adalah contoh pengaturan build untuk aplikasi. `baseDirectory`Disetel `.amplify-hosting` untuk menunjukkan bahwa output build ada di `.amplify-hosting` folder. Selama konten `.amplify-hosting` folder sesuai dengan spesifikasi penerapan Amplify Hosting, aplikasi akan berhasil diterapkan.

```
version: 1
frontend:
  preBuild:
    commands:
      - npm install
  build:
    commands:
      - npm run build
  artifacts:
    baseDirectory: .amplify-hosting
```

Setelah aplikasi dikonfigurasi untuk menggunakan adaptor kerangka kerja, Anda dapat menerapkannya ke Amplify Hosting. Untuk petunjuk terperinci, lihat Menerapkan aplikasi SSR untuk Amplify

Menerapkan situs web statis untuk Amplify dari bucket Amazon S3

Anda dapat menggunakan integrasi antara Amplify Hosting dan Amazon S3 untuk meng-host konten situs web statis yang disimpan S3 hanya dengan beberapa klik. Menyebarkan ke Amplify Hosting memberi Anda manfaat dan fitur berikut.

- Penyebaran otomatis ke jaringan pengiriman AWS konten (CDN) yang tersedia secara global yang didukung oleh CloudFront
- Dukungan HTTPS
- Hubungkan situs web Anda dengan mudah ke domain khusus menggunakan konsol Amplify
- Bawa sertifikat SSL Kustom Anda sendiri
- Pantau situs web Anda dengan log dan CloudWatch metrik akses bawaan
- Siapkan perlindungan kata sandi untuk situs web Anda
- Buat aturan pengalihan dan penulisan ulang di konsol Amplify

Anda dapat memulai proses penerapan dari konsol Amplify, konsol, AWS CLI atau. AWS SDKs Anda hanya dapat menerapkan ke Amplify dari bucket tujuan umum Amazon S3 yang terletak di akun Anda sendiri. Amplify tidak mendukung lintas akun S3 akses ember.

Saat Anda menerapkan aplikasi dari bucket tujuan umum Amazon S3 ke Amplify Hosting AWS , biaya didasarkan pada model harga Amplify. Untuk informasi selengkapnya, silakan lihat Harga AWS Amplify.

> **Important**
>
> Amplify Hosting tidak tersedia di semua Wilayah AWS tempat Amazon S3 tersedia. Untuk menerapkan situs web statis ke Amplify Hosting, bucket tujuan umum Amazon S3 yang berisi situs web Anda harus berada di wilayah tempat Amplify tersedia. Untuk daftar wilayah tempat Amplify tersedia, lihat Amplify endpoint di bagian. Referensi Umum Amazon Web Services

Lihat topik berikut untuk mempelajari cara menerapkan dan memperbarui situs web statis dari Amazon S3 ke Amplify Hosting.

Topik

Menyebarkan situs web statis dari S3 menggunakan konsol Amplify

Gunakan petunjuk berikut untuk menerapkan situs web statis baru dari bucket tujuan umum Amazon S3 menggunakan konsol Amplify.

Untuk menerapkan situs web statis dari bucket tujuan umum Amazon S3 menggunakan konsol Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih Deploy tanpa Git.
4. Pilih Berikutnya.
5. Pada halaman Mulai penyebaran manual, lakukan hal berikut.
    a. Untuk nama Aplikasi, masukkan nama aplikasi Anda.
    b. Untuk nama Cabang, masukkan nama cabang yang akan digunakan.
6. Untuk Metode, pilih Amazon S3.
7. Untuk S3 lokasi objek yang akan dihosting, pilih Browse. Pilih bucket tujuan umum Amazon S3 yang akan digunakan, lalu pilih Pilih awalan.
8. Pilih Simpan dan deploy.

Membuat kebijakan bucket untuk menerapkan situs web statis dari S3 menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk menyebarkan situs web statis dari Amazon S3 ke Amplify Hosting. Jika Anda menerapkan situs web menggunakan SDK, Anda harus membuat kebijakan bucket sendiri yang memberikan izin Amplify Hosting untuk mengambil objek di S3 bucket.

Untuk mempelajari selengkapnya tentang cara membuat kebijakan bucket, lihat Kebijakan Bucket untuk Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Contoh kebijakan bucket berikut memberikan izin Amplify Hosting untuk mencantumkan bucket dan mengambil objek bucket untuk Amplify id aplikasi Akun AWS, dan cabang yang ditentukan.

Untuk menggunakan contoh ini:

- Ganti *amzn-s3-demo-website-bucket/prefix* dengan nama bucket dan awalan situs web Anda.
- Ganti *111122223333* dengan Akun AWS id Anda.
- Ganti *region-id* dengan tempat Wilayah AWS aplikasi Amplify Anda berada, seperti. **us-east-1**
- Ganti *app_id* dengan Amplify id aplikasi Anda. Informasi ini tersedia di konsol Amplify.
- Ganti *branch_name* dengan nama cabang Anda.

> **Note**
>
> Dalam kebijakan bucket Anda, ARN `aws:SourceArn` harus berupa cabang ARN yang disandikan URL (pengkodean persen).

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "AllowAmplifyToListPrefix_appid_branch_prefix_",
            "Effect": "Allow",
            "Principal": {
                "Service": "amplify.amazonaws.com"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::amzn-s3-demo-website-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn%3Aaws%3Aamplify%3Aregion-
id%3A111122223333%3Aapps%2Fapp_id%2Fbranches%2Fbranch_name",
                    "s3:prefix": ""
```

```
                }
            }
        },
        {
            "Sid": "AllowAmplifyToReadPrefix__appid_branch_prefix_",
            "Effect": "Allow",
            "Principal": {
                "Service": "amplify.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-website-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn%3Aaws%3Aamplify%3Aregion-
id%3A111122223333%3Aapps%2Fapp_id%2Fbranches%2Fbranch_name"
                }
            }
        },
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::amzn-s3-demo-website-bucket/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

Memperbarui situs web statis yang digunakan untuk Amplify dari S3 bucket

Jika Anda memperbarui salah satu objek untuk situs web statis secara umum S3 bucket yang dihosting di Amplify, Anda harus menerapkan ulang aplikasi ke Amplify Hosting agar perubahan diterapkan. Amplify Hosting tidak secara otomatis mendeteksi perubahan pada S3 bucket. Kami menyarankan Anda menggunakan AWS Command Line Interface (CLI) untuk memperbarui situs web Anda.

Sinkronkan pembaruan ke S3

Setelah Anda membuat perubahan pada file proyek situs web Anda, gunakan perintah s3 sync berikut untuk menyinkronkan perubahan yang Anda buat ke direktori sumber lokal Anda dengan bucket tujuan umum Amazon S3 target Anda. Untuk menggunakan contoh ini, ganti *<source>* dengan nama direktori lokal Anda dan *<target>* dengan nama bucket Amazon S3 Anda.

```
aws s3 sync <source> <target>
```

Menerapkan ulang situs web ke Amplify Hosting

Gunakan perintah amplify start-deployment berikut untuk menerapkan ulang aplikasi yang diperbarui di bucket Amazon S3 untuk Amplify Hosting. Untuk menggunakan contoh ini, ganti *<app_id>* dengan id aplikasi Amplify Anda, *<branch_name>* dengan nama cabang Anda, dan *s3://amzn-s3-demo-website-bucket/prefix* dengan S3 ember dan awalan.

```
aws amplify start-deployment --app-id <app_id> --branch-name <branch_name> --source-
url s3://amzn-s3-demo-website-bucket/prefix --source-url-type BUCKET_PREFIX
```

Memperbarui sebuah S3 penerapan untuk menggunakan bucket dan awalan alih-alih file.zip

Jika Anda sudah memiliki situs web statis yang sudah digunakan untuk Amplify Hosting dari file.zip di bucket tujuan umum Amazon S3, Anda dapat memperbarui penerapan aplikasi untuk menggunakan nama bucket dan awalan yang berisi objek yang akan dihosting. Jenis penerapan ini menghilangkan kebutuhan untuk mengunggah file terpisah ke bucket Anda yang berisi konten zip keluaran build.

Untuk memigrasikan situs web statis dari file.zip ke isi bucket

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Pada halaman Semua aplikasi, pilih nama aplikasi yang digunakan secara manual yang ingin Anda migrasi dari menggunakan file.zip ke menggunakan file aplikasi secara langsung.
3. Pada halaman Ikhtisar aplikasi, pilih Menyebarkan pembaruan.
4. Pada halaman Terapkan pembaruan, untuk Metode, pilih Amazon S3.

5. Untuk S3 lokasi objek yang akan dihosting, pilih Browse. Pilih bucket yang akan digunakan, lalu pilih Pilih awalan.
6. Pilih Simpan dan deploy.

Menyebarkan aplikasi ke Amplify tanpa repositori Git

Penerapan manual memungkinkan Anda memublikasikan aplikasi web Anda dengan Amplify Hosting tanpa menghubungkan penyedia Git. Anda dapat menarik dan melepas folder zip dari desktop Anda dan meng-host situs Anda dalam hitungan detik. Atau, Anda dapat mereferensikan aset di bucket Amazon S3 atau menentukan URL publik ke lokasi penyimpanan file Anda.

> **Note**
>
> Penerapan manual memiliki batas ukuran file.zip maksimum 5GB karena kendala operasi salinan Amazon S3. Jika ada artefak build yang melebihi ukuran ini, pertimbangkan untuk memecahnya menjadi arsip yang lebih kecil atau menggunakan metode penerapan alternatif.

Untuk Amazon S3, Anda juga dapat mengatur AWS Lambda pemicu untuk memperbarui situs Anda setiap kali aset baru diunggah. Lihat Menerapkan file yang disimpan di Amazon S3, Dropbox, atau Desktop Anda ke postingan blog konsol untuk AWS Amplify detail selengkapnya tentang pengaturan skenario ini.

Amplify Hosting tidak mendukung penerapan manual untuk aplikasi yang dirender sisi server (SSR). Untuk informasi selengkapnya, lihat Menyebarkan aplikasi yang dirender sisi server dengan Amplify Hosting.

Seret dan lepas penerapan manual

Cara men-deploy aplikasi secara manual dengan seret dan jatuhkan

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Di sudut kanan atas, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih Deploy tanpa Git. Lalu, pilih Selanjutnya.
4. Pada halaman Mulai penerapan manual, untuk nama Aplikasi, masukkan nama aplikasi Anda.
5. Untuk nama Cabang, masukkan nama yang bermakna, seperti **`development`** atau**`production`**.
6. Untuk Metode, pilih Seret dan jatuhkan.

7. Seret dan lepas folder dari desktop Anda ke zona drop atau gunakan folder Choose .zip untuk memilih file dari komputer Anda. File yang Anda seret dan lepas atau pilih harus berupa folder zip yang berisi konten keluaran build Anda.
8. Pilih Simpan dan deploy.

Amazon S3 atau penyebaran manual URL

> **Note**
>
> Jika Anda menggunakan situs web statis dari S3, prosedur berikut mengharuskan Anda mengunggah folder zip dengan konten keluaran build Anda ke S3 bucket. Kami menyarankan Anda menyebarkan situs web statis langsung dari S3 menggunakan nama bucket dan awalan. Untuk informasi lebih lanjut tentang proses yang disederhanakan ini, lihatMenerapkan situs web statis untuk Amplify dari bucket Amazon S3.

Cara men-deploy aplikasi secara manual dari Amazon S3 atau URL publik

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Di sudut kanan atas, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih Deploy tanpa Git. Lalu, pilih Selanjutnya.
4. Pada halaman Mulai penerapan manual, untuk nama Aplikasi, masukkan nama aplikasi Anda.
5. Untuk nama Cabang, masukkan nama yang bermakna, seperti **`development`** atau**`production`**.
6. Untuk Metode, pilih Amazon S3 atau URL mana saja.
7. Langkah untuk mengunggah file bergantung pada metode pengunggahan.
   - Amazon S3
     a. Untuk S3 location of objects to host, pilih Browse S3. Kemudian, pilih nama bucket Amazon S3 dari daftar. Daftar kontrol akses (ACLs) harus diaktifkan untuk bucket yang Anda pilih. Untuk informasi selengkapnya, lihat Memecahkan masalah akses bucket Amazon S3 untuk penerapan manual.
     b. Pilih nama file.zip yang akan digunakan.
     c. Pilih Pilih awalan.
   - URL mana saja

- Untuk URL Sumber Daya, masukkan URL ke file.zip yang akan digunakan.

8. Pilih Simpan dan deploy.

> **Note**
>
> Saat Anda membuat folder zip, pastikan Anda zip konten output build Anda dan bukan folder tingkat atas. Misalnya, jika output build menghasilkan folder bernama “build” atau “publik”, navigasi ke folder tersebut terlebih dahulu, kemudian pilih semua isi, lalu zip dari sana. Jika tidak, pesan kesalahan “Akses Ditolak” akan ditampilkan karena direktori root situs tidak akan diinisialisasi dengan benar.

Memecahkan masalah akses bucket Amazon S3 untuk penerapan manual

Saat membuat bucket Amazon S3, Anda menggunakan setelan Kepemilikan Objek Amazon S3 untuk mengontrol apakah daftar kontrol akses ACLs () diaktifkan atau dinonaktifkan untuk bucket. Untuk menerapkan aplikasi secara manual ke Amplify dari bucket Amazon S3 ACLs , harus diaktifkan di bucket.

Jika Anda mendapatkan `AccessControlList` kesalahan saat menerapkan dari bucket Amazon S3, bucket dibuat ACLs dengan dinonaktifkan dan Anda harus mengaktifkannya di konsol Amazon S3. Untuk petunjuknya, lihat Menyetel Kepemilikan Objek pada bucket yang ada di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Mengelola konfigurasi build untuk aplikasi Amplify

Anda dapat menyesuaikan setelan dan konfigurasi build untuk penerapan Amplify. Saat Anda menerapkan aplikasi, Amplify secara otomatis mendeteksi kerangka kerja frontend dan setelan build terkait. Anda dapat menyesuaikan setelan build dalam spesifikasi build (buildspec) aplikasi untuk menambahkan variabel lingkungan, menjalankan perintah build, dan menentukan dependensi build.

Image build default Amplify hadir dengan beberapa paket dan dependensi yang sudah terinstal, tetapi Anda juga dapat menggunakan fitur pembaruan paket langsung untuk menentukan versi tertentu, atau memastikan bahwa versi terbaru selalu terinstal. Jika memiliki dependensi tertentu yang memerlukan waktu instalasi lama selama build menggunakan kontainer default Amplify, Anda dapat membuat gambar build khusus. Anda juga dapat menyesuaikan ukuran instans build untuk menyediakan penerapan aplikasi dengan CPU, memori, dan sumber daya ruang disk yang dibutuhkan.

Build dimulai secara otomatis dengan setiap komit ke repositori Git Anda dan dengan setiap penerapan baru. Anda dapat mengatur fitur webhook yang masuk untuk memulai build tanpa komit ke repositori Git Anda.

Fitur pemberitahuan build memungkinkan Anda berbagi informasi dengan anggota tim tentang keberhasilan dan kegagalan build.

Topik

Mengonfigurasi pengaturan build untuk aplikasi Amplify

Saat Anda men-deploy aplikasi, Amplify secara otomatis mendeteksi kerangka kerja frontend dan pengaturan build terkait dengan memeriksa `package.json` file aplikasi di repositori Git Anda. Anda memiliki opsi berikut untuk menyimpan pengaturan build aplikasi:

- Simpan pengaturan build di konsol Amplify - Konsol Amplify secara otomatis mendeteksi pengaturan build dan menyimpannya agar dapat diakses oleh konsol Amplify. Amplify menerapkan

pengaturan ini ke semua cabang Anda, kecuali ada file `amplify.yml` yang disimpan di repositori Anda.

- Simpan pengaturan build di repositori Anda - Unduh file `amplify.yml` dan tambahkan ke root repositori Anda.

> **Note**
>
> Pengaturan build dapat dilihat di menu Hosting konsol Amplify hanya jika aplikasi diatur untuk deployment kontinu dan terhubung ke repositori git. Untuk langkah-langkah seputar jenis deployment ini, lihat Memulai.

Membangun referensi spesifikasi

Spesifikasi build (buildspec) untuk aplikasi Amplify adalah kumpulan pengaturan YAMAL dan perintah build yang Amplify gunakan untuk menjalankan build Anda. Daftar berikut menjelaskan pengaturan ini dan bagaimana mereka digunakan.

versi

Nomor versi YAMLAmplify.

appRoot

Jalur dalam repositori tempat aplikasi berada. Diabaikan kecuali beberapa aplikasi didefinisikan.

env

Tambahkan variabel lingkungan ke bagian ini. Anda juga dapat menambahkan variabel lingkungan menggunakan konsol.

backend

Menjalankan perintah CLI Amplify untuk menyediakan backend, memperbarui fungsi Lambda, atau skema GraphQL sebagai bagian dari deployment kontinu.

frontend

Menjalankan perintah build frontend.

pengujian

Menjalankan perintah selama fase tes. Pelajari cara menambahkan tes ke aplikasi.

fase membangun

Frontend, backend, dan tes terdiri dari tiga fase yang mewakili perintah yang dijalankan selama setiap urutan build.

- preBuild - Skrip preBuild berjalan sebelum build sebenarnya dimulai, tetapi setelah Amplify menginstal dependensi.
- build - Perintah build Anda.
- postBuild - Skrip post-build berjalan setelah build selesai dan Amplify telah menyalin semua artefak yang diperlukan ke direktori output.

buildpath

Jalur yang digunakan untuk menjalankan build. Amplify menggunakan jalur ini untuk menemukan artefak build Anda. Jika Anda tidak menentukan jalur, Amplify menggunakan root aplikasi monorepo. `apps/app`

artefak>direktori dasar

Direktori tempat artefak build Anda disimpan.

artefak > file

Tentukan file dari artefak yang ingin Anda terapkan. Masukkan `**/*` untuk memasukkan semua file.

cache

Menentukan dependensi build-time seperti folder node_modules. Selama build pertama, jalur yang disediakan di sini di-cache. Pada build berikutnya, Amplify mengembalikan cache ke jalur yang sama sebelum menjalankan perintah Anda.

Amplify menganggap semua jalur cache yang disediakan relatif terhadap root proyek Anda. Namun, Amplify tidak mengizinkan melintasi di luar root proyek. Misalnya, jika Anda menentukan jalur absolut, build akan berhasil tanpa kesalahan, tetapi jalur tidak akan di-cache.

Sintaks YAMAL spesifikasi build

Contoh spesifikasi build berikut menunjukkan sintaks YAMAL dasar.

```
version: 1
env:
  variables:
    key: value
```

```
backend:
  phases:
    preBuild:
      commands:
        - *enter command*
    build:
      commands:
        - *enter command*
    postBuild:
        commands:
        - *enter command*
frontend:
  buildpath:
  phases:
    preBuild:
      commands:
        - cd react-app
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    files:
        - location
        - location
    discard-paths: yes
    baseDirectory: location
  cache:
    paths:
        - path # A cache path relative to the project root
        - path # Traversing outside of the project root is not allowed
test:
  phases:
    preTest:
      commands:
        - *enter command*
    test:
      commands:
        - *enter command*
    postTest:
      commands:
        - *enter command*
  artifacts:
    files:
```

```
        - location
        - location
    configFilePath: *location*
    baseDirectory: *location*
```

Mengedit spesifikasi build

Anda dapat menyesuaikan setelan build aplikasi dengan mengedit spesifikasi build (buildspec) di konsol Amplify. Pengaturan build diterapkan ke semua cabang di aplikasi Anda, kecuali untuk cabang dengan `amplify.yml` file yang disimpan di repositori Git.

Untuk mengedit pengaturan build di konsol Amplify

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang pengaturan build akan Anda edit.
3. Di panel navigasi, pilih Pengaturan build.
4. Pada halaman Pengaturan build, di bagian Spesifikasi pembuatan aplikasi, pilih Edit.
5. Di jendela Edit build spec, masukkan pembaruan Anda.
6. Pilih Simpan.

Anda dapat menggunakan contoh yang dijelaskan dalam topik berikut untuk memperbarui setelan build untuk skenario tertentu.

Topik

- Mengonfigurasi pengaturan build khusus cabang dengan skrip
- Mengatur perintah untuk menavigasi ke subfolder
- Men-deploy backend dengan front end untuk aplikasi Gen 1
- Menyiapkan folder output
- Menginstal paket sebagai bagian dari build
- Menggunakan registri npm privat
- Menginstal paket OS
- Menetapkan penyimpanan kunci-nilai untuk setiap build
- Melompati build untuk penerapan
- Mematikan build otomatis pada setiap komit

- Mengonfigurasi build dan deploy frontend berbasis diff
- Mengonfigurasi build backend berbasis diff untuk aplikasi Gen 1

Mengonfigurasi pengaturan build khusus cabang dengan skrip

Anda dapat menggunakan penulisan bash shell untuk menentukan pengaturan build khusus cabang. Sebagai contoh, skrip berikut menggunakan variabel lingkungan sistem $ AWS_BRANCH untuk menjalankan serangkaian perintah jika nama cabang main dan serangkaian perintah lain jika nama cabang dev.

```
frontend:
  phases:
    build:
      commands:
        - if [ "${AWS_BRANCH}" = "main" ]; then echo "main branch"; fi
        - if [ "${AWS_BRANCH}" = "dev" ]; then echo "dev branch"; fi
```

Mengatur perintah untuk menavigasi ke subfolder

Untuk monorepo, pengguna ingin `cd` masuk ke dalam folder untuk menjalankan build. Setelah Anda menjalankan `cd` perintah, perintah akan berlaku di semua tahap build sehingga Anda tidak perlu mengulangi perintah di fase berbeda.

```
version: 1
env:
  variables:
    key: value
frontend:
  phases:
    preBuild:
      commands:
        - cd react-app
        - npm ci
    build:
      commands:
        - npm run build
```

Men-deploy backend dengan front end untuk aplikasi Gen 1

> **Note**
>
> Bagian ini hanya berlaku untuk aplikasi Amplify Gen 1. Backend Gen 1 dibuat menggunakan Amplify Studio dan antarmuka baris perintah Amplify (CLI).

`amplifyPush`Perintah adalah skrip pembantu yang membantu Anda menjalankan deployment backend. Pengaturan build berikut secara otomatis menentukan lingkungan backend yang tepat untuk di-deploy di cabang saat ini.

```
version: 1
env:
  variables:
    key: value
backend:
  phases:
    build:
      commands:
        - amplifyPush --simple
```

Menyiapkan folder output

Pengaturan build berikut mengatur direktori output ke folder publik.

```
frontend:
  phases:
    commands:
      build:
        - yarn run build
  artifacts:
    baseDirectory: public
```

Menginstal paket sebagai bagian dari build

Anda dapat menggunakan `yarn` perintah `npm` or untuk menginstal paket selama build.

```
frontend:
```

```
  phases:
    build:
      commands:
        - npm install -g <package>
        - <package> deploy
        - yarn run build
  artifacts:
    baseDirectory: public
```

Menggunakan registri npm privat

Anda dapat menambahkan referensi ke registri privat di pengaturan build Anda atau menambahkannya sebagai variabel lingkungan.

```
build:
  phases:
    preBuild:
      commands:
        - npm config set <key> <value>
        - npm config set registry https://registry.npmjs.org
        - npm config set always-auth true
        - npm config set email hello@amplifyapp.com
        - yarn install
```

Menginstal paket OS

Gambar AL2 023 Amplify menjalankan kode Anda dengan nama pengguna yang tidak memiliki hak istimewa. `amplify` Amplify memberikan hak istimewa kepada pengguna ini untuk menjalankan perintah OS menggunakan perintah Linux. `sudo` Jika Anda ingin menginstal paket OS untuk dependensi yang hilang, Anda dapat menggunakan perintah seperti `yum` dan `rpm` dengan. `sudo`

Contoh bagian build berikut menunjukkan sintaks untuk menginstal paket OS menggunakan perintah. `sudo`

```
build:
  phases:
    preBuild:
      commands:
        - sudo yum install -y <package>
```

Menetapkan penyimpanan kunci-nilai untuk setiap build

`envCache` menyediakan penyimpanan kunci-nilai pada waktu build. Nilai-nilai yang disimpan di `envCache` hanya dapat diubah selama build dan dapat digunakan kembali pada build berikutnya. Dengan `envCache`, informasi dapat disimpan di lingkungan yang di-deploy dan tersedia untuk kontainer build di build yang berturutan. Tidak seperti nilai yang disimpan di `envCache`, perubahan pada variabel lingkungan selama build tidak muncul pada build selanjutnya.

Contoh penggunaan:

```
envCache --set <key> <value>
envCache --get <key>
```

Melompati build untuk penerapan

Untuk melompati build otomatis pada penerapan tertentu, sertakan teks [skip-cd] di akhir pesan penerapan.

Mematikan build otomatis pada setiap komit

Anda dapat mengonfigurasi Amplify untuk menonaktifkan build otomatis di setiap penerapan kode. Untuk mengatur, pilih Pengaturan aplikasi, Pengaturan cabang, kemudian temukan bagian Cabang yang mencantumkan cabang-cabang yang terhubung. Pilih cabang, lalu pilih Tindakan, Nonaktifkan build otomatis. Penerapan baru untuk cabang tersebut tidak akan lagi memulai build baru.

Mengonfigurasi build dan deploy frontend berbasis diff

Anda dapat mengonfigurasi Amplify agar menggunakan build frontend berbasis diff. Jika diaktifkan, Amplify mencoba untuk menjalankan diff di folder Anda `appRoot` atau `/src/` folder secara default di awal setiap build. Jika tidak menemukan perbedaan apa pun, Amplify akan melompati langkah build, tes (jika dikonfigurasi), dan deploy frontend, serta tidak memperbarui aplikasi yang di-hosting.

Untuk mengonfigurasi build dan deploy frontend berbasis diff

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi tempat konfigurasi build dan deploy frontend berbasis diff dilakukan.
3. Di panel navigasi, pilih Hosting, variabel Lingkungan.
4. Di bagian Variabel lingkungan, pilih Kelola variabel.

5. Langkah untuk mengonfigurasi variabel lingkungan bervariasi, bergantung pada apakah Anda mengaktifkan atau menonaktifkan build dan deploy frontend berbasis diff.
   - Untuk mengaktifkan build dan deploy frontend berbasis diff
     a. Di bagian Kelola variabel, bagian Variabel, masukkan `AMPLIFY_DIFF_DEPLOY`.
     b. Untuk Nilai, masukkan `true`.
   - Untuk menonaktifkan build dan deploy frontend berbasis diff
     - Lakukan salah satu dari berikut ini:
       - Di bagian Kelola variabel, cari `AMPLIFY_DIFF_DEPLOY`. Untuk Nilai, masukkan `false`.
       - Hapus variabel lingkungan `AMPLIFY_DIFF_DEPLOY`.
6. Pilih Simpan.

Anda juga dapat mengatur variabel lingkungan `AMPLIFY_DIFF_DEPLOY_ROOT` agar menimpa path default dengan path yang berkaitan dengan root repo Anda, seperti `dist`.

Mengonfigurasi build backend berbasis diff untuk aplikasi Gen 1

> **Note**
>
> Bagian ini hanya berlaku untuk aplikasi Amplify Gen 1. Backend Gen 1 dibuat menggunakan Amplify Studio dan antarmuka baris perintah Amplify (CLI).

Anda dapat mengonfigurasi Amplify agar menggunakan build backend berbasis diff menggunakan variabel lingkungan. `AMPLIFY_DIFF_BACKEND` Saat Anda mengaktifkan build backend berbasis diff, di awal setiap build Amplify mencoba untuk menjalankan diff di folder dalam repositori Anda. `amplify` Jika tidak menemukan perbedaan apa pun, Amplify akan melompati langkah build backend, dan tidak memperbarui sumber daya backend Anda. Jika proyek Anda tidak memiliki folder `amplify` di repositori, Amplify akan mengabaikan nilai variabel lingkungan `AMPLIFY_DIFF_BACKEND`.

Jika saat ini Anda memiliki perintah khusus yang ditentukan dalam pengaturan build fase backend Anda, build backend bersyarat tidak akan berfungsi. Jika Anda ingin perintah kustom tersebut berjalan, Anda harus memindahkannya ke fase frontend setelan build di `amplify.yml` file aplikasi Anda.

Untuk mengonfigurasi build backend berbasis diff

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi tempat konfigurasi build backend berbasis diff dilakukan.
3. Di panel navigasi, pilih Hosting, variabel Lingkungan.
4. Di bagian Variabel lingkungan, pilih Kelola variabel.
5. Langkah untuk mengonfigurasi variabel lingkungan bervariasi, bergantung pada apakah Anda mengaktifkan atau menonaktifkan build backend berbasis diff.
   - Untuk mengaktifkan build backend berbasis diff
     a. Di bagian Kelola variabel, bagian Variabel, masukkan `AMPLIFY_DIFF_BACKEND`.
     b. Untuk Nilai, masukkan `true`.
   - Untuk menonaktifkan build backend berbasis diff
     - Lakukan salah satu dari berikut ini:
       - Di bagian Kelola variabel, cari `AMPLIFY_DIFF_BACKEND`. Untuk Nilai, masukkan `false`.
       - Hapus variabel lingkungan `AMPLIFY_DIFF_BACKEND`.
6. Pilih Simpan.

Mengonfigurasi pengaturan build monorepo

Monorepo adalah tempat untuk menyimpan beberapa proyek atau layanan mikro dalam satu repositori. Anda dapat menggunakan Amplify untuk men-deploy aplikasi dalam monorepo tanpa membuat beberapa konfigurasi build atau konfigurasi cabang.

Amplify mendukung aplikasi dalam monorepos generik serta aplikasi di monorepos yang dibuat menggunakan ruang kerja npm, ruang kerja pnpm, ruang kerja Yarn, Nx, dan Turborepo. Ketika Anda men-deploy aplikasi, Amplify secara otomatis mendeteksi alat build monorepo yang Anda gunakan. Amplify secara otomatis menerapkan pengaturan build untuk aplikasi di ruang kerja npm, ruang kerja Yarn, atau Nx. Aplikasi Turborepo dan pnpm memerlukan konfigurasi tambahan. Untuk informasi selengkapnya, lihat Mengkonfigurasi aplikasi Turborepo dan pnpm monorepo.

Anda dapat menyimpan pengaturan build untuk monorepo di konsol Amplify atau mengunduh `amplify.yml` file dan menambahkannya ke root repositori Anda. Amplify menerapkan pengaturan yang disimpan di konsol ke semua cabang, kecuali ditemukan file `amplify.yml` di repositori Anda.

Ketika ada `amplify.yml` file, pengaturannya menimpa pengaturan build yang disimpan di konsol Amplify.

Sintaks YAMAL spesifikasi build monorepo

Sintaks YAML untuk spesifikasi build monorepo tidak sama dengan sintaks YAML untuk repo yang berisi satu aplikasi. Untuk monorepo, Anda menyatakan setiap proyek dalam daftar aplikasi. Anda harus memberikan `appRoot` kunci tambahan berikut untuk setiap aplikasi yang Anda nyatakan dalam spesifikasi build monorepo:

appRoot

> Root, dalam repositori, tempat aplikasi dimulai. Kunci ini harus ada, dan memiliki nilai yang sama dengan variabel lingkungan `AMPLIFY_MONOREPO_APP_ROOT`. Untuk langkah-langkah seputar pengaturan variabel lingkungan ini, lihat Mengatur variabel lingkungan AMPLIFY_MONOREPO_APP_ROOT.

Contoh spesifikasi build monorepo berikut menjelaskan cara menyatakan beberapa aplikasi Amplify di repo yang sama. Dua aplikasi, `react-app` dan `angular-app`, dinyatakan dalam daftar `applications`. Kunci `appRoot` untuk setiap aplikasi menunjukkan bahwa aplikasi terletak di folder root `apps` dalam repo.

`buildpath`Atribut diatur `/` untuk menjalankan dan membangun aplikasi dari root proyek monorepo. `baseDirectory`Atribut adalah jalur relatif dari`buildpath`.

Sintaks YAML spesifikasi build monorepo

```
version: 1
applications:
  - appRoot: apps/react-app
    env:
      variables:
        key: value
    backend:
      phases:
        preBuild:
          commands:
            - *enter command*
        build:
          commands:
            - *enter command*
```

```
        postBuild:
            commands:
            - *enter command*
    frontend:
      buildPath: / # Run install and build from the monorepo project root
      phases:
        preBuild:
          commands:
            - *enter command*
            - *enter command*
        build:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        discard-paths: yes
        baseDirectory: location
      cache:
        paths:
            - path
            - path
    test:
      phases:
        preTest:
          commands:
            - *enter command*
        test:
          commands:
            - *enter command*
        postTest:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        configFilePath: *location*
        baseDirectory: *location*
  - appRoot: apps/angular-app
    env:
      variables:
        key: value
```

```
    backend:
      phases:
        preBuild:
          commands:
            - *enter command*
        build:
          commands:
            - *enter command*
        postBuild:
            commands:
            - *enter command*
    frontend:
      phases:
        preBuild:
          commands:
            - *enter command*
            - *enter command*
        build:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        discard-paths: yes
        baseDirectory: location
      cache:
        paths:
            - path
            - path
    test:
      phases:
        preTest:
          commands:
            - *enter command*
        test:
          commands:
            - *enter command*
        postTest:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
```

```
            - location
        configFilePath: *location*
        baseDirectory: *location*
```

Aplikasi yang menggunakan contoh spesifikasi build berikut, akan dibangun di bawah root proyek dan artefak build akan ditempatkan di`/packages/nextjs-app/.next`.

```
applications:
  - frontend:
      buildPath: '/'  # run install and build from monorepo project root
      phases:
        preBuild:
          commands:
            - npm install
        build:
          commands:
            - npm run build --workspace=nextjs-app
      artifacts:
        baseDirectory: packages/nextjs-app/.next
        files:
          - '**/*'
      cache:
        paths:
          - node_modules/**/*
    appRoot: packages/nextjs-app
```

Mengatur variabel lingkungan AMPLIFY_MONOREPO_APP_ROOT

Saat Anda men-deploy aplikasi yang disimpan dalam monorepo, variabel lingkungan `AMPLIFY_MONOREPO_APP_ROOT` aplikasi harus memiliki nilai yang sama dengan path root aplikasi, bergantung pada root repositori. Misalnya, monorepo bernama `ExampleMonorepo` dengan folder root bernama `apps`, yang berisi `app1`, `app2`, dan `app3` memiliki struktur direktori berikut:

```
ExampleMonorepo
  apps
    app1
    app2
    app3
```

Dalam contoh ini, nilai variabel lingkungan `AMPLIFY_MONOREPO_APP_ROOT` untuk `app1` adalah `apps/app1`.

Ketika Anda men-deploy aplikasi monorepo menggunakan konsol Amplify, konsol secara otomatis menetapkan variabel lingkungan `AMPLIFY_MONOREPO_APP_ROOT` menggunakan nilai yang Anda tentukan untuk path ke root aplikasi. Namun, jika aplikasi monorepo Anda sudah ada di Amplify atau di-deploy menggunakan AWS CloudFormation, Anda harus secara manual mengatur variabel `AMPLIFY_MONOREPO_APP_ROOT` lingkungan di bagian Variabel lingkungan dalam konsol Amplify.

Mengatur variabel lingkungan AMPLIFY_MONOREPO_APP_ROOT secara otomatis selama deployment

Instruksi berikut menjelaskan cara men-deploy aplikasi monorepo dengan konsol Amplify. Amplify secara otomatis menetapkan variabel lingkungan `AMPLIFY_MONOREPO_APP_ROOT` menggunakan folder root aplikasi yang Anda tentukan di konsol.

Untuk men-deploy aplikasi monorepo dengan konsol Amplify

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih Buat aplikasi baru di sudut kanan atas.
3. Pada halaman Mulai membangun dengan Amplify, pilih penyedia Git Anda, lalu pilih Berikutnya.
4. Di halaman Tambahkan cabang repositori, lakukan langkah berikut:
    a. Pilih nama repositori Anda dari daftar.
    b. Pilih nama cabang yang akan digunakan.
    c. Pilih Aplikasi saya adalah monorepo
    d. Masukkan path ke aplikasi di monorepo Anda, misalnya, **`apps/app1`**.
    e. Pilih Berikutnya.
5. Di halaman Pengaturan aplikasi, Anda dapat menggunakan pengaturan default atau menyesuaikan pengaturan build untuk aplikasi. Di bagian variabel Lingkungan, Amplify menetapkan `AMPLIFY_MONOREPO_APP_ROOT` ke jalur yang Anda tentukan di langkah 4d.
6. Pilih Berikutnya.
7. Di halaman Tinjauan, pilih Simpan dan deploy.

Mengatur variabel lingkungan AMPLIFY_MONOREPO_APP_ROOT untuk aplikasi yang ada

Ikuti langkah-langkah berikut untuk mengatur variabel `AMPLIFY_MONOREPO_APP_ROOT` lingkungan untuk aplikasi yang telah di-deploy ke Amplify atau telah dibuat menggunakan variabel lingkungan untuk aplikasi yang telah di-deploy ke Amplify atau telah dibuat menggunakan. CloudFormation

Untuk mengatur variabel lingkungan AMPLIFY_MONOREPO_APP_ROOT untuk aplikasi yang ada

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih nama aplikasi yang variabel lingkungannya akan diatur.
3. Di panel navigasi, pilih Hosting, lalu pilih Variabel lingkungan.
4. Di halaman Variabel lingkungan, pilih Kelola variabel.
5. Di bagian Kelola variabel, lakukan langkah-langkah berikut:
   a. Pilih Tambahkan baru.
   b. Untuk Variabel, masukkan kunci `AMPLIFY_MONOREPO_APP_ROOT`.
   c. Untuk Nilai, masukkan path ke aplikasi, misalnya **`apps/app1`**.
   d. Untuk Cabang, Amplify secara default menerapkan variabel lingkungan ke semua cabang.
6. Pilih Simpan.

Mengkonfigurasi aplikasi Turborepo dan pnpm monorepo

Alat pembuatan monorepo ruang kerja Turborepo dan pnpm mendapatkan informasi konfigurasi dari file. `.npmrc` Saat Anda menerapkan aplikasi monorepo yang dibuat dengan salah satu alat ini, Anda harus memiliki `.npmrc` file di direktori root proyek Anda.

Dalam `.npmrc` file, atur linker untuk menginstal paket Node ke`hoisted`. Anda dapat menyalin baris berikut ke file Anda.

```
node-linker=hoisted
```

Untuk informasi selengkapnya tentang *`.npmrc`* file dan pengaturan, lihat pnpm .npmrc di dokumentasi pnpm.

Pnpm tidak disertakan dalam container build default Amplify. Untuk ruang kerja pnpm dan aplikasi Turborepo, Anda harus menambahkan perintah untuk menginstal pnpm di fase pengaturan build aplikasi Anda. `preBuild`

Contoh kutipan berikut dari spesifikasi build menunjukkan `preBuild` fase dengan perintah untuk menginstal pnpm.

```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
          commands:
            - npm install -g pnpm
```

Menyesuaikan gambar build

Anda dapat menggunakan gambar build kustom untuk menyediakan lingkungan build khusus untuk aplikasi Amplify. Jika memiliki dependensi tertentu yang memerlukan waktu instalasi lama selama build menggunakan kontainer default Amplify, Anda dapat membuat gambar Docker sendiri dan menggunakannya sebagai referensi selama build. Gambar dapat di-hosting di Amazon Elastic Container Registry Public.

Agar dapat bekerja sebagai gambar build Amplify, gambar build kustom harus memenuhi persyaratan berikut.

Ketentuan gambar build kustom

1. Distribusi Linux yang mendukung GNU C Library (glibc), seperti Amazon Linux, dikompilasi untuk arsitektur x86-64.
2. cURL: Ketika meluncurkan gambar kustom Anda, kami mengunduh build runner kami ke kontainer Anda sehingga harus ada cURL. Jika dependensi ini hilang, build langsung gagal tanpa output karena build-runner kami tidak mampu menghasilkan output apa pun.
3. Git: Git harus terinstal pada gambar untuk membuat klon repositori Git Anda. Jika dependensi ini hilang, langkah repositori Kloning akan gagal.
4. OpenSSH: Agar repositori Anda dapat dikloning dengan aman, OpenSSH harus menyiapkan sementara kunci SSH selama build. Paket OpenSSH menyediakan perintah yang dibutuhkan build runner untuk melakukan ini.
5. Bash dan The Bourne Shell: Kedua utilitas ini digunakan untuk menjalankan perintah pada waktu pembuatan. Jika tidak diinstal, build Anda mungkin gagal sebelum memulai.
6. Node.js+npm: Pelari build kami tidak menginstal Node. Sebaliknya, itu bergantung pada Node dan NPM yang diinstal pada gambar. Ini hanya diperlukan untuk build yang memerlukan paket NPM

atau perintah terkait Node. Namun, kami sangat menyarankan untuk menginstalnya karena ketika ada, Amplify build runner dapat menggunakan alat ini untuk meningkatkan eksekusi build. Fitur penggantian paket Amplify menggunakan NPM untuk menginstal paket Hugo-Extended saat Anda mengatur penggantian untuk Hugo.

Paket berikut tidak diperlukan, tetapi kami sangat menyarankan Anda menginstalnya.

1. NVM (Node Version Manager): Kami menyarankan Anda menginstal pengelola versi ini jika Anda perlu menangani versi yang berbeda dariNode. Saat Anda menyetel override, fitur penggantian paket Amplify digunakan NVM untuk mengubah versi Node.js sebelum setiap build.
2. Wget: Amplify dapat menggunakan Wget utilitas untuk mengunduh file selama proses pembuatan. Kami menyarankan Anda menginstalnya di gambar kustom Anda.
3. Tar: Amplify dapat menggunakan Tar utilitas untuk membuka kompres file yang diunduh selama proses pembuatan. Kami menyarankan Anda menginstalnya di gambar kustom Anda.

Mengonfigurasi gambar build kustom

Gunakan langkah-langkah berikut untuk mengonfigurasi gambar build kustom untuk aplikasi di konsol Amplify.

Untuk mengonfigurasi gambar build kustom yang di-hosting di Amazon ECR

1. Lihat Memulai di Panduan pengguna Amazon ECR Public untuk mengatur repositori Amazon ECR Public dengan gambar Docker.
2. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
3. Pilih aplikasi yang gambar build kustomnya akan dikonfigurasi.
4. Di panel navigasi, pilih Hosting, Build settings.
5. Di halaman Pengaturan build, di bagian Pengaturan gambar build, pilih Edit.
6. Pada halaman Edit pengaturan gambar build, perluas menu Build image, dan pilih Custom Build Image.
7. Masukkan nama repo Amazon ECR Public yang Anda buat di langkah pertama. Di sinilah gambar build Anda di-hosting. Misalnya, jika nama repo ecr-examplerepo, Anda harus memasukkan **`public.ecr.aws/xxxxxxxx/ecr-examplerepo`**.
8. Pilih Simpan.

Menggunakan versi paket dan dependensi tertentu dalam image build

Pembaruan paket langsung memungkinkan Anda untuk menentukan versi paket dan dependensi yang digunakan dalam gambar build default Amplify. Gambar build default dilengkapi dengan beberapa paket dan dependensi yang telah terinstal sebelumnya (misalnya Hugo, CLI Amplify, Yarn, dll.). Dengan pembaruan paket langsung, Anda dapat menimpa versi dependensi ini dan menentukan versi tertentu, atau memastikan bahwa versi terbaru selalu terinstal.

Jika pembaruan paket langsung diaktifkan, sebelum build berjalan, build runner akan terlebih dahulu memperbarui (atau menurunkan versi) dependensi terkait. Cara ini akan meningkatkan waktu build sesuai dengan waktu yang diperlukan untuk memperbarui dependensi, tetapi, kelebihannya, Anda dapat memastikan versi dependensi yang sama digunakan untuk membangun aplikasi Anda.

> **Warning**
>
> Menyetel versi Node.js ke yang terbaru menyebabkan build gagal. Sebagai gantinya, Anda harus menentukan versi Node.js yang tepat, seperti`18`,`21.5`, atau`v0.1.2`.

Untuk mengonfigurasi pembaruan paket langsung

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang pembaruan paket langsungnya akan dikonfigurasi.
3. Di panel navigasi, pilih Hosting, Build settings.
4. Di halaman Pengaturan build, di bagian Pengaturan gambar build, pilih Edit.
5. Pada halaman Edit pengaturan gambar build, daftar pembaruan paket Live, pilih Tambah baru.
6. Untuk Package, pilih dependensi yang akan diganti.
7. Untuk Versi, pilih versi default terbaru atau masukkan versi dependensi tertentu. Jika Anda menggunakan versi terbaru, dependensi akan selalu ditingkatkan ke versi terbaru yang tersedia.
8. Pilih Simpan.

Mengonfigurasi instance build untuk aplikasi Amplify

Amplify Hosting menawarkan ukuran instans build yang dapat dikonfigurasi yang memungkinkan Anda menyediakan instance build aplikasi dengan CPU, memori, dan sumber daya ruang disk

yang diperlukan. Sebelum rilis fitur ini, Amplify menyediakan konfigurasi instance build ukuran tetap sebesar 8 GiB memori dan 4 v. CPUs

Amplify mendukung tiga jenis instance build:Standard,Large, dan. XLarge Jika Anda tidak menentukan tipe instans, Amplify akan menggunakan instans defaultStandard. Anda dapat mengonfigurasi jenis instance build untuk aplikasi menggunakan konsol Amplify, the AWS CLI, atau. SDKs

Biaya untuk setiap jenis instance build dihitung per menit build. Untuk rincian harga, lihat AWS Amplify Harga.

Tabel berikut menjelaskan spesifikasi komputasi untuk setiap tipe instans build:

Tipe instans build	v CPUs	Memori	Ruang disk
Standard	4 v CPUs	8 GiB	128 GB
Large	8 v CPUs	16 GiB	128 GB
XLarge	36 v CPUs	72 GiB	256 GB

Topik

Memahami tipe instance build

Pengaturan tipe instans build dikonfigurasi pada tingkat aplikasi dan meluas ke semua cabang aplikasi. Detail kunci berikut berlaku untuk jenis instance build:

- Jenis instans build yang Anda konfigurasikan untuk aplikasi secara otomatis berlaku untuk cabang yang dibuat secara otomatis dan pratinjau permintaan tarik.
- Kuota layanan pekerjaan bersamaan berlaku di semua jenis instans build di Anda. Akun AWS Misalnya, jika batas pekerjaan Concurrent Anda adalah lima, Anda dapat menjalankan hingga maksimal 5 build di semua jenis instans di Anda. Akun AWS

- Biaya untuk setiap jenis instance build dihitung per menit build. Proses alokasi instance build dapat memerlukan waktu overhead tambahan sebelum build Anda dimulai. Untuk instance yang lebih besar, terutama XLarge, build Anda mungkin mengalami latensi sebelum build dimulai, karena waktu overhead ini. Namun, Anda hanya ditagih untuk waktu pembuatan aktual, bukan waktu overhead.

Anda dapat mengonfigurasi tipe instans build saat membuat aplikasi baru atau memperbarui tipe instans pada aplikasi yang ada. Untuk petunjuk tentang mengonfigurasi setelan ini di konsol Amplify, lihat. Mengonfigurasi tipe instans build di konsol Amplify Anda juga dapat memperbarui pengaturan ini menggunakan pengaturan SDKs. Untuk informasi selengkapnya, lihat CreateApp, dan UpdateApp APIs di Referensi Amplify API.

Jika Anda memiliki aplikasi yang ada di akun Anda yang dibuat sebelum rilis fitur tipe instans build yang dapat disesuaikan, mereka menggunakan tipe Standard instans default. Saat Anda memperbarui jenis instance build untuk aplikasi yang sudah ada, build apa pun yang diantrian atau sedang berlangsung sebelum pembaruan Anda akan menggunakan jenis instance build yang telah dikonfigurasi sebelumnya. Misalnya, jika Anda memiliki aplikasi yang sudah ada dengan **`main`** cabang yang di-deploy ke Amplify dan memperbarui tipe instance build-nya dari Standard ke Large, semua build baru yang Anda mulai dari **`main`** cabang akan menggunakan tipe instance build Large. Namun, setiap build yang sedang berlangsung pada saat Anda memperbarui jenis instance build akan terus berjalan di instance Standard.

Mengonfigurasi tipe instans build di konsol Amplify

Gunakan prosedur berikut untuk mengonfigurasi tipe instans build saat Anda membuat aplikasi Amplify baru.

Untuk mengonfigurasi jenis instance build untuk aplikasi baru

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.
4. Di halaman Tambahkan cabang repositori, lakukan langkah berikut:
   a. Dalam daftar repositori yang baru diperbarui, pilih nama repositori yang akan dihubungkan.
   b. Di daftar Cabang, pilih nama cabang repositori yang akan dihubungkan.

c. Pilih Berikutnya.

5. Pada halaman Pengaturan aplikasi, buka bagian Pengaturan lanjutan.
6. Untuk jenis instans Build, pilih jenis instans yang Anda inginkan dari daftar.
7. Jika Anda menerapkan aplikasi berbasis runtime Node.js, konfigurasikan ukuran memori heap untuk secara efektif memanfaatkan jenis instance besar. Anda dapat melakukannya di halaman Pengaturan aplikasi dengan menyetel variabel lingkungan atau memperbarui setelan build. Untuk informasi selengkapnya, lihat Mengkonfigurasi memori heap aplikasi untuk memanfaatkan jenis instance besar .
   - Tetapkan variabel lingkungan
     a. Di bagian Pengaturan lanjutan, Variabel lingkungan, pilih Tambahkan baru.
     b. Untuk Key enter**`NODE_OPTIONS`**.
     c. Untuk Nilai, masukkan `--max-old-space-size=`*`memory_size_in_mb`*. Ganti *`memory_size_in_mb`* dengan ukuran memori heap yang Anda inginkan dalam megabyte.
   - Memperbarui pengaturan build
     a. Di bagian Build settings, pilih Edit file YML.
     b. Tambahkan perintah berikut ke `preBuild` fase. Ganti *`memory_size_in_mb`* dengan ukuran memori heap yang Anda inginkan dalam megabyte.

```
export NODE_OPTIONS='--max-old-space-size=memory_size_in_mb'
```

     c. Pilih Simpan.
8. Pilih, Berikutnya.
9. Di halaman Tinjauan, pilih Simpan dan deploy.

Gunakan prosedur berikut untuk mengonfigurasi tipe instans build aplikasi Amplify yang ada.

Untuk mengonfigurasi jenis instance build untuk aplikasi yang sudah ada

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang tipe instans build akan dikonfigurasi.
3. Di panel navigasi, pilih Hosting, lalu pilih Pengaturan build.
4. Pada halaman Pengaturan build, di bagian Pengaturan lanjutan, pilih Edit.

5. Pada halaman Edit pengaturan, untuk jenis instans Build, pilih jenis instans yang Anda inginkan dari daftar.
6. Pilih Simpan. Perubahan ini akan berlaku saat berikutnya Anda menyebarkan aplikasi.
7. (Opsional) Untuk segera menyebarkan aplikasi yang diperbarui, lakukan hal berikut:

    a. Di panel navigasi, pilih Ikhtisar.
    b. Pada halaman ikhtisar aplikasi Anda, pilih cabang yang akan digunakan kembali.
    c. Pada halaman Deployment, pilih deployment, seperti deployment terbaru. Kemudian, pilih Redeploy versi ini. Penyebaran baru akan dimulai.
    d. Saat penerapan selesai, setelan build aplikasi akan menunjukkan bahwa cabang menggunakan tipe instance build yang diperbarui.

Mengkonfigurasi memori heap aplikasi untuk memanfaatkan jenis instance besar

Jika Anda membangun aplikasi intensif memori, gunakan bagian ini untuk memahami cara mengkonfigurasi aplikasi Anda untuk memanfaatkan jenis instans besar. Bahasa pemrograman dan kerangka kerja sering mengandalkan mengalokasikan memori dinamis, juga dikenal sebagai memori heap, selama runtime untuk mengelola kebutuhan memori aplikasi. Memori heap diminta oleh lingkungan runtime dan dialokasikan oleh sistem operasi host. Secara default, lingkungan runtime memberlakukan batas ukuran heap maksimum yang tersedia untuk aplikasi. Ini berarti bahwa tidak ada memori tambahan yang akan tersedia untuk aplikasi di luar ukuran heap, meskipun sistem operasi host atau wadah memiliki jumlah memori yang lebih besar yang tersedia.

Sebagai contoh, lingkungan runtime JavaScript Node.js v8 memberlakukan batas ukuran heap default yang bergantung pada beberapa faktor, termasuk ukuran memori host. Akibatnya, Standard dan instance Large build memiliki ukuran heap Node.js default 2096 MB dan XLarge instance memiliki ukuran heap default 4144 MB. Oleh karena itu, membangun aplikasi dengan persyaratan memori 6000 MB menggunakan ukuran heap Node.js default pada semua jenis instance build Amplify akan menghasilkan build yang gagal karena kesalahan. out-of-memory

Untuk mengatasi batas memori ukuran heap default Node.js, Anda dapat melakukan salah satu dari berikut ini:

- Tetapkan variabel `NODE_OPTIONS` lingkungan dalam aplikasi Amplify Anda ke nilai. `--max-old-space-size=`*`memory_size_in_mb`* Untuk`memory_size_in_mb`, tentukan ukuran memori heap yang Anda inginkan dalam megabyte.

Untuk petunjuk, lihat Mengatur variabel lingkungan.

- Tambahkan perintah berikut ke `preBuild` fase dalam spesifikasi build aplikasi Amplify Anda.

```
export NODE_OPTIONS='--max-old-space-size=memory_size_in_mb'
```

Anda dapat memperbarui spesifikasi build di konsol Amplify atau di `amplify.yml` file aplikasi di repositori project. Untuk petunjuk, lihat Mengonfigurasi pengaturan build untuk aplikasi Amplify.

Contoh berikut Amplify build specification menetapkan ukuran memori heap Node.js menjadi 7000 MB untuk membangun aplikasi frontend React:

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        # Set the heap size to 7000 MB
        - export NODE_OPTIONS='--max-old-space-size=7000'
        # To check the heap size memory limit in MB
        - node -e "console.log('Total available heap size (MB):',
 v8.getHeapStatistics().heap_size_limit / 1024 / 1024)"
        - npm ci --cache .npm --prefer-offline
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: build
    files:
      - '**/*'
  cache:
    paths:
      - .npm/**/*
```

Untuk secara efektif memanfaatkan jenis instance besar, penting untuk memiliki ukuran memori heap yang cukup dikonfigurasi. Mengonfigurasi ukuran heap kecil untuk aplikasi intensif memori kemungkinan akan mengakibatkan kegagalan build. Log build aplikasi mungkin tidak secara langsung menunjukkan out-of-memory kesalahan karena runtime aplikasi dapat mogok secara tak terduga. Mengonfigurasi ukuran heap sebesar memori host dapat mengakibatkan sistem operasi host bertukar atau menghentikan proses lain, dan berpotensi mengganggu proses build Anda. Sebagai

referensi, Node.js merekomendasikan pengaturan ukuran heap maksimum 1536 MB pada mesin dengan sekitar 2000 MB memori untuk meninggalkan beberapa memori untuk kegunaan lain.

Ukuran heap optimal tergantung pada kebutuhan aplikasi dan penggunaan sumber daya Anda. Jika Anda menemukan out-of-memory kesalahan, mulailah dengan ukuran tumpukan sedang dan kemudian secara bertahap tingkatkan sesuai kebutuhan. Sebagai pedoman, sebaiknya mulai dengan 6000 MB untuk tipe Standard instans, 12000 MB untuk tipe Large instans, dan 60000 MB untuk tipe instans. XLarge

Membuat webhook masuk untuk memulai build

Atur webhook masuk di Konsol Amplify untuk memulai build tanpa menerapkan kode ke repositori Git. Anda dapat menggunakan webhook dengan alat CMS tanpa kepala (seperti Contentful atau GraphCMS) untuk memulai build setiap kali konten berubah, atau untuk menjalankan build harian menggunakan layanan, seperti Zapier.

Cara Membuat webhook masuk

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang akan dibuatkan webhook.
3. Di panel navigasi, pilih Hosting, lalu Buat pengaturan.
4. Di halaman Pengaturan build, gulir ke bawah ke bagian Webhook masuk, lalu pilih Buat webhook.
5. Di kotak dialog Buat webhook, lakukan hal berikut:
    a. Untuk Nama webhook, masukkan nama untuk webhook.
    b. Untuk Cabang yang akan dibuat, pilih cabang yang akan dibuat di permintaan webhook masuk.
    c. Pilih Buat webhook.
6. Di bagian Webhook masuk, lakukan salah satu langkah berikut:
    - Salin URL webhook dan tempelkan ke alat CMS tanpa kepala atau layanan lain untuk memulai build.
    - Jalankan perintah curl di jendela terminal untuk memulai build baru.

Menyiapkan notifikasi email untuk build

Anda dapat menyiapkan notifikasi email untuk AWS Amplify aplikasi untuk memberi tahu pemangku kepentingan atau anggota tim ketika build berhasil atau gagal. Amplify Hosting membuat topik Amazon Simple Notification Service (SNS) di akun Anda dan menggunakannya untuk mengonfigurasi notifikasi email. Notifikasi dapat dikonfigurasi agar dapat diterapkan ke semua cabang atau cabang tertentu di aplikasi Amplify.

Mengatur notifikasi email

Gunakan langkah-langkah berikut untuk mengatur notifikasi email untuk semua cabang atau cabang tertentu di aplikasi Amplify.

Mengatur notifikasi email untuk aplikasi Amplify

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang notifikasi emailnya akan diatur.
3. Di panel navigasi, pilih Hosting, Bangun pemberitahuan. Pada halaman Pemberitahuan build, pilih Kelola notifikasi.
4. Pada halaman Kelola pemberitahuan, pilih Tambah baru.
5. Lakukan salah satu tindakan berikut:
   - Untuk mengirim notifikasi untuk satu cabang, untuk Email, masukkan alamat email tujuan notifikasi dikirim. Untuk Cabang, pilih nama cabang tujuan notifikasi dikirim.
   - Untuk mengirim notifikasi untuk semua cabang yang terhubung, untuk Email, masukkan alamat email tujuan notifikasi dikirim. Untuk Cabang, pilih Semua Cabang.
6. Pilih Simpan.

Menghubungkan domain kustom

Anda dapat menghubungkan aplikasi yang Anda deploy dengan Amplify Hosting ke domain kustom. Saat Anda menggunakan Amplify untuk men-deploy aplikasi web Anda, Amplify menghostingnya untuk Anda di `amplifyapp.com` domain default dengan URL seperti. `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com` Saat Anda menghubungkan aplikasi ke domain kustom, pengguna melihat aplikasi di-hosting di URL kustom, misalnya`https://www.example.com`.

Anda dapat membeli domain kustom melalui registrar domain terakreditasi seperti Amazon Route 53 atau. GoDaddy Route 53 adalah layanan web Sistem Nama Domain (DNS) Amazon. Untuk informasi lebih lanjut seputar penggunaan Route 53, lihat Tentang Amazon Route 53. Untuk daftar pendaftar domain terakreditasi pihak ketiga, lihat Direktori Panitera Terakreditasi di situs web ICANN.

Saat menyiapkan domain kustom, Anda dapat menggunakan sertifikat terkelola default yang disediakan Amplify untuk Anda atau Anda dapat menggunakan sertifikat kustom Anda sendiri. Anda dapat mengubah sertifikat yang digunakan domain kapan saja. Untuk informasi mendetail seputar pengelolaan sertifikat, lihatMenggunakan sertifikat SSL/TLS.

Sebelum Anda melanjutkan dengan menyiapkan domain kustom, verifikasi bahwa Anda telah memenuhi prasyarat berikut.

- Anda memiliki nama domain terdaftar.
- Anda memiliki sertifikat yang dikeluarkan oleh atau diimpor ke AWS Certificate Manager.
- Anda telah menerapkan aplikasi Anda ke Amplify Hosting.

  Untuk informasi selengkapnya seputar penyelesaian langkah ini, lihatMemulai dengan menerapkan aplikasi ke Amplify Hosting.
- Anda memiliki pengetahuan dasar seputar domain dan terminologi DNS.

  Untuk informasi lebih lanjut seputar domain dan DNS, lihat Memahami terminologi dan konsep DNS.

Topik

- Menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga
- Memperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy
- Memperbarui sertifikat SSL/TLS untuk domain
- Mengelola subdomain
- Menyiapkan subdomain
- Menyiapkan subdomain otomatis untuk domain kustom Amazon Route 53
- Memecahkan masalah domain kustom

Memahami terminologi dan konsep DNS

Jika Anda belum mengenal berbagai istilah dan konsep yang terkait dengan Sistem Nama Domain (DNS), topik berikut dapat membantu Anda memahami langkah-langkah menambahkan domain kustom.

Terminologi DNS

Berikut daftar istilah yang sering ditemui seputar DNS. Istilah tersebut dapat membantu Anda memahami langkah-langkah menambahkan domain kustom.

CNAME

Canonical Record Name (CNAME) adalah jenis catatan DNS yang menutupi domain untuk serangkaian halaman web dan membuatnya tampak seolah-olah berada di tempat lain. CNAME mengarahkan subdomain ke nama domain yang sepenuhnya memenuhi syarat (FQDN). Misalnya, Anda dapat membuat catatan CNAME baru untuk memetakan subdomain www.example.com, di mana www adalah subdomain, ke domain FQDN branch-name.d1m7bkiki6tdw1.cloudfront.net yang ditetapkan ke aplikasi Anda di konsol Amplify.

ANAME

Catatan ANAME mirip seperti catatan CNAME, tetapi di tingkat root. ANAME mengarahkan root domain Anda ke FQDN. FQDN tersebut mengarah ke alamat IP.

Server nama

Server nama adalah server di internet yang khusus menangani kueri mengenai lokasi berbagai layanan suatu nama domain. Jika Anda mengatur domain di Amazon Route 53, daftar server nama sudah ditetapkan ke domain Anda.

Catatan NS

Catatan NS mengarah ke server nama yang mencari detail domain Anda.

Verifikasi DNS

Sistem Nama Domain (DNS) mirip seperti buku telepon yang menerjemahkan nama domain yang dapat dibaca manusia ke alamat IP mudah ditemukan komputer. Saat Anda mengetik **`https://google.com`** di peramban, operasi pencarian dijalankan di penyedia DNS untuk menemukan Alamat IP server yang meng-host situs web.

Penyedia DNS berisi catatan domain dan Alamat IP yang sesuai. Catatan DNS yang paling umum digunakan adalah catatan CNAME, ANAME, dan NS.

Amplify menggunakan catatan CNAME untuk memverifikasi bahwa Anda adalah pemilik domain kustom. Jika Anda meng-host domain dengan Route 53, verifikasi dilakukan secara otomatis atas nama Anda. Namun, jika meng-host domain dengan penyedia pihak ketiga GoDaddy, seperti, Anda harus memperbarui pengaturan DNS domain secara manual dan menambahkan catatan CNAME baru yang disediakan Amplify.

Proses aktivasi domain kustom

Saat menghubungkan aplikasi Amplify ke domain kustom di konsol Amplify, ada beberapa langkah yang harus diselesaikan Amplify sebelum Anda dapat menampilkan aplikasi menggunakan domain kustom Anda. Daftar berikut menjelaskan setiap langkah dalam proses penyiapan dan aktivasi domain.

Pembuatan SSL/TLS

Jika Anda menggunakan sertifikat yang dikelola, AWS Amplify menerbitkan sertifikat SSL/TLS untuk menyiapkan domain kustom yang aman.

Konfigurasi dan verifikasi SSL/TLS

Sebelum menerbitkan sertifikat yang dikelola, Amplify memverifikasi bahwa Anda adalah pemilik domain. Untuk domain yang dikelola Amazon Route 53, Amplify secara otomatis memperbarui catatan verifikasi DNS. Untuk domain yang dikelola di luar Route 53, Anda harus menambahkan secara manual catatan verifikasi DNS yang disediakan di konsol Amplify ke domain Anda dengan penyedia DNS pihak ketiga.

Jika Anda menggunakan sertifikat khusus, Anda bertanggung jawab untuk memvalidasi kepemilikan domain.

Aktivasi domain

Domain berhasil diverifikasi. Untuk domain yang dikelola di luar Route 53, Anda harus menambahkan secara manual catatan CNAME yang disediakan di konsol Amplify ke domain Anda dengan penyedia DNS pihak ketiga.

Menggunakan sertifikat SSL/TLS

Sebuah SSL/TLS certificate is a digital document that allows web browsers to identify and establish encrypted network connections to web sites using the secure SSL/TLS protokol. Saat menyiapkan domain kustom, Anda dapat menggunakan sertifikat terkelola default yang disediakan Amplify untuk Anda atau Anda dapat menggunakan sertifikat kustom Anda sendiri.

Dengan sertifikat yang dikelola, Amplify menerbitkan sertifikat SSL/TLS untuk semua domain yang terhubung ke aplikasi Anda sehingga semua lalu lintas diamankan melalui HTTPS/2. Sertifikat default yang dihasilkan AWS Certificate Manager (ACM) berlaku selama 13 bulan dan diperpanjang secara otomatis selama aplikasi Anda di-hosting dengan Amplify.

> **Warning**
>
> Amplify tidak dapat memperpanjang sertifikat jika catatan verifikasi CNAME telah diubah atau dihapus dalam pengaturan DNS dengan penyedia domain Anda. Anda harus menghapus dan menambahkan domain lagi di konsol Amplify.

Untuk menggunakan sertifikat kustom, Anda harus terlebih dahulu mendapatkan sertifikat dari otoritas sertifikat pihak ketiga pilihan Anda. Amplify Hosting mendukung dua jenis sertifikat: RSA (Rivest-Shamir-Adleman) dan ECDSA (Elliptic Curve Digital Signature Algorithm). Setiap jenis sertifikat harus sesuai dengan persyaratan berikut.

Sertifikat RSA

- Amplify Hosting mendukung kunci RSA 1024-bit, 2048-bit, 3072-bit, dan 4096-bit.
- AWS Certificate Manager (ACM) mengeluarkan sertifikat RSA dengan kunci hingga 2048-bit.
- Untuk menggunakan sertifikat RSA 3072-bit atau 4096-bit, dapatkan sertifikat secara eksternal dan impor ke ACM. Ini kemudian akan tersedia untuk digunakan dengan Amplify Hosting.

Sertifikat ECDSA

- Amplify Hosting mendukung kunci 256-bit.
- Gunakan kurva elips prime256v1 untuk mendapatkan sertifikat ECDSA untuk Amplify Hosting.

Setelah Anda mendapatkan sertifikat, impor ke AWS Certificate Manager. ACM adalah layanan yang memungkinkan Anda dengan mudah menyediakan, mengelola, dan menyebarkan sertifikat SSL/TLS publik dan pribadi untuk digunakan dengan Layanan AWS dan sumber daya internal Anda yang terhubung. Pastikan Anda meminta atau mengimpor sertifikat di Wilayah US East (N. Virginia) (us-east-1).

Pastikan sertifikat kustom Anda mencakup semua subdomain yang ingin Anda tambahkan. Anda dapat menggunakan wildcard di awal nama domain Anda untuk mencakup beberapa subdomain. Misalnya, jika domain Anda`example.com`, Anda dapat menyertakan domain `*.example.com` wildcard. Ini akan mencakup subdomain seperti `product.example.com` dan. `api.example.com`

Setelah sertifikat kustom Anda tersedia di ACM, Anda akan dapat memilihnya selama proses pengaturan domain. Untuk petunjuk tentang mengimpor sertifikat ke dalam AWS Certificate Manager, lihat Mengimpor sertifikat ke AWS Certificate Manager dalam AWS Certificate Manager Panduan Pengguna.

Jika Anda memperbarui atau mengimpor ulang sertifikat kustom Anda di ACM, Amplify akan menyegarkan data sertifikat yang terkait dengan domain kustom Anda. Dalam kasus sertifikat yang diimpor, ACM tidak mengelola perpanjangan secara otomatis. Anda bertanggung jawab untuk memperbarui sertifikat kustom Anda dan mengimpornya lagi.

Anda dapat mengubah sertifikat yang digunakan untuk domain setiap saat. Misalnya, Anda dapat beralih dari sertifikat terkelola default ke sertifikat kustom atau mengubah dari sertifikat kustom ke sertifikat terkelola. Selain itu, Anda dapat mengubah sertifikat kustom yang digunakan ke sertifikat kustom yang berbeda. Untuk petunjuk tentang memperbarui sertifikat, lihat Memperbarui sertifikat SSL/TLS untuk domain.

Menambahkan domain kustom yang dikelola Amazon Route 53

Amazon Route 53 adalah layanan DNS yang dapat diskalakan dan sangat tersedia. Untuk informasi selengkapnya, lihat Amazon Route 53 di Panduan Developer Amazon Route 53. Jika Anda sudah memiliki domain Route 53, gunakan petunjuk berikut untuk menghubungkan domain kustom Anda ke aplikasi Amplify Anda.

Cara menambahkan domain kustom yang dikelola Route 53

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang akan dihubungkan ke domain kustom.
3. Di panel navigasi, pilih Hosting, Domain khusus.
4. Pada halaman Custom domain, pilih Add domain.
5. Masukkan nama domain root Anda. Misalnya, jika nama domain Anda https://example.com, masukkan**`example.com`**.

   Saat Anda mulai mengetik, domain root yang sudah Anda kelola di Route 53 muncul dalam daftar. Anda dapat memilih domain yang ingin Anda gunakan dari daftar. Jika belum memiliki domain dan domain tersedia, Anda dapat membeli domain tersebut di Amazon Route 53.
6. Setelah Anda memasukkan nama domain Anda, pilih Konfigurasi domain.
7. Secara default, Amplify secara otomatis membuat dua entri subdomain untuk domain Anda. Misalnya, jika nama domain Anda adalah example.com, Anda akan melihat subdomain https://www.example.comdan https://example.comdengan pengalihan diatur dari domain root ke subdomain www.

   (Opsional) Anda dapat mengubah konfigurasi default jika ingin menambahkan subdomain saja. Untuk mengubah konfigurasi default, pilih Penulisan ulang dan pengalihan dari panel navigasi, lalu konfigurasi domain Anda.
8. Pilih sertifikat SSL/TLS untuk digunakan. Anda dapat menggunakan sertifikat terkelola default yang disediakan Amplify untuk Anda, atau sertifikat pihak ketiga khusus yang telah Anda impor. AWS Certificate Manager
   - Gunakan sertifikat terkelola Amplify default.
     - Pilih Amplify sertifikat terkelola.
   - Gunakan sertifikat pihak ketiga khusus.
     a. Pilih sertifikat SSL khusus.
     b. Pilih sertifikat yang akan digunakan dari daftar.
9. Pilih Tambahkan domain.

> **Note**
>
> DNS memerlukan waktu hingga 24 jam untuk menyebar dan menerbitkan sertifikat. Untuk bantuan seputar mengatasi kesalahan yang terjadi, lihat Memecahkan masalah domain kustom.

Menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga

Jika tidak menggunakan Amazon Route 53 untuk mengelola domain, Anda dapat menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga ke aplikasi yang di-deploy dengan Amplify.

Jika Anda menggunakan GoDaddy, lihat the section called “Memperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy” instruksi khusus untuk penyedia ini.

Cara menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi tempat domain kustom akan ditambahkan.
3. Di panel navigasi, pilih Hosting, Domain khusus.
4. Pada halaman Custom domain, pilih Add domain.
5. Masukkan nama domain root Anda. Misalnya, jika nama domain Anda https://example.com, masukkan**`example.com`**.
6. Amplify mendeteksi bahwa Anda tidak menggunakan domain Route 53 dan memberi Anda opsi untuk membuat zona yang dihosting di Route 53.
   - Untuk membuat zona yang dihosting di Route 53
     a. Pilih Buat zona yang dihosting di Rute 53.
     b. Pilih Konfigurasikan domain.
     c. Server nama zona yang dihosting ditampilkan di konsol. Lanjutkan ke situs penyedia DNS dan tambahkan server nama ke pengaturan DNS Anda.
     d. Pilih Saya telah menambahkan server nama di atas ke registri domain saya.
     e. Lanjutkan ke langkah tujuh.
   - Untuk melanjutkan dengan konfigurasi manual

a. Pilih konfigurasi Manual

b. Pilih Konfigurasikan domain.

c. Lanjutkan ke langkah tujuh.

7. Secara default, Amplify secara otomatis membuat dua entri subdomain untuk domain Anda. Misalnya, jika nama domain Anda adalah example.com, Anda akan melihat subdomain https://www.example.comdan https://example.comdengan pengalihan diatur dari domain root ke subdomain www.

   (Opsional) Anda dapat mengubah konfigurasi default jika ingin menambahkan subdomain saja. Untuk mengubah konfigurasi default, pilih Penulisan ulang dan pengalihan dari panel navigasi dan konfigurasi domain Anda.

8. Pilih sertifikat SSL/TLS untuk digunakan. Anda dapat menggunakan sertifikat terkelola default yang disediakan Amplify untuk Anda, atau sertifikat pihak ketiga khusus yang telah Anda impor. AWS Certificate Manager

   - Gunakan sertifikat terkelola Amplify default.

     - Pilih Amplify sertifikat terkelola.

   - Gunakan sertifikat pihak ketiga khusus.

     a. Pilih sertifikat SSL khusus.

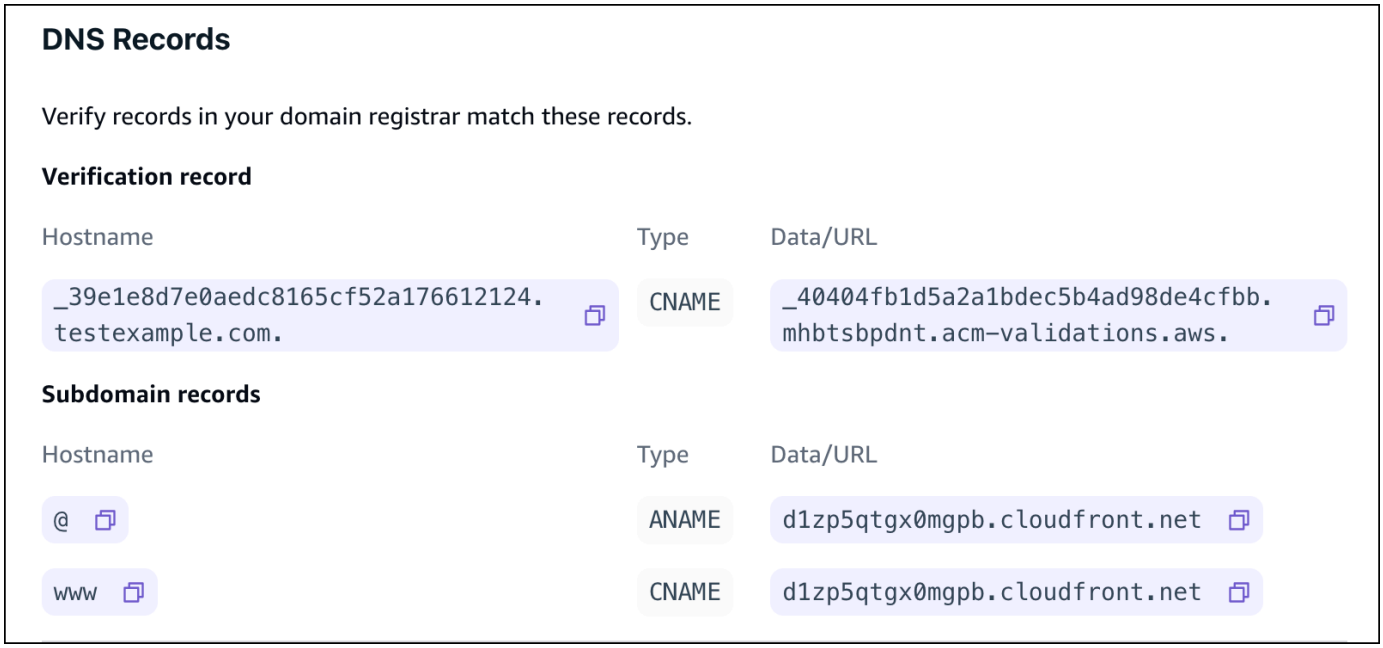
     b. Pilih sertifikat yang akan digunakan dari daftar.

9. Pilih Tambahkan domain.

10. Jika Anda memilih Buat zona yang dihosting di Route 53 di langkah enam, lanjutkan ke langkah 15.

    Jika memilih konfigurasi Manual, di langkah enam, Anda harus memperbarui catatan DNS Anda dengan penyedia domain pihak ketiga.

    Di menu Tindakan, pilih Lihat catatan DNS. Tangkapan layar berikut menunjukkan catatan DNS yang ditampilkan di konsol.

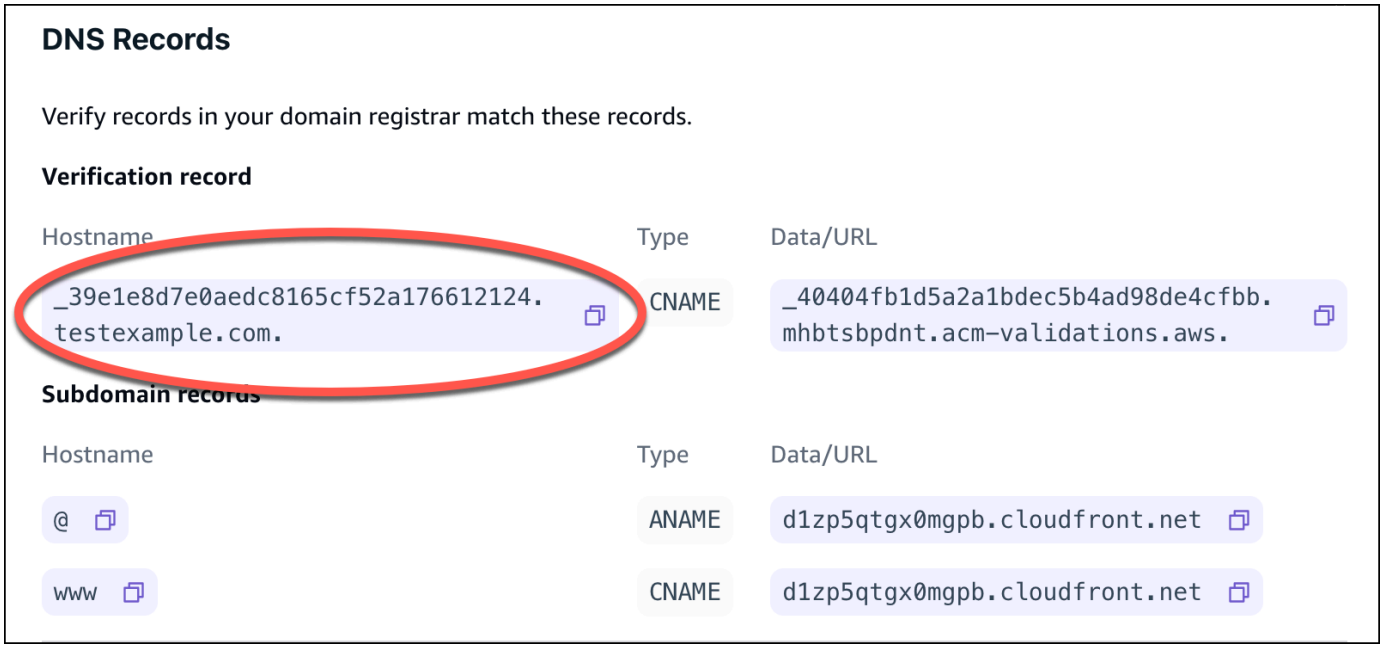


11. Lakukan salah satu tindakan berikut:

    - Jika Anda menggunakan GoDaddy, pergi keMemperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy.
    - Jika Anda menggunakan penyedia DNS pihak ketiga yang berbeda, lanjutkan ke langkah berikutnya dalam panduan ini.
12. Buka situs web penyedia DNS, masuk ke akun Anda, lalu cari pengaturan manajemen DNS untuk domain Anda. Anda akan mengkonfigurasi dua catatan CNAME.
13. Konfigurasi catatan CNAME pertama untuk mengarahkan subdomain Anda ke server AWS validasi.

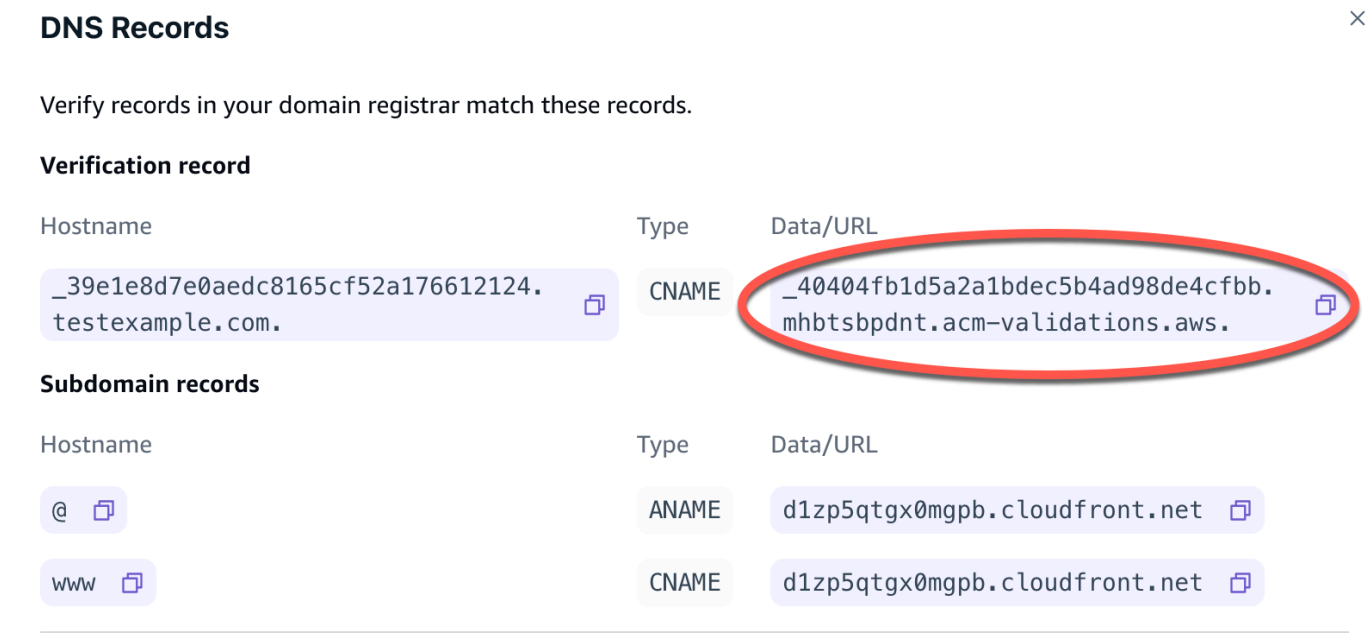
    Jika konsol Amplify menampilkan catatan DNS untuk verifikasi kepemilikan subdomain Anda, seperti _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, hanya masukkan untuk nama subdomain catatan CNAME. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

    Tangkapan layar berikut menunjukkan lokasi catatan verifikasi yang akan digunakan.



Jika konsol Amplify menampilkan catatan server validasi ACM, seperti _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, masukkan untuk nilai catatan CNAME. **`_cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`**

Tangkapan layar berikut menunjukkan lokasi catatan verifikasi ACM yang akan digunakan.



Amplify menggunakan informasi ini untuk memverifikasi kepemilikan domain dan menghasilkan sertifikat SSL/TLS untuk domain Anda. Setelah Amplify memvalidasi kepemilikan domain, semua lalu lintas akan dilayani menggunakan HTTPS/2.

> **Note**
>
> Sertifikat Amplify default yang dihasilkan AWS Certificate Manager (ACM) berlaku selama 13 bulan dan diperpanjang secara otomatis selama aplikasi Anda di-hosting dengan Amplify. Amplify tidak dapat memperbarui sertifikat jika catatan verifikasi CNAME telah diubah atau dihapus. Anda harus menghapus dan menambahkan domain lagi di konsol Amplify.

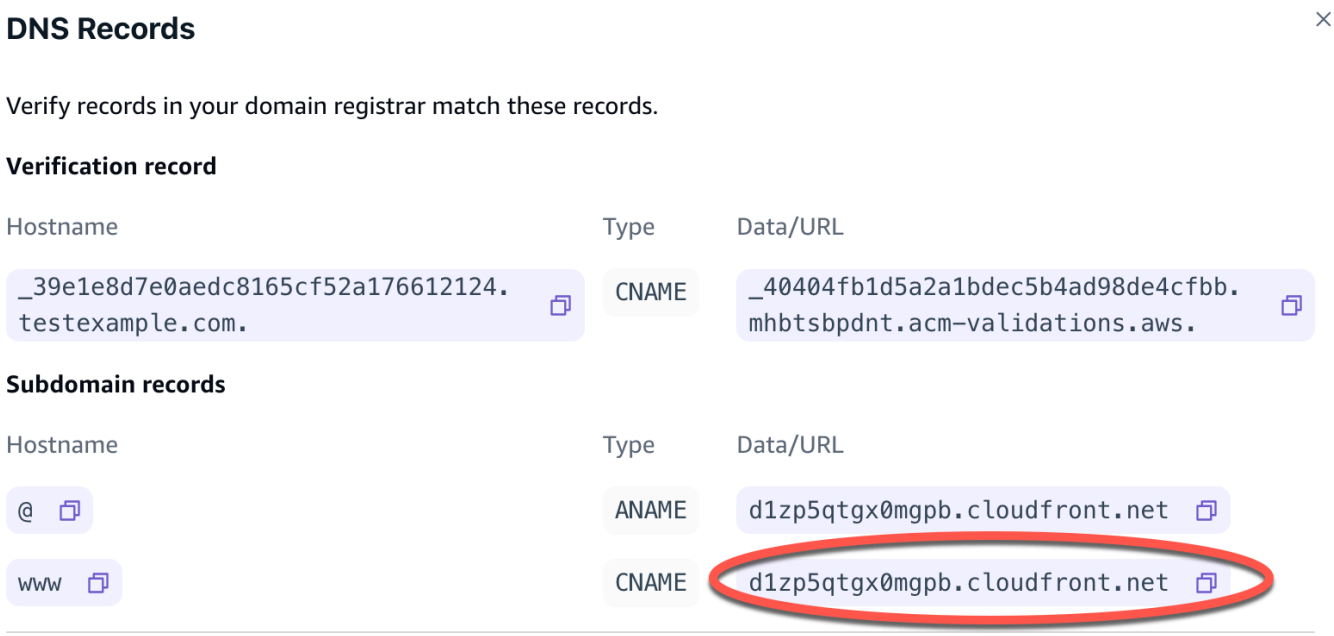
> **Important**
>
> Anda harus melakukan langkah ini segera setelah menambahkan domain kustom Anda di konsol Amplify. AWS Certificate Manager (ACM) segera mulai mencoba memverifikasi kepemilikan. Seiring waktu, frekuensi pemeriksaan menjadi berkurang. Jika Anda menambahkan atau memperbarui catatan CNAME beberapa jam setelah membuat aplikasi, aplikasi Anda akan terus menampilkan status menunggu verifikasi.

14. Konfigurasi catatan CNAME kedua untuk mengarahkan subdomain Anda ke domain Amplify. Misalnya, jika subdomain Anda adalah www.example.com, masukkan www untuk nama subdomain.

    Jika konsol Amplify menampilkan domain untuk aplikasi Anda sebagai d111111abcdef8.cloudfront.net, masukkan domain Amplify.
    `d111111abcdef8.cloudfront.net`

    Jika memiliki lalu lintas produksi, Anda sebaiknya memperbarui catatan CNAME setelah status domain Anda menampilkan tulisan TERSEDIA di konsol Amplify.

    Tangkapan layar berikut menunjukkan lokasi catatan nama domain yang akan digunakan.



15. Konfigurasi catatan ANAME/ALIAS agar mengarah ke domain root aplikasi Anda (misalnya). https://example.com Catatan ANAME mengarahkan root domain Anda ke hostname. Jika memiliki lalu lintas produksi, Anda sebaiknya memperbarui catatan ANAME setelah status domain Anda menampilkan tulisan TERSEDIA di konsol. Untuk penyedia DNS yang tidak memiliki dukungan ANAME/ALIAS, kami sangat menyarankan migrasi DNS Anda ke Route 53. Untuk informasi selengkapnya, lihat Mengonfigurasi Amazon Route 53 sebagai layanan DNS Anda.

> **Note**
>
> Verifikasi kepemilikan domain dan propagasi DNS untuk domain pihak ketiga dapat membutuhkan waktu hingga 48 jam. Untuk bantuan seputar mengatasi kesalahan yang terjadi, lihat Memecahkan masalah domain kustom.

Memperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy

Jika GoDaddy penyedia DNS Anda, gunakan petunjuk berikut untuk memperbarui data DNS Anda di GoDaddy UI untuk menyelesaikan penyambungan aplikasi Amplify ke domain Anda. GoDaddy

Cara menambahkan domain kustom yang dikelola GoDaddy

1. Sebelum Anda dapat memperbarui catatan DNS Anda GoDaddy, selesaikan langkah satu hingga sembilan prosedurthe section called “Menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga”.
2. Masuk ke GoDaddy akun Anda.
3. Di daftar domain, cari domain yang akan ditambahkan, lalu pilih Kelola DNS.
4. Pada halaman DNS, GoDaddy menampilkan daftar catatan untuk domain Anda di bagian Catatan DNS. Anda harus menambahkan dua catatan CNAME baru.
5. Buat catatan CNAME pertama untuk mengarahkan subdomain Anda ke domain Amplify.

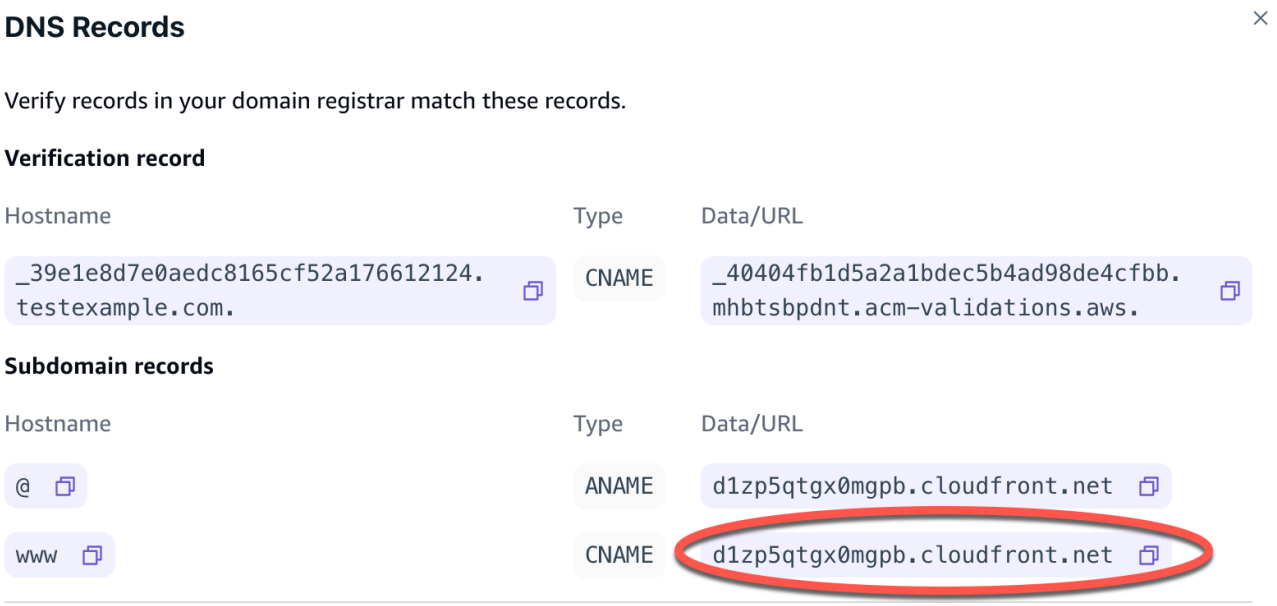
   a. Di bagian DNS Records, pilih Add New Record.

   b. Untuk Type, pilih CNAME.

   c. Untuk Nama, masukkan subdomain saja. Misalnya, jika subdomain Anda adalah www.example.com, masukkan www untuk Nama.

   d. Untuk Nilai, lihat catatan DNS Anda di konsol Amplify, lalu masukkan nilai. Jika konsol Amplify menampilkan domain untuk aplikasi Anda sebagai d111111abcdef8.cloudfront.net, masukkan untuk Nilai. **`d111111abcdef8.cloudfront.net`**

   Tangkapan layar berikut menunjukkan lokasi catatan nama domain yang akan digunakan.



   e. Pilih Simpan.

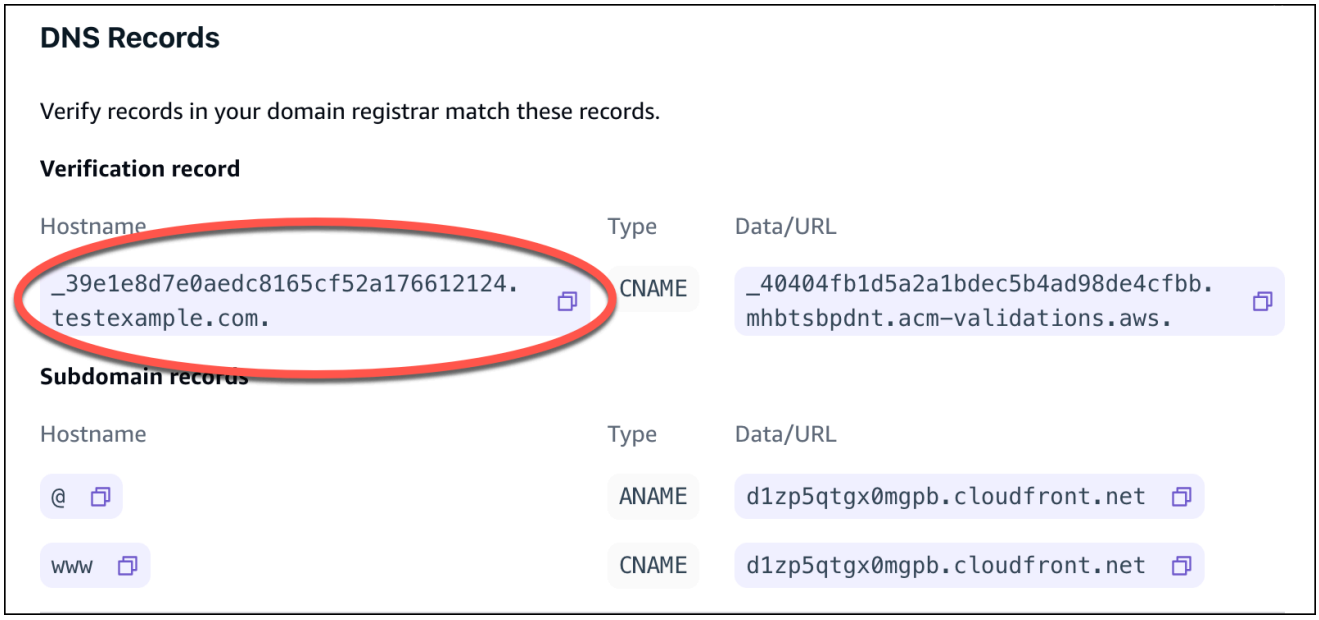
6. Buat catatan CNAME kedua mengarah ke server validasi AWS Certificate Manager (ACM). Satu ACM tervalidasi menghasilkan satu sertifikat SSL/TLS untuk domain Anda.

   a. Untuk Type, pilih CNAME.

   b. Untuk Nama, masukkan subdomain.

      Misalnya, jika catatan DNS di konsol Amplify untuk verifikasi kepemilikan subdomain Anda adalah _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, hanya masukkan untuk Nama. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

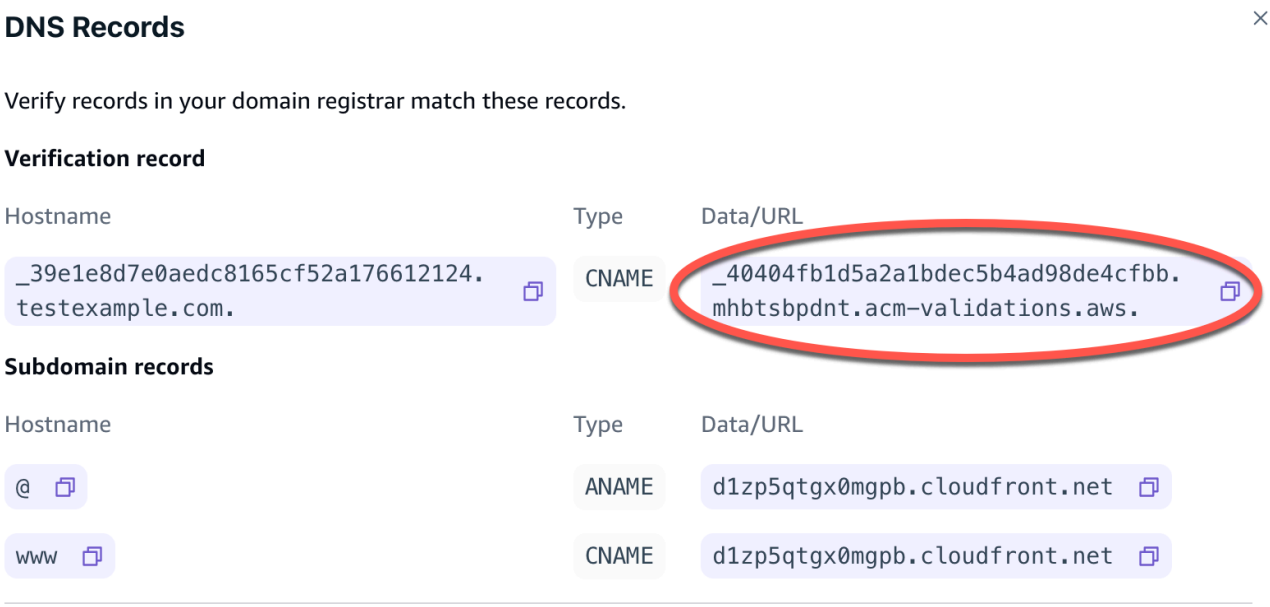
      Tangkapan layar berikut menunjukkan lokasi catatan verifikasi yang akan digunakan.



   c. Untuk Value, masukkan sertifikat validasi ACM.

      Misalnya, jika server validasi adalah _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, masukkan _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws untuk Nilai.

      Tangkapan layar berikut menunjukkan lokasi catatan verifikasi ACM yang akan digunakan.



d. Pilih Simpan.

> **Note**
>
> Sertifikat Amplify default yang dihasilkan AWS Certificate Manager (ACM) berlaku selama 13 bulan dan diperpanjang secara otomatis selama aplikasi Anda di-hosting dengan Amplify. Amplify tidak dapat memperbarui sertifikat jika catatan verifikasi CNAME telah diubah atau dihapus. Anda harus menghapus dan menambahkan domain lagi di konsol Amplify.

7. Langkah ini tidak diperlukan untuk subdomain. GoDaddy tidak mendukung ANAME/ALIAS records. For DNS providers that do not have ANAME/ALIAS dukungan, kami sangat menyarankan migrasi DNS Anda ke Amazon Route 53. Untuk informasi selengkapnya, lihat Mengonfigurasi Amazon Route 53 sebagai layanan DNS Anda.

   Jika Anda ingin tetap GoDaddy sebagai penyedia dan memperbarui domain root, tambahkan Penerusan dan atur penerusan domain:

   a. Pada halaman DNS, cari menu di bagian atas halaman dan pilih Penerusan.

   b. Di bagian Domain, pilih Tambahkan Penerusan.

   c. Pilih http://, lalu masukkan nama subdomain Anda untuk diteruskan ke (misalnya, www.example.com) untuk URL Tujuan.

   d. Untuk Tipe Penerusan, pilih Sementara (302).

e. Pilih, Simpan.

Memperbarui sertifikat SSL/TLS untuk domain

Anda dapat mengubah sertifikat SSL/TLS yang digunakan untuk domain kapan saja. Misalnya, Anda dapat mengubah dari menggunakan sertifikat terkelola menjadi menggunakan sertifikat khusus. Ini sangat membantu jika Anda ingin mengelola sertifikat dan pemberitahuan kedaluwarsanya. Anda juga dapat mengubah sertifikat kustom yang digunakan untuk domain. Membuat perubahan pada sertifikat SSL tidak akan menimbulkan downtime untuk domain aktif Anda. Untuk informasi selengkapnya tentang sertifikat, lihat Menggunakan sertifikat SSL/TLS.

Gunakan prosedur berikut untuk memperbarui jenis sertifikat atau sertifikat khusus yang digunakan untuk domain.

Untuk memperbarui sertifikat domain

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang akan Anda perbarui.
3. Di panel navigasi, pilih Hosting, Domain khusus.
4. Pada halaman Custom domain, pilih konfigurasi Domain.
5. Pada halaman detail untuk domain Anda, cari bagian Sertifikat SSL Kustom. Prosedur untuk memperbarui sertifikat Anda bervariasi tergantung pada jenis perubahan yang ingin Anda lakukan.
   - Untuk mengubah dari sertifikat kustom ke sertifikat terkelola Amplify default
     - Pilih Amplify sertifikat terkelola.
   - Untuk mengubah dari sertifikat terkelola ke sertifikat kustom
     a. Pilih sertifikat SSL khusus.
     b. Pilih sertifikat yang akan digunakan dari daftar.
   - Untuk mengubah sertifikat kustom ke sertifikat kustom yang berbeda
     - Untuk sertifikat SSL khusus, pilih sertifikat baru yang akan digunakan dari daftar.
6. Pilih Simpan. Detail status untuk domain akan menunjukkan bahwa Amplify telah memulai proses pembuatan SSL untuk sertifikat terkelola atau proses konfigurasi untuk sertifikat kustom.

Mengelola subdomain

Subdomain adalah bagian dari URL dan muncul sebelum nama domain. Misalnya, www adalah subdomain dari www.amazon.com dan aws adalah subdomain dari aws.amazon.com. Jika sudah memiliki situs web produksi, Anda mungkin akan menghubungkan subdomain saja. Subdomain juga bisa terdiri dari beberapa level, misalnya beta.alpha.example.com memiliki subdomain multilevel beta.alpha.

Cara menambahkan subdomain saja

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi tempat subdomain akan ditambahkan.
3. Di panel navigasi, pilih Hosting, lalu pilih Domain khusus.
4. Pada halaman Custom domain, pilih Add domain.
5. Masukkan nama domain root, lalu pilih Konfigurasi domain. Misalnya, jika nama domain Anda https://example.com, masukkan example.com.
6. Pilih Jangan sertakan root, lalu modifikasi nama subdomain. Misalnya jika domain adalah example.com, Anda dapat memodifikasinya agar hanya menambahkan subdomain alpha.
7. Pilih Tambahkan domain.

Cara menambahkan subdomain multilevel

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi tempat subdomain multilevel akan ditambahkan.
3. Di panel navigasi, pilih Hosting, lalu pilih Domain khusus.
4. Pada halaman Custom domain, pilih Add domain.
5. Masukkan nama domain dengan subdomain, pilih Jangan sertakan root, lalu modifikasi subdomain untuk menambahkan level baru.

   Misalnya, jika Anda memiliki domain bernama alpha.example.com dan ingin membuat subdomain multilevel beta.alpha.example.com, Anda perlu memasukkan beta sebagai nilai subdomain.
6. Pilih Tambahkan domain.

Cara menambahkan atau mengedit subdomain

Setelah menambahkan domain kustom ke aplikasi, Anda dapat mengedit subdomain yang ada atau menambahkan subdomain baru.

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi tempat subdomain akan dikelola.
3. Di panel navigasi, pilih Hosting, lalu pilih Domain khusus.
4. Pada halaman Custom domain, pilih konfigurasi Domain.
5. Di bagian Subdomain, Anda dapat mengedit subdomain yang ada sesuai kebutuhan.
6. (Opsional) Untuk menambahkan subdomain baru, pilih Tambahkan baru.
7. Pilih Simpan.

Menyiapkan subdomain

Amplify Hosting sekarang mendukung subdomain wildcard. Subdomain wildcard adalah subdomain catch-all yang memungkinkan Anda mengarahkan subdomain yang ada dan yang tidak ada ke cabang spesifik aplikasi Anda. Jika Anda menggunakan wildcard untuk mengaitkan semua subdomain dalam aplikasi ke cabang tertentu, Anda dapat menayangkan konten yang sama kepada pengguna aplikasi di subdomain apa pun dan menghindari mengonfigurasi setiap subdomain satu per satu.

Untuk membuat subdomain wildcard, tentukan tanda bintang (*) sebagai nama subdomain. Misalnya, jika Anda menentukan subdomain wildcard `*.example.com` untuk cabang tertentu aplikasi Anda, URL apa pun yang diakhiri dengan example.com akan dirutekan ke cabang. Dalam hal ini, permintaan untuk `dev.example.com` dan `prod.example.com` akan diarahkan ke `*.example.com` subdomain.

Perhatikan bahwa Amplify mendukung subdomain wildcard hanya untuk domain kustom. Anda tidak dapat menggunakan fitur ini dengan `amplifyapp.com` domain default.

Persyaratan berikut berlaku untuk subdomain

- Nama subdomain harus ditentukan dengan tanda bintang (*) saja.
- Anda tidak dapat menggunakan wildcard untuk mengganti bagian dari nama subdomain, seperti ini: *domain.example.com.

- Anda tidak dapat mengganti subdomain di tengah nama domain, seperti ini: subdomain.*.example.com.
- Secara default, semua sertifikat yang disediakan Amplify mencakup semua subdomain untuk domain kustom.

Cara menambahkan atau menghapus subdomain

Setelah menambahkan domain kustom ke aplikasi, Anda dapat menambahkan subdomain wildcard untuk cabang aplikasi.

1. Masuk ke AWS Management Console dan buka konsol Amplify Hosting.
2. Pilih aplikasi tempat subdomain wildcard akan dikelola.
3. Di panel navigasi, pilih Hosting, lalu pilih Domain khusus.
4. Pada halaman Custom domain, pilih konfigurasi Domain.
5. Di bagian Subdomain, Anda dapat menambahkan atau menghapus subdomain wildcard.
   - Cara menambahkan subdomain
     a. Pilih Tambahkan baru.
     b. Untuk subdomain, masukkan file. *****
     c. Untuk cabang aplikasi Anda, pilih nama cabang dari daftar.
     d. Pilih Simpan.
   - Untuk menghapus subdomain wildcard
     a. Pilih Hapus di sebelah nama subdomain. Lalu lintas ke subdomain yang tidak dikonfigurasi secara eksplisit berhenti, dan Amplify Hosting mengembalikan kode status 404 ke permintaan tersebut.
     b. Pilih Simpan.

Menyiapkan subdomain otomatis untuk domain kustom Amazon Route 53

Setelah aplikasi terhubung ke domain kustom di Route 53, Amplify memungkinkan Anda untuk secara otomatis membuat subdomain untuk cabang yang baru terhubung. Misalnya, jika Anda

menghubungkan cabang dev, Amplify dapat secara otomatis membuat dev.exampledomain.com. Jika Anda menghapus cabang, subdomain terkait akan dihapus secara otomatis.

Cara mengatur pembuatan subdomain otomatis untuk cabang yang baru terhubung

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang terhubung ke domain kustom yang dikelola di Route 53.
3. Di panel navigasi, pilih Hosting, lalu pilih Domain khusus.
4. Pada halaman Custom domain, pilih konfigurasi Domain.
5. Di bagian Pembuatan subdomain otomatis, nyalakan fitur.

> **Note**
>
> Fitur ini hanya tersedia untuk domain root, misalnya exampledomain.com. Konsol Amplify tidak menampilkan kotak centang ini jika domain Anda sudah menjadi subdomain, misalnya dev.exampledomain.com.

Pratinjau web dengan subdomain

Setelah Anda mengaktifkan pembuatan subdomain otomatis menggunakan instruksi sebelumnya, pratinjau web permintaan tarik aplikasi Anda juga akan dapat diakses dengan subdomain yang dibuat secara otomatis. Ketika permintaan tarik ditutup, cabang dan subdomain terkait secara otomatis dihapus. Untuk informasi lebih lanjut tentang pengaturan pratinjau web untuk permintaan tarik, lihat Pratinjau web untuk permintaan tarik.

Memecahkan masalah domain kustom

Jika Anda mengalami masalah saat menambahkan domain kustom ke aplikasi di AWS Amplify konsol, baca Memecahkan masalah domain kustom di bagian pemecahan masalah Amplify. Jika Anda tidak melihat solusi untuk masalah Anda di sana, hubungi Dukungan. Untuk informasi selengkapnya, lihat Membuat kasus dukungan di Panduan AWS Dukungan Pengguna.

Dukungan firewall untuk Amplify situs yang dihosting

Dukungan firewall untuk Amplify situs host memungkinkan Anda untuk melindungi aplikasi web Anda dengan integrasi langsung dengan. AWS WAF AWS WAF memungkinkan Anda mengonfigurasi seperangkat aturan, yang disebut daftar kontrol akses web (web ACL), yang memungkinkan, memblokir, atau memantau (menghitung) permintaan web berdasarkan aturan dan kondisi keamanan web yang dapat disesuaikan yang Anda tentukan. Saat mengintegrasikan aplikasi Amplify AWS WAF, Anda mendapatkan lebih banyak kontrol dan visibilitas ke lalu lintas HTTP yang diterima oleh aplikasi Anda. Untuk mempelajari selengkapnya AWS WAF, lihat Cara AWS WAF Kerja di Panduan AWS WAF Pengembang.

Dukungan firewall tersedia Wilayah AWS di semua tempat Amplify Hosting beroperasi. Integrasi ini berada di bawah sumber daya AWS WAF global, mirip dengan CloudFront. Web ACLs dapat dilampirkan ke beberapa aplikasi Amplify Hosting, tetapi mereka harus berada di Wilayah yang sama.

Anda dapat menggunakannya AWS WAF untuk melindungi aplikasi Amplify Anda dari eksploitasi web umum, seperti injeksi SQL dan skrip lintas situs. Hal ini dapat memengaruhi ketersediaan dan kinerja aplikasi Anda, membahayakan keamanan, atau menggunakan sumber daya yang berlebihan. Misalnya, Anda dapat membuat aturan untuk mengizinkan atau memblokir permintaan dari rentang alamat IP tertentu, permintaan dari blok CIDR, permintaan yang berasal dari negara atau wilayah tertentu, atau permintaan yang berisi kode SQL atau skrip yang tidak terduga.

Anda juga dapat membuat aturan yang cocok dengan string tertentu atau pola ekspresi reguler di header HTTP, metode, string kueri, URI, dan badan permintaan (terbatas pada 8 KB pertama). Selain itu, Anda dapat membuat aturan untuk memblokir acara dari agen pengguna, bot, dan pencakar konten tertentu. Misalnya, Anda dapat menggunakan aturan berbasis tarif untuk menentukan jumlah permintaan web yang diizinkan oleh setiap IP klien dalam periode 5 menit yang terus diperbarui.

Untuk mempelajari lebih lanjut tentang jenis aturan yang didukung dan AWS WAF fitur tambahan, lihat Panduan AWS WAF Pengembang dan Referensi AWS WAF API.

> **Important**
>
> Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. AWS WAF bukanlah solusi untuk semua masalah keamanan internet dan Anda harus mengonfigurasinya untuk memenuhi tujuan keamanan dan kepatuhan Anda. Untuk membantu Anda memahami cara menerapkan model tanggung jawab bersama saat

menggunakan AWS WAF, lihat Keamanan dalam penggunaan AWS WAF layanan oleh Anda.

Topik

- Mengaktifkan AWS WAF aplikasi Amplify di AWS Management Console
- Putuskan hubungan ACL web dari aplikasi Amplify
- Mengaktifkan AWS WAF aplikasi Amplify menggunakan AWS CDK
- Bagaimana Amplify terintegrasi dengan AWS WAF
- Harga firewall untuk aplikasi Amplify

Mengaktifkan AWS WAF aplikasi Amplify di AWS Management Console

Anda dapat mengaktifkan AWS WAF perlindungan untuk aplikasi Amplify baik di konsol Amplify atau di konsol. AWS WAF

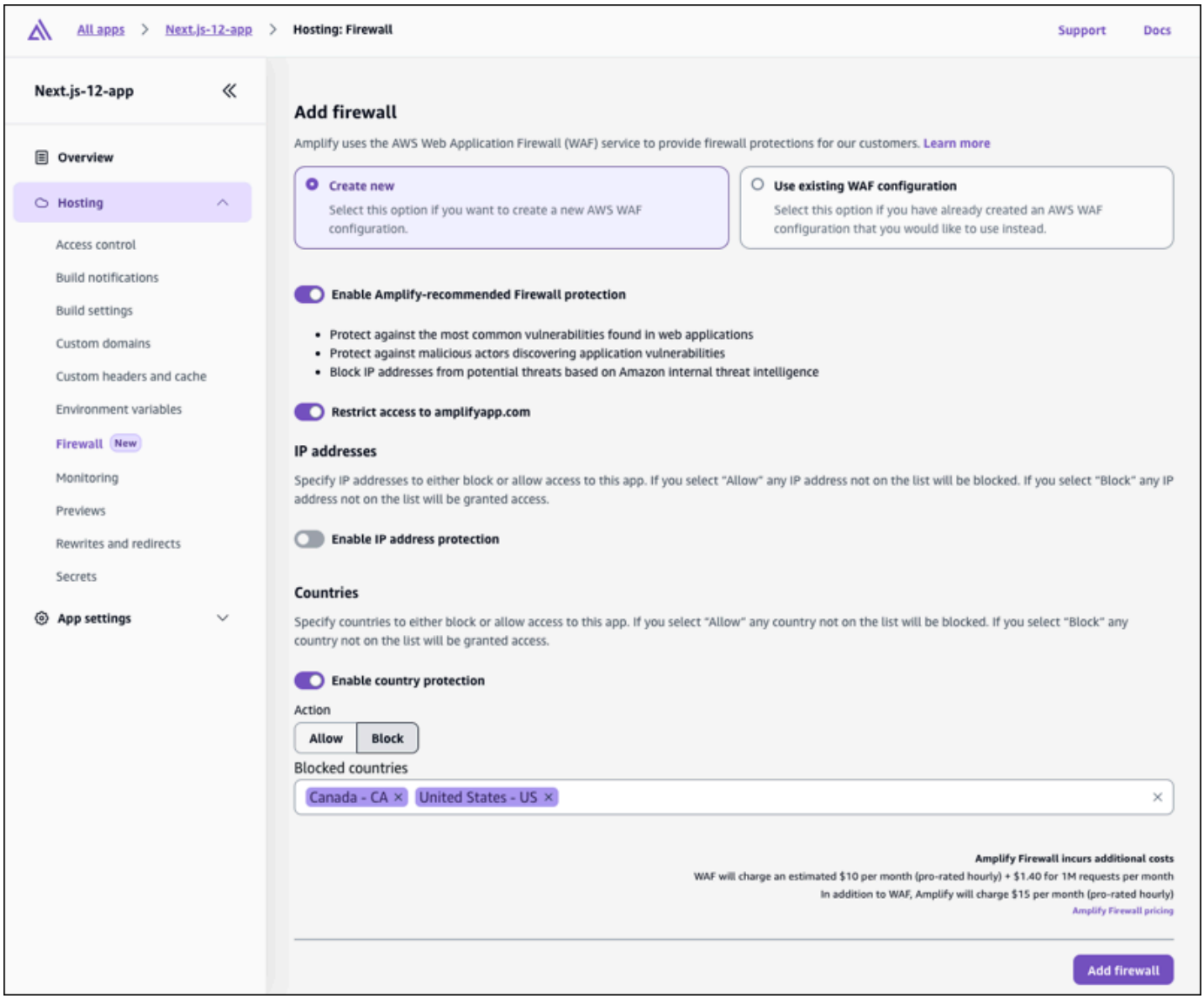
- Amplify console — Anda dapat mengaktifkan kemampuan Firewall untuk aplikasi Amplify yang ada dengan mengaitkan ACL AWS WAF web ke aplikasi Anda di konsol Amplify. Gunakan perlindungan satu klik untuk membuat ACL web dengan aturan pra-konfigurasi yang kami anggap sebagai praktik terbaik untuk sebagian besar aplikasi. Anda memiliki opsi untuk menyesuaikan akses berdasarkan alamat IP dan negara. Petunjuk di bagian ini menjelaskan pengaturan perlindungan satu klik.
- AWS WAF konsol — Gunakan ACL web yang telah dikonfigurasi sebelumnya yang Anda buat di AWS WAF konsol atau dengan menggunakan. AWS WAF APIs Anda harus membuat web ACLs yang ingin Anda kaitkan dengan aplikasi Amplify di Wilayah Global (CloudFront). Web regional ACLs mungkin sudah ada di Anda Akun AWS, tetapi tidak kompatibel dengan Amplify. Untuk petunjuk memulai, lihat Menyiapkan AWS WAF dan komponennya di Panduan AWS WAF Pengembang.

Gunakan prosedur berikut AWS WAF untuk mengaktifkan aplikasi yang ada di konsol Amplify.

AWS WAF Aktifkan aplikasi Amplify yang ada

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Pada halaman Semua aplikasi, pilih nama aplikasi yang digunakan untuk mengaktifkan fitur Firewall.
3. Di panel navigasi, pilih Hosting, lalu pilih Firewall.

   Tangkapan layar berikut menunjukkan cara menavigasi ke halaman Add firewall di konsol Amplify.



4. Pada halaman Add firewall, tindakan Anda akan tergantung pada apakah Anda ingin membuat AWS WAF konfigurasi baru atau menggunakan yang sudah ada.

- Buat AWS WAF konfigurasi baru.

  a. Pilih Buat baru.

  b. Secara opsional, aktifkan salah satu konfigurasi berikut:

     i. Aktifkan Aktifkan perlindungan Firewall yang direkomendasikan oleh Amplift.

     ii. Aktifkan Batasi akses ke amplifyapp.com untuk mencegah akses ke aplikasi Anda di domain Amplify default.

     iii. Untuk alamat IP, aktifkan Aktifkan perlindungan alamat IP.

        A. Untuk Tindakan, pilih Izinkan jika Anda ingin menentukan alamat IP yang akan memiliki akses dan semua yang lain akan diblokir. Pilih Blokir jika Anda ingin menentukan alamat IP yang akan diblokir dan semua yang lain akan memiliki akses.

        B. Untuk versi IP, pilih salah satu IPV4atau IPV6.

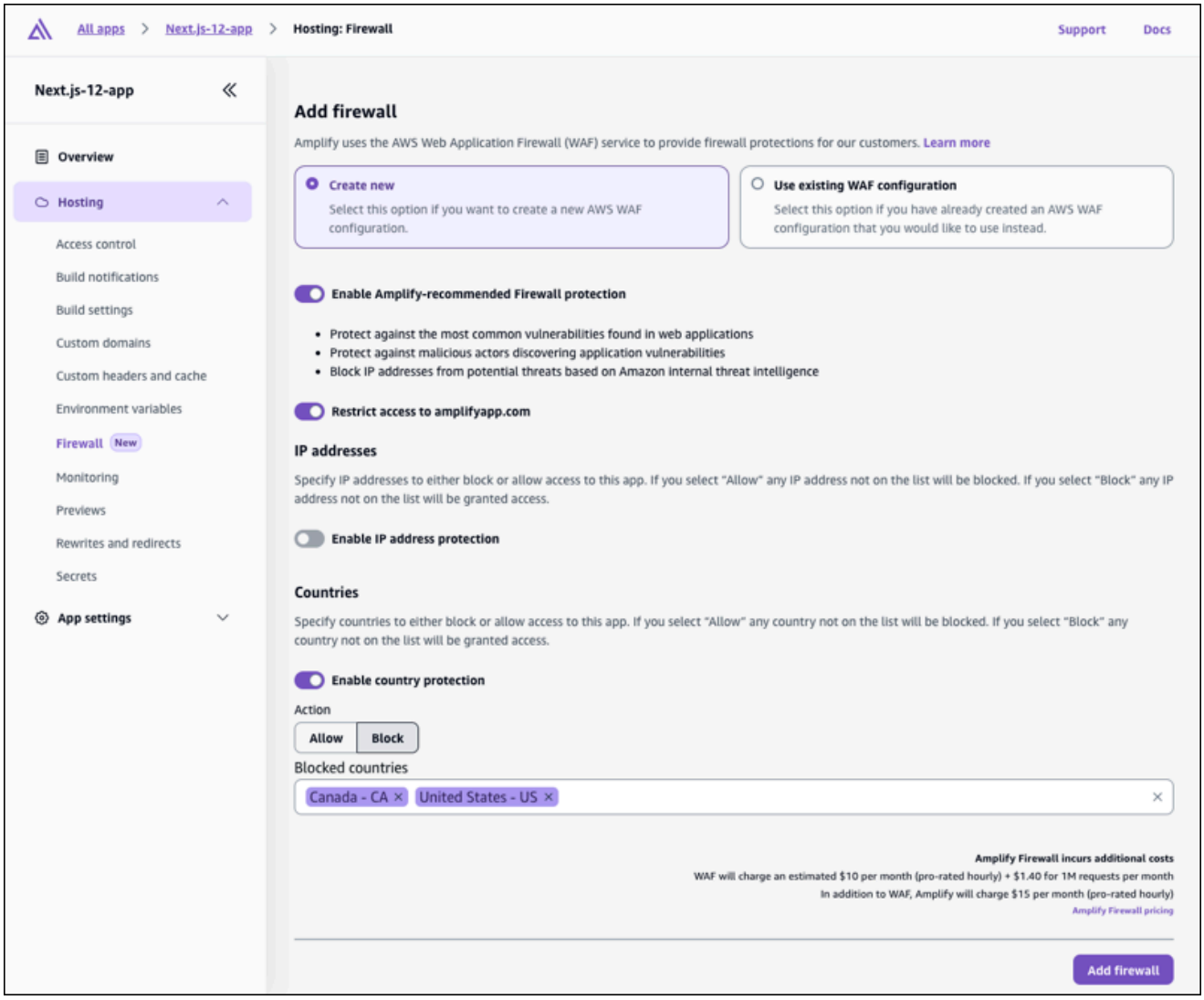
        C. Di kotak teks alamat IP, masukkan alamat IP yang diizinkan atau diblokir, satu per baris, dalam format CIDR.

     iv. Untuk Negara, aktifkan Aktifkan perlindungan negara.

        A. Untuk Tindakan, pilih Izinkan jika Anda ingin menentukan negara yang akan memiliki akses dan semua negara lain akan diblokir. Pilih Blokir jika Anda ingin menentukan negara yang akan diblokir dan semua negara lain akan memiliki akses.

        B. Untuk Negara, pilih negara yang diizinkan atau diblokir dari daftar.

  Screenshot berikut menunjukkan cara mengaktifkan AWS WAF konfigurasi baru untuk aplikasi.



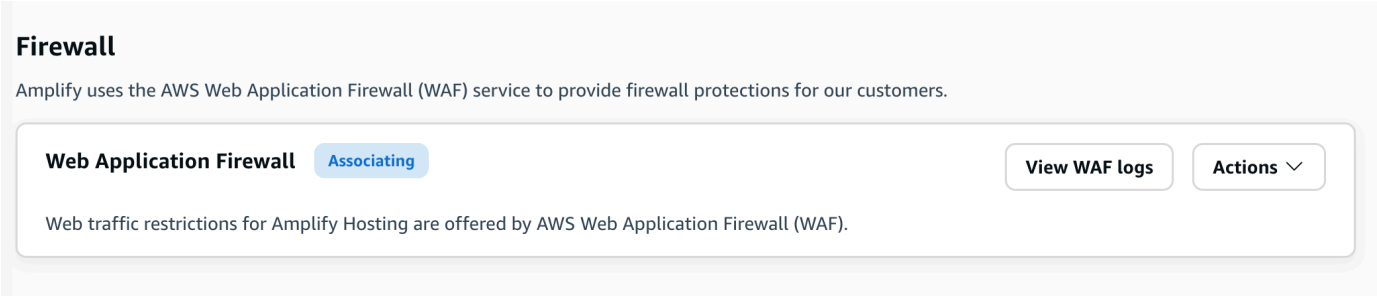
- Gunakan AWS WAF konfigurasi yang ada.

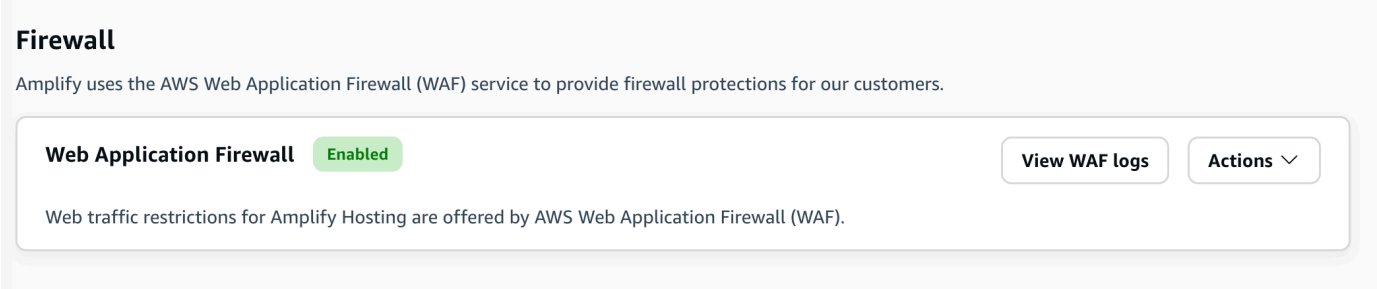
  a. Pilih Gunakan AWS WAF konfigurasi yang ada.

  b. Pilih konfigurasi yang disimpan dari daftar web ACLs di AWS WAF dalam Anda Akun AWS. ACL web yang Anda kaitkan dengan aplikasi Amplify harus dibuat di Wilayah Global CloudFront (). Web regional ACLs mungkin sudah ada di Anda Akun AWS, tetapi tidak kompatibel dengan Amplify.

5. Pilih Tambahkan firewall.
6. Pada halaman Firewall, status Associating ditampilkan untuk menunjukkan bahwa AWS WAF pengaturan sedang disebarkan. Ketika proses selesai, status berubah menjadi Diaktifkan.

   Tangkapan layar berikut menunjukkan status kemajuan firewall di konsol Amplify, yang menunjukkan kapan konfigurasi Mengaitkan AWS WAF dan Diaktifkan.





Putuskan hubungan ACL web dari aplikasi Amplify

Anda tidak dapat menghapus ACL web yang terkait dengan aplikasi Amplify. Anda harus terlebih dahulu memisahkan ACL web dari aplikasi di konsol Amplify. Kemudian Anda dapat menghapusnya di AWS WAF konsol.

Untuk memisahkan ACL web dari aplikasi Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Pada halaman Semua aplikasi, pilih nama aplikasi untuk memisahkan ACL web.
3. Di panel navigasi, pilih Hosting, lalu pilih Firewall.
4. Pada halaman Firewall, pilih Actions, lalu pilih Disassociate firewall.
5. Di modal konfirmasi, masukkan**`disassociate`**, lalu pilih Disassociate firewall.
6. Pada halaman Firewall, status Disassociating ditampilkan untuk menunjukkan bahwa AWS WAF pengaturan sedang disebarkan.

   Ketika proses selesai, Anda dapat menghapus ACL web di AWS WAF konsol.

Mengaktifkan AWS WAF aplikasi Amplify menggunakan AWS CDK

Anda dapat menggunakan AWS Cloud Development Kit (AWS CDK) AWS WAF untuk mengaktifkan aplikasi Amplify. Untuk mempelajari lebih lanjut tentang menggunakan CDK, lihat Apa itu CDK? di Panduan AWS Cloud Development Kit (AWS CDK) Pengembang.

Contoh TypeScript kode berikut menunjukkan cara membuat AWS CDK aplikasi dengan dua tumpukan CDK: satu untuk Amplify dan satu untuk. AWS WAF Perhatikan bahwa AWS WAF tumpukan harus dikerahkan ke Wilayah AS Timur (Virginia N.) (us-east-1). Tumpukan aplikasi Amplify dapat digunakan ke Wilayah yang berbeda. Anda harus membuat ACL web yang ingin Anda kaitkan dengan aplikasi Amplify di Wilayah Global CloudFront (). Web regional ACLs mungkin sudah ada di Anda Akun AWS, tetapi tidak kompatibel dengan Amplify.

```
import * as cdk from "aws-cdk-lib";
import { Construct } from "constructs";
import * as wafv2 from "aws-cdk-lib/aws-wafv2";
import * as amplify from "aws-cdk-lib/aws-amplify";

interface WafStackProps extends cdk.StackProps {
  appArn: string;
}

export class AmplifyStack extends cdk.Stack {
  public readonly appArn: string;
  constructor(scope: Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);
    const amplifyApp = new amplify.CfnApp(this, "AmplifyApp", {
      name: "MyApp",
    });
    this.appArn = amplifyApp.attrArn;
  }
}

export class WAFStack extends cdk.Stack {
  constructor(scope: Construct, id: string, props: WafStackProps) {
    super(scope, id, props);
    const webAcl = new wafv2.CfnWebACL(this, "WebACL", {
      defaultAction: { allow: {} },
      scope: "CLOUDFRONT",
      rules: [
        // Add your own rules here.
      ],
```

```
      visibilityConfig: {
        cloudWatchMetricsEnabled: true,
        metricName: "my-metric-name",
        sampledRequestsEnabled: true,
      },
    });

    new wafv2.CfnWebACLAssociation(this, "WebACLAssociation", {
      resourceArn: props.appArn,
      webAclArn: webAcl.attrArn,
    });
  }
}

const app = new cdk.App();

// Create AmplifyStack in your desired Region.
const amplifyStack = new AmplifyStack(app, 'AmplifyStack', {
  env: { region: 'us-west-2' },
});

// Create WAFStack in IAD region, passing appArn from AmplifyStack.
new WAFStack(app, 'WAFStack', {
  env: { region: 'us-east-1' },
  crossRegionReferences: true,

  appArn: amplifyStack.appArn,  // Pass appArn from AmplifyStack.
});
```

Bagaimana Amplify terintegrasi dengan AWS WAF

Daftar berikut memberikan rincian spesifik tentang bagaimana dukungan Firewall terintegrasi dengan AWS WAF dan kendala yang perlu dipertimbangkan saat membuat web ACLs dan mengaitkannya dengan aplikasi Amplify.

- Anda dapat mengaktifkan AWS WAF untuk semua jenis aplikasi Amplify. Ini termasuk kerangka kerja yang didukung, aplikasi yang dirender sisi server (SSR), dan situs yang sepenuhnya statis. AWS WAF didukung untuk aplikasi Amplify Gen 1 dan Gen 2.
- Anda harus membuat web ACLs yang ingin Anda kaitkan dengan aplikasi Amplify di Wilayah Global (CloudFront). Web regional ACLs mungkin sudah ada di Anda Akun AWS, tetapi tidak kompatibel dengan Amplify.

- ACL web dan aplikasi Amplify harus dibuat dalam hal yang sama. Akun AWS Anda dapat menggunakan AWS Firewall Manager untuk mereplikasi AWS WAF aturan di seluruh Akun AWS, untuk menyederhanakan menjaga aturan organisasi tetap terpusat dan didistribusikan di beberapa. Akun AWS Untuk informasi lebih lanjut, lihat AWS Firewall Manager dalam Panduan Pengembang AWS WAF .
- Anda dapat berbagi ACL web yang sama di beberapa aplikasi Amplify secara bersamaan. Akun AWS Semua aplikasi harus berada di Wilayah yang sama.
- Saat Anda mengaitkan ACL web dengan aplikasi Amplify, ACL web akan menempel ke setiap cabang di aplikasi secara default. Saat Anda membuat cabang baru, mereka akan memiliki ACL web.
- Saat Anda mengaitkan ACL web ke aplikasi Amplify, ACL akan secara otomatis dikaitkan dengan semua domain aplikasi. Namun, Anda dapat mengonfigurasi aturan yang berlaku untuk satu nama domain menggunakan aturan pencocokan host-header.
- Anda tidak dapat menghapus ACL web yang terkait dengan aplikasi Amplify. Sebelum menghapus ACL web di AWS WAF konsol, Anda harus memisahkannya dari aplikasi.

Amplify kebijakan sumber daya ACL web

Untuk memungkinkan Amplify mengakses ACL web Anda, kebijakan sumber daya dilampirkan ke ACL web selama asosiasi. Amplify membuat kebijakan sumber daya ini secara otomatis, tetapi Anda dapat melihatnya menggunakan API. AWS WAFV2 GetPermissionPolicy Izin IAM berikut diperlukan untuk mengaitkan ACL web ke aplikasi Amplify.

- memperkuat: AssociateWeb ACL
- wafv2: ACL AssociateWeb
- wafv2: PutPermissionPolicy
- wafv2: GetPermissionPolicy

Harga firewall untuk aplikasi Amplify

Biaya penerapan AWS WAF pada aplikasi Amplify dihitung berdasarkan dua komponen berikut:

- AWS WAF penggunaan - Anda akan dikenakan biaya untuk AWS WAF penggunaan Anda sesuai dengan model harga. AWS WAF AWS WAF Biaya didasarkan pada daftar kontrol akses web (web

ACLs) yang Anda buat, jumlah aturan yang Anda tambahkan per ACL web, dan jumlah permintaan web yang Anda terima. Untuk detail harga, lihat AWS WAF Harga.

- Biaya integrasi Amplify Hosting - Ada $15.00 per bulan, per biaya aplikasi saat Anda melampirkan ACL web ke aplikasi Amplify. Ini diprorata setiap jam.

Deployment cabang fitur dan alur kerja tim

Amplify Hosting dirancang untuk bekerja dengan cabang fitur dan GitFlow alur kerja. Amplify menggunakan cabang Git untuk membuat penerapan baru setiap kali Anda menghubungkan cabang baru di repositori Anda. Setelah Anda menghubungkan cabang pertama Anda, Anda membuat cabang fitur tambahan.

Untuk menambahkan cabang ke aplikasi

1. Pilih aplikasi yang ingin Anda tambahkan cabang.
2. Pilih Pengaturan aplikasi, lalu Pengaturan cabang.
3. Pada halaman Pengaturan cabang, pilih Tambah cabang.
4. Pilih cabang dari repositori Anda.
5. Pilih Tambah cabang.
6. Menerapkan ulang aplikasi Anda.

Setelah menambahkan cabang, aplikasi Anda memiliki dua penerapan yang tersedia di domain default Amplify, seperti dan. https://main.appid.amplifyapp.comhttps://dev.appid.amplifyapp.com Ini mungkin berbeda dari team-to-team, tetapi biasanya cabang utama melacak kode rilis dan merupakan cabang produksi Anda. Cabang develop digunakan sebagai cabang integrasi untuk menguji fitur baru. Ini memungkinkan penguji beta untuk menguji fitur yang belum dirilis pada penerapan cabang pengembangan, tanpa memengaruhi pengguna akhir produksi mana pun pada penerapan cabang utama.

Topik

- Alur kerja tim dengan aplikasi Amplify Gen 2 fullstack
- Alur kerja tim dengan aplikasi Amplify Gen 1 fullstack
- Deployment cabang fitur berbasis pola
- Pembuatan waktu pembuatan otomatis konfigurasi Amplify (hanya aplikasi Gen 1)
- Build backend bersyarat (hanya aplikasi Gen 1)
- Gunakan backend Amplify di seluruh aplikasi (hanya aplikasi Gen 1)

Alur kerja tim dengan aplikasi Amplify Gen 2 fullstack

AWS Amplify Gen 2 memperkenalkan pengalaman pengembang TypeScript berbasis kode pertama untuk mendefinisikan backend. Untuk mempelajari alur kerja fullstack dengan aplikasi Amplify Gen 2, lihat Alur kerja Fullstack di dokumen Amplify.

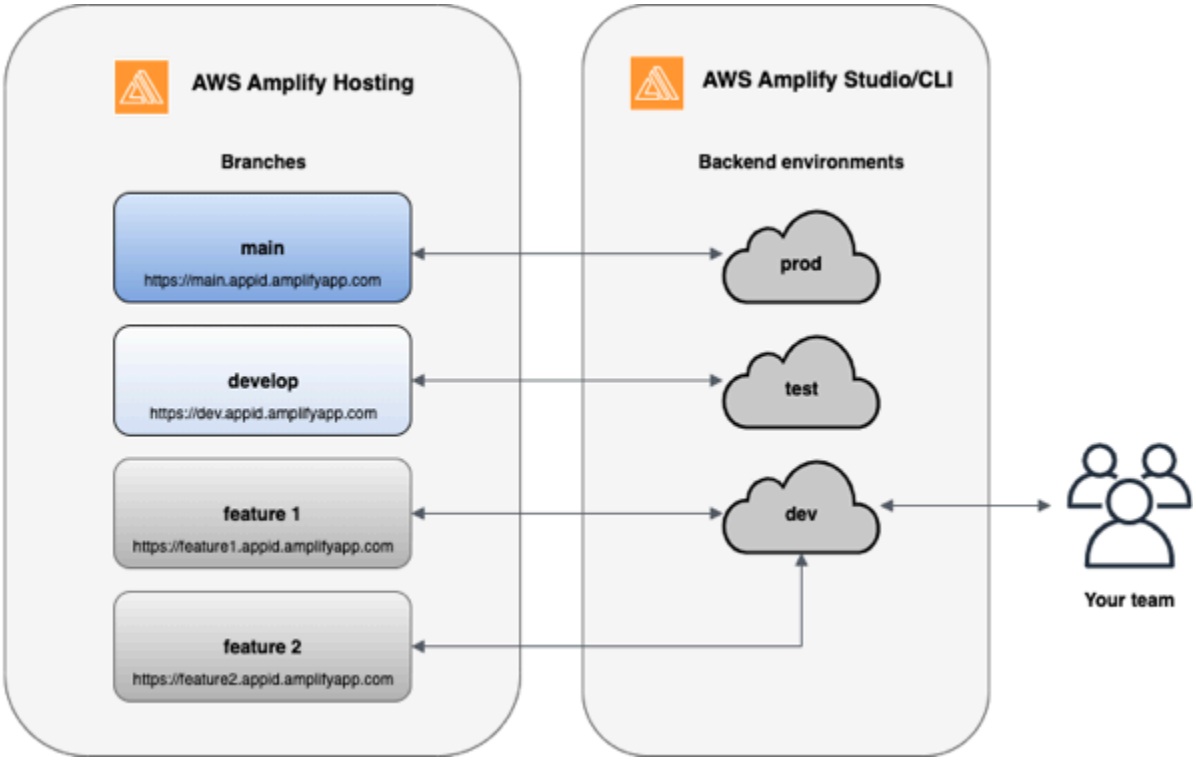
Alur kerja tim dengan aplikasi Amplify Gen 1 fullstack

Penyebaran cabang fitur terdiri dari frontend, dan lingkungan backend opsional. Frontend dibangun dan disebarkan ke jaringan pengiriman konten global (CDN), sedangkan backend digunakan oleh Amplify Studio atau Amplify CLI ke. AWS Untuk mempelajari cara mengatur skenario penerapan ini, lihatMembangun backend untuk aplikasi.

Amplify Hosting terus menerapkan sumber daya backend seperti fungsi GraphQL dan APIs Lambda dengan penerapan cabang fitur Anda. Anda dapat menggunakan model percabangan berikut untuk menerapkan backend dan frontend Anda dengan Amplify Hosting.

Alur kerja cabang fitur

- Buat lingkungan backend prod, test, dan dev dengan Amplify Studio atau Amplify CLI.
- Petakan backend prod ke cabang utama.
- Petakan backend pengujian ke cabang develop.
- Anggota tim dapat menggunakan lingkungan backend dev untuk menguji cabang fitur individual.



1. Instal CLI Amplify untuk memulai proyek Amplify baru.

```
npm install -g @aws-amplify/cli
```

2. Mulai lingkungan backend prod untuk proyek Anda. Jika Anda tidak memiliki proyek, buat satu menggunakan alat bootstrap seperti create-react-app atau Gatsby.

```
create-react-app next-unicorn
cd next-unicorn
amplify init
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: prod
...
amplify push
```

3. Tambahkan lingkungan backend test dan dev.

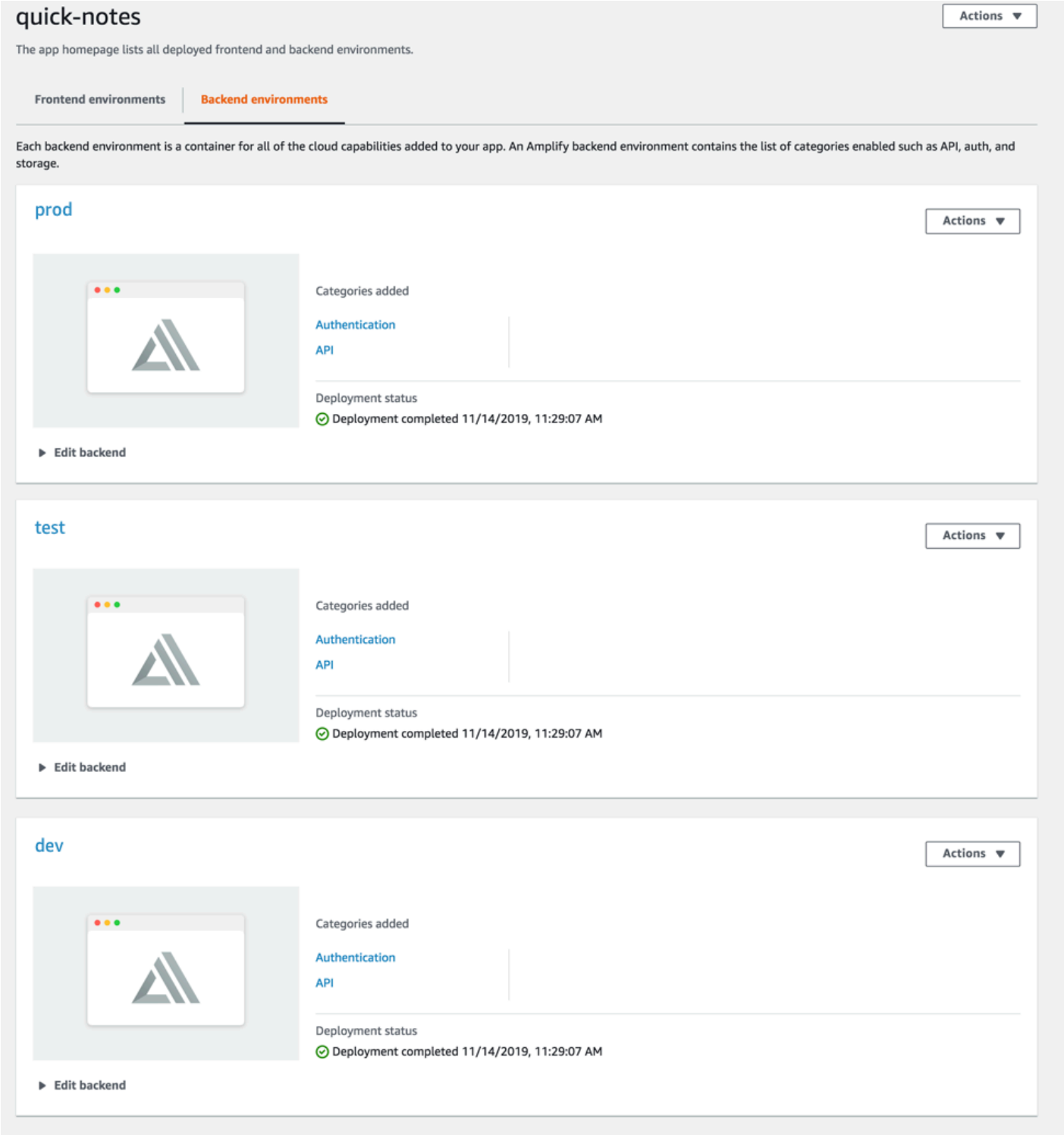
```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: test
...
amplify push
```

```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: dev
...
amplify push
```

4. Dorong kode ke repositori Git pilihan Anda (dalam contoh ini, kami menganggap Anda mendorong kode ke main).

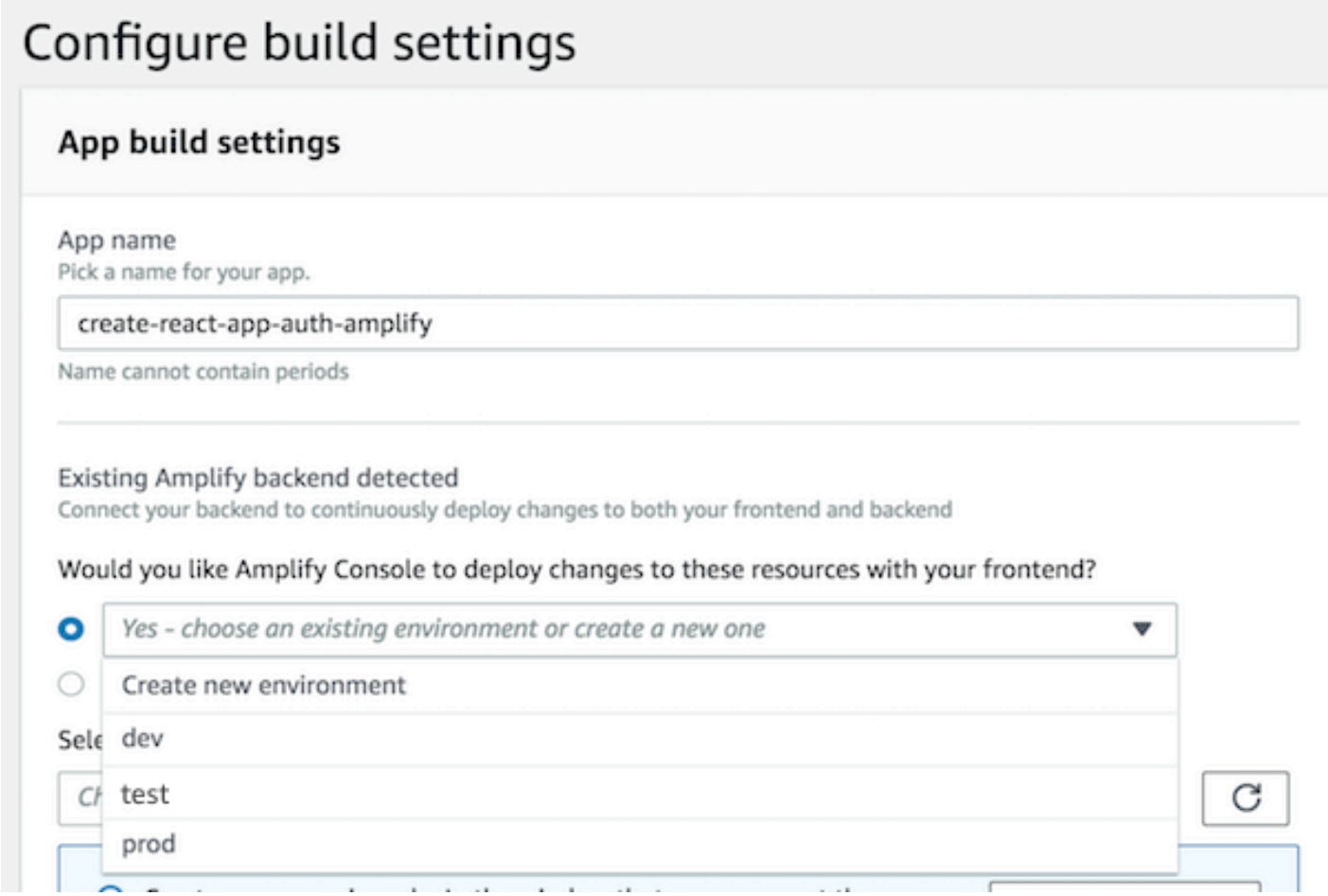
```
git commit -am 'Added dev, test, and prod environments'
git push origin main
```

5. Kunjungi Amplify di AWS Management Console untuk melihat lingkungan backend Anda saat ini. Navigasi ke satu tingkat di atas dari breadcrumb untuk melihat daftar semua lingkungan backend yang dibuat di tab Lingkungan backend.

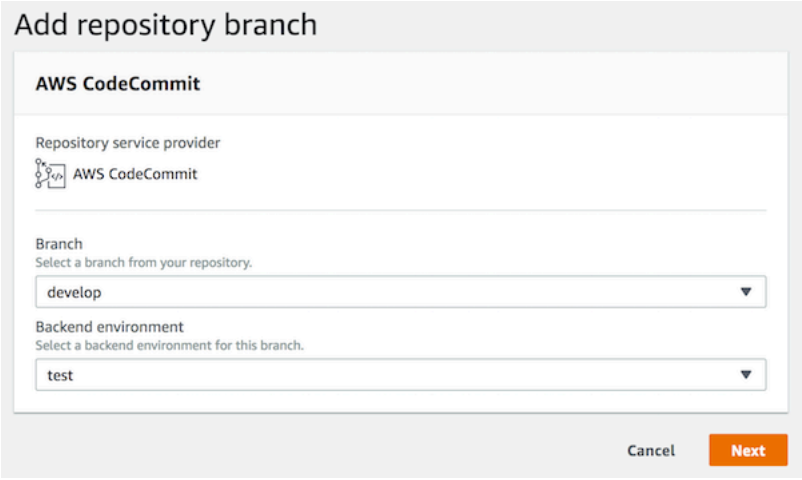


6. Beralih ke tab Lingkungan frontend, lalu hubungkan penyedia repositori dan cabang main.
7. Pada halaman pengaturan build, pilih lingkungan backend yang ada untuk menyiapkan penerapan berkelanjutan dengan cabang utama. Pilih prod dari daftar dan berikan peran layanan ke Amplify.

Pilih Simpan dan deploy. Setelah build selesai, Anda akan mendapatkan penerapan cabang utama yang tersedia di. https://main.appid.amplifyapp.com



8. Connect develop branch di Amplify (asumsikan develop dan main branch adalah sama pada saat ini). Pilih lingkungan backend test.



9. Amplify sekarang sudah diatur. Anda dapat mulai mengerjakan fitur baru di cabang fitur. Tambahkan fungsionalitas backend menggunakan lingkungan backend dev dari workstation lokal Anda.

```
git checkout -b newinternet
amplify env checkout dev
amplify add api
...
amplify push
```

10. Setelah pengerjaan fitur selesai, terapkan kode dan buat permintaan tarik untuk ditinjau secara internal.

```
git commit -am 'Decentralized internet v0.1'
git push origin newinternet
```

11. Untuk melihat seperti apa perubahan itu, buka konsol Amplify dan hubungkan cabang fitur Anda. Catatan: Jika Anda telah AWS CLI menginstal pada sistem Anda (Bukan Amplify CLI), Anda dapat menghubungkan cabang langsung dari terminal Anda. Anda dapat menemukan appid dengan membuka Pengaturan aplikasi > Umum > AppARN: arn:aws:amplify:<region>:<region>:apps/<appid>

```
aws amplify create-branch --app-id <appid> --branch-name <branchname>
aws amplify start-job --app-id <appid> --branch-name <branchname> --job-type RELEASE
```

12. Fitur Anda akan dapat diakses di https://newinternet.appid.amplifyapp.comuntuk dibagikan dengan rekan tim Anda. Jika tidak ada masalah, gabungkan PR ke cabang develop.

```
git checkout develop
git merge newinternet
git push
```

13. Ini akan memulai build yang akan memperbarui backend serta frontend di Amplify dengan penerapan cabang di. https://dev.appid.amplifyapp.com Anda dapat membagikan tautan ini kepada pemangku kepentingan internal agar mereka dapat meninjau fitur baru.

14. Hapus cabang fitur Anda dari Git, Amplify, dan hapus lingkungan backend dari cloud (Anda selalu dapat memutar yang baru berdasarkan dengan menjalankan 'amplify env checkout prod' dan menjalankan 'amplify env add').

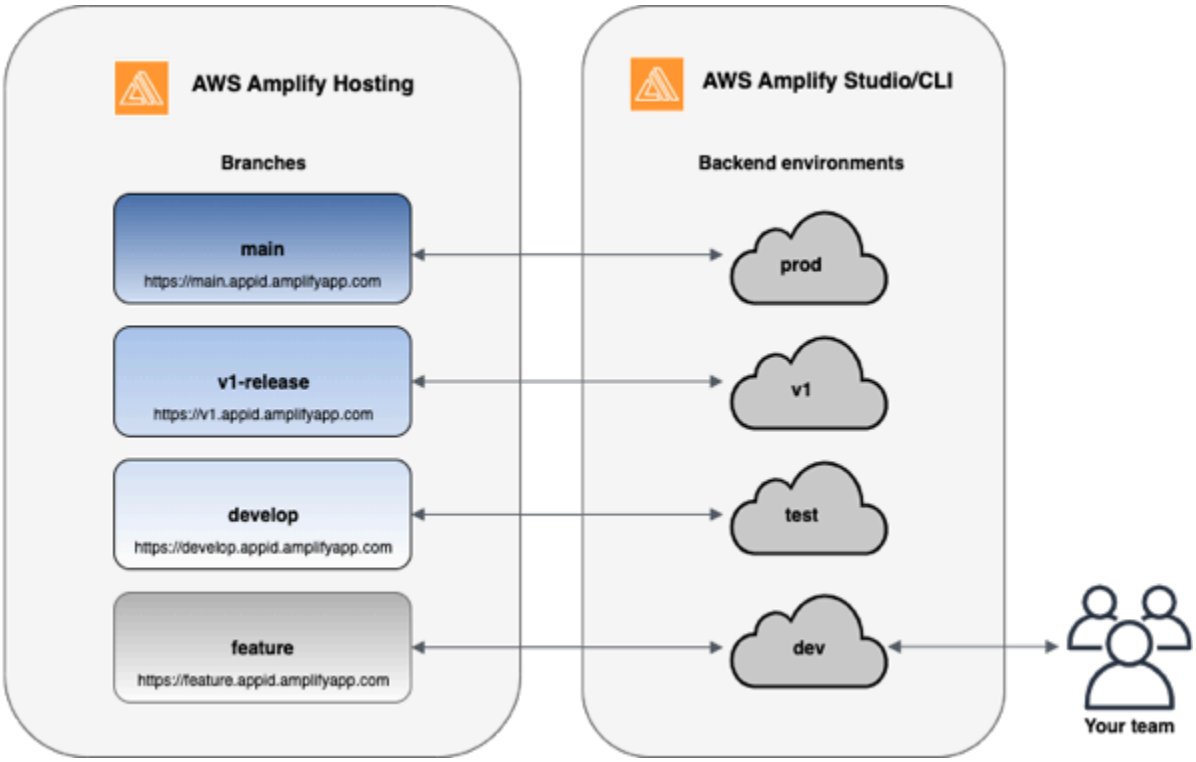
```
git push origin --delete newinternet
```

```
aws amplify delete-branch --app-id <appid> --branch-name <branchname>
amplify env remove dev
```

GitFlow alur kerja

GitFlow menggunakan dua cabang untuk mencatat sejarah proyek. Cabang utama hanya melacak kode rilis, dan cabang pengembangan digunakan sebagai cabang integrasi untuk fitur baru. GitFlow menyederhanakan pengembangan paralel dengan mengisolasi pengembangan baru dari pekerjaan yang telah selesai. Pengembangan baru (seperti fitur dan perbaikan bug non-darurat) dilakukan di cabang fitur. Ketika developer siap merilis kode, cabang fitur akan digabungkan kembali ke cabang develop integrasi. Satu-satunya penerapan ke cabang main adalah penggabungan dari cabang release dan cabang hotfix (untuk perbaikan bug darurat).

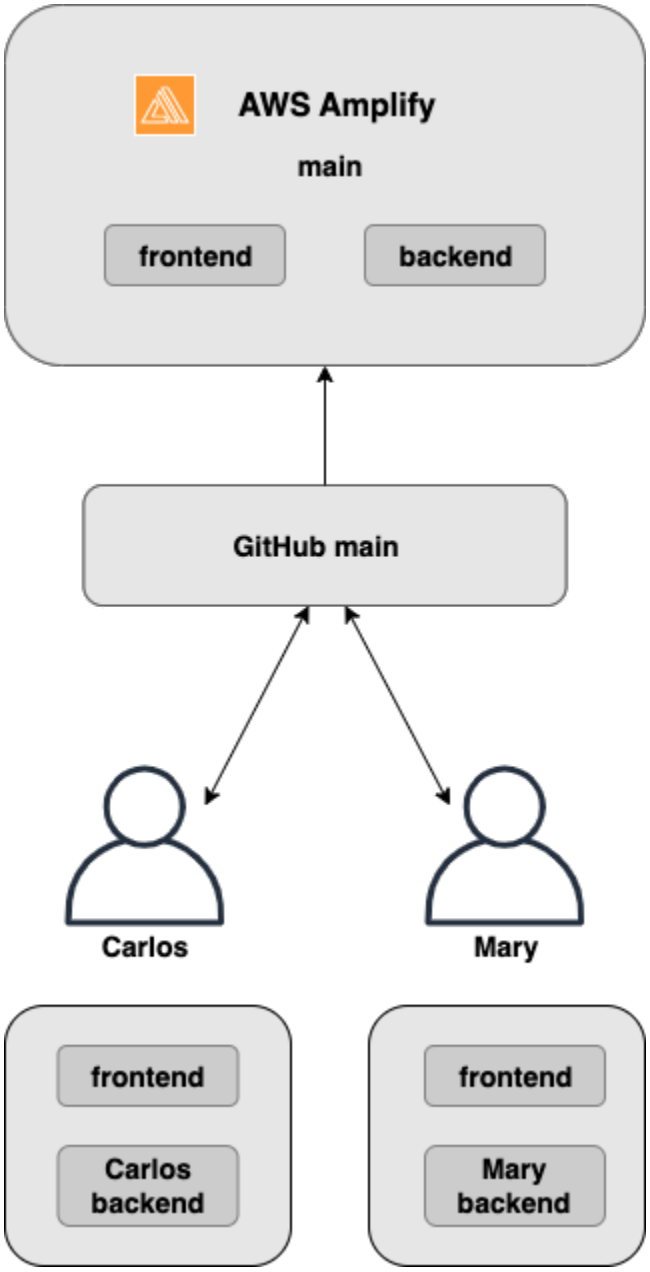
Diagram di bawah ini menunjukkan pengaturan yang disarankan dengan GitFlow. Anda dapat mengikuti proses yang sama seperti yang dijelaskan di bagian alur kerja cabang fitur di atas.



Sandbox per developer

- Setiap developer dalam tim menciptakan lingkungan sandbox di cloud yang terpisah dari komputer lokal mereka. Dengan cara ini, developer dapat bekerja masing-masing tanpa menimpa perubahan anggota tim lainnya.

- Setiap cabang di Amplify memiliki backend sendiri. Ini memastikan bahwa Amplify menggunakan repositori Git sebagai sumber kebenaran tunggal untuk menyebarkan perubahan, daripada mengandalkan pengembang di tim untuk secara manual mendorong backend atau front end mereka ke produksi dari komputer lokal mereka.



1. Instal CLI Amplify untuk memulai proyek Amplify baru.

```
npm install -g @aws-amplify/cli
```

2. Inisialisasi lingkungan backend mary untuk proyek Anda. Jika Anda tidak memiliki proyek, buat satu menggunakan alat bootstrap seperti create-react-app atau Gatsby.

```
cd next-unicorn
amplify init
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: mary
...
amplify push
```

3. Dorong kode ke repositori Git pilihan Anda (dalam contoh ini kita akan menganggap Anda didorong ke main.

```
git commit -am 'Added mary sandbox'
git push origin main
```

4. Connect repo > main Anda ke Amplify.
5. Konsol Amplify akan mendeteksi lingkungan backend yang dibuat oleh Amplify CLI. Pilih Buat lingkungan baru dari dropdown dan berikan peran layanan ke Amplify. Pilih Simpan dan deploy. Setelah build selesai, Anda akan mendapatkan penerapan cabang utama yang tersedia https://main.appid.amplifyapp.comdengan lingkungan backend baru yang ditautkan ke cabang.
6. Connect develop branch di Amplify (asumsikan develop dan main branch sama pada saat ini) dan pilih Create

Deployment cabang fitur berbasis pola

Penerapan cabang berbasis pola memungkinkan Anda untuk secara otomatis menerapkan cabang yang cocok dengan pola tertentu ke Amplify. Tim produk yang menggunakan cabang fitur atau GitFlow alur kerja untuk rilis mereka, sekarang dapat menentukan pola seperti **`release**`** untuk secara otomatis menyebarkan cabang Git yang dimulai dengan 'rilis' ke URL yang dapat dibagikan.

1. Pilih Pengaturan aplikasi, lalu Pengaturan cabang.
2. Pada halaman Pengaturan cabang, pilih Edit.
3. Pilih Deteksi otomatis cabang untuk secara otomatis menghubungkan cabang ke Amplify yang cocok dengan kumpulan pola.
4. Di kotak autodetection Cabang - pola, masukkan pola untuk menyebarkan cabang secara otomatis.

- **`*`** – Men-deploy semua cabang di repositori Anda.
- **`release*`**— Menyebarkan semua cabang yang dimulai dengan kata 'rilis'.
- **`release*/`** – Men-deploy semua cabang yang sesuai dengan pola ‘release /’.
- Tentukan beberapa pola dalam daftar yang dipisahkan koma. Misalnya, **`release*, feature*`**.

5. Siapkan perlindungan kata sandi otomatis untuk semua cabang yang dibuat secara otomatis dengan memilih kontrol akses deteksi otomatis Cabang.
6. Untuk aplikasi Gen 1 yang dibangun dengan backend Amplify, Anda dapat memilih untuk membuat lingkungan baru untuk setiap cabang yang terhubung, atau mengarahkan semua cabang ke backend yang ada.
7. Pilih Simpan.

Deployment cabang fitur berbasis pola untuk aplikasi yang terhubung ke domain kustom

Anda dapat menggunakan deployment cabang fitur berbasis pola untuk aplikasi yang terhubung ke domain kustom Amazon Route 53.

- Untuk langkah-langkah seputar pengaturan deployment cabang fitur berbasis pola, lihat Menyiapkan subdomain otomatis untuk domain kustom Amazon Route 53
- Untuk langkah-langkah seputar cara menghubungkan aplikasi Amplify ke domain kustom yang dikelola di Route 53, lihat Menambahkan domain kustom yang dikelola Amazon Route 53
- Untuk informasi lebih lanjut seputar penggunaan Route 53, lihat Tentang Amazon Route 53.

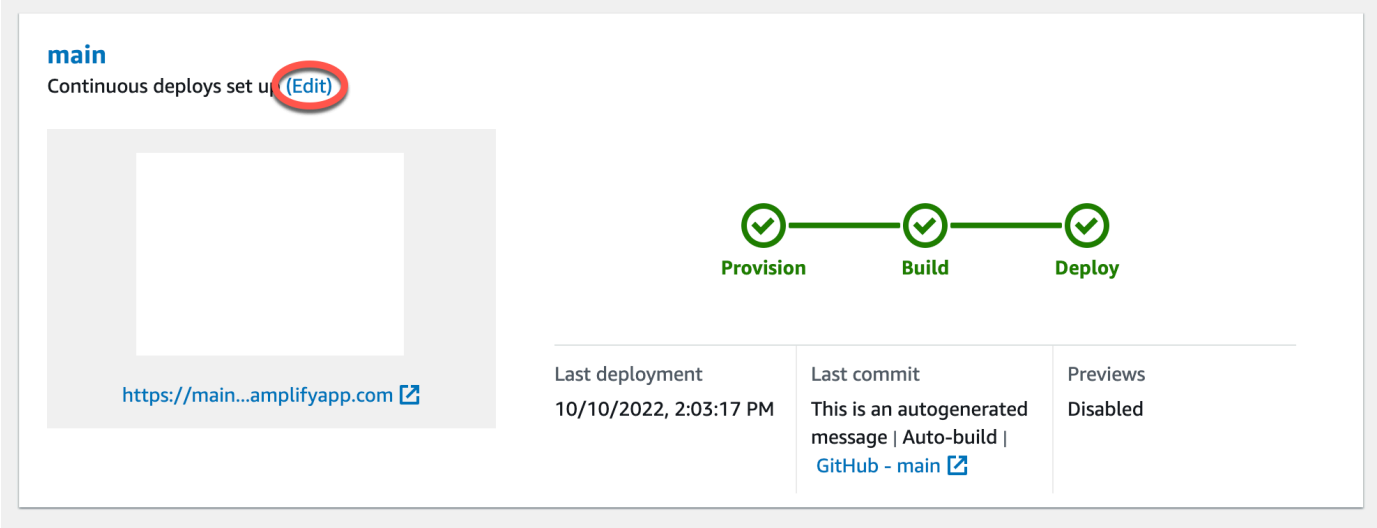
Pembuatan waktu pembuatan otomatis konfigurasi Amplify (hanya aplikasi Gen 1)

> **Note**
>
> Informasi di bagian ini hanya untuk aplikasi Gen 1. Jika Anda ingin menerapkan perubahan infrastruktur dan kode aplikasi secara otomatis dari cabang fitur untuk aplikasi Gen 2, lihat Penerapan cabang Fullstack di dokumen Amplify

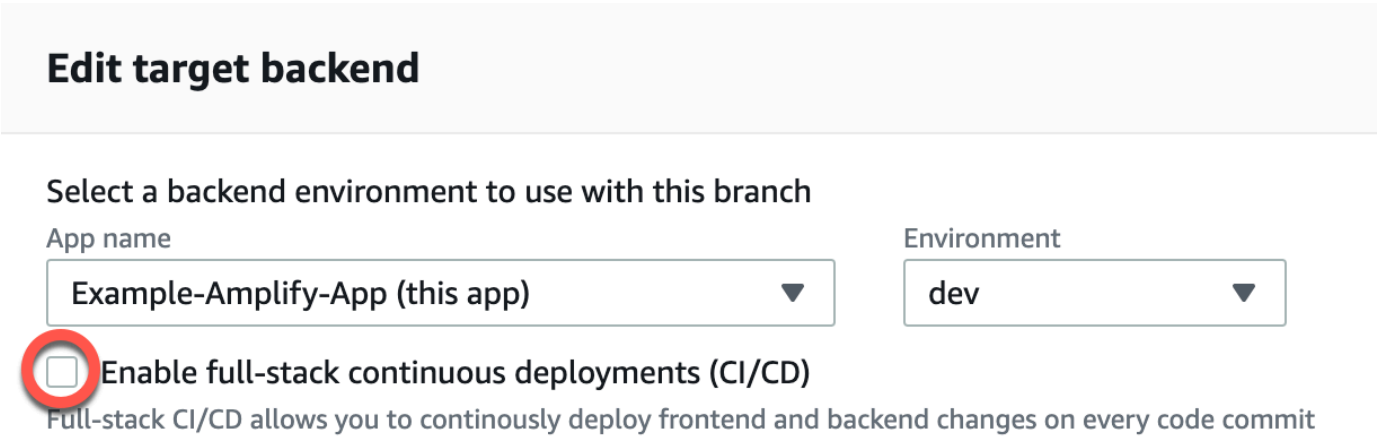
Amplify mendukung pembuatan waktu pembuatan otomatis dari file konfigurasi `aws-exports.js` Amplify untuk aplikasi Gen 1. Jika Anda menonaktifkan deployment CI/CD full stack, aplikasi Anda dapat membuat secara otomatis file `aws-exports.js` dan memastikan bahwa pembaruan tidak dilakukan ke backend Anda pada waktu build.

Cara membuat **`aws-exports.js`** secara otomatis pada waktu build

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang akan diedit.
3. Pilih tab Lingkungan hosting.
4. Tentukan cabang yang akan diedit, lalu pilih Edit.



5. Pada halaman backend target Edit, hapus centang Enable full-stack continuous deployments (CI/CD) untuk mematikan full-stack CI/CD untuk backend ini.



6. Pilih peran layanan yang ada untuk memberi Amplify izin yang diperlukan untuk membuat perubahan pada backend aplikasi Anda. Untuk membuat peran layanan, pilih Buat peran baru. Untuk informasi selengkapnya tentang pembuatan peran layanan, lihat Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend.
7. Pilih Simpan. Amplify menerapkan perubahan ini saat berikutnya Anda membuat aplikasi.

Build backend bersyarat (hanya aplikasi Gen 1)

> **Note**
>
> Informasi di bagian ini hanya untuk aplikasi Gen 1. Amplify Gen 2 memperkenalkan pengalaman pengembang TypeScript berbasis kode pertama. Oleh karena itu, fitur ini tidak diperlukan untuk backend Gen 2.

Amplify mendukung build backend bersyarat di semua cabang di aplikasi Gen 1. Untuk mengonfigurasi build backend bersyarat, atur variabel lingkungan `AMPLIFY_DIFF_BACKEND` ke `true`. Mengaktifkan build backend bersyarat membantu mempercepat build tempat perubahan dibuat hanya pada frontend.

Ketika Anda mengaktifkan build backend berbasis diff, di awal setiap build, Amplify mencoba untuk menjalankan diff di folder `amplify` dalam repositori Anda. Jika tidak menemukan perbedaan apa pun, Amplify akan melompati langkah build backend, dan tidak memperbarui sumber daya backend Anda. Jika proyek Anda tidak memiliki folder `amplify` di repositori, Amplify akan mengabaikan nilai variabel lingkungan `AMPLIFY_DIFF_BACKEND`. Untuk langkah-langkah seputar pengaturan variabel lingkungan `AMPLIFY_DIFF_BACKEND`, lihat Mengonfigurasi build backend berbasis diff untuk aplikasi Gen 1.

Jika saat ini Anda memiliki perintah khusus yang ditentukan dalam pengaturan build fase backend Anda, build backend bersyarat tidak akan berfungsi. Jika Anda ingin perintah kustom tersebut berjalan, Anda harus memindahkannya ke fase frontend setelan build di `amplify.yml` file aplikasi Anda. Untuk informasi selengkapnya tentang memperbarui `amplify.yml` file, lihatMembangun referensi spesifikasi.

Gunakan backend Amplify di seluruh aplikasi (hanya aplikasi Gen 1)

> **Note**
>
> Informasi di bagian ini hanya untuk aplikasi Gen 1. Jika Anda ingin berbagi sumber daya backend untuk aplikasi Gen 2, lihat Berbagi sumber daya di seluruh cabang di dokumen Amplify

Amplify memungkinkan Anda untuk menggunakan kembali lingkungan backend yang ada di semua aplikasi Gen 1 Anda di wilayah tertentu. Anda dapat melakukannya ketika membuat aplikasi baru, menghubungkan cabang baru ke aplikasi yang ada, atau memperbarui frontend yang ada agar mengarah ke lingkungan backend yang berbeda.

Menggunakan kembali backend saat membuat aplikasi baru

Cara menggunakan kembali backend saat membuat aplikasi Amplify baru

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Untuk membuat backend baru yang akan digunakan pada contoh ini, lakukan langkah-langkah berikut:
   a. Di panel navigasi, pilih Semua aplikasi.
   b. Pilih Aplikasi baru, Buat aplikasi.
   c. Masukkan nama untuk aplikasi Anda, seperti **`Example-Amplify-App`**.
   d. Pilih Konfirmasi deployment.
3. Untuk menghubungkan frontend ke backend baru Anda, pilih tab Lingkungan Hosting.
4. Pilih penyedia git Anda, kemudian pilih Hubungkan cabang.
5. Di halaman Tambahkan cabang repositori, untuk Repositori yang baru diperbarui, pilih nama repositori Anda. Untuk Cabang, pilih cabang dari repositori Anda untuk dihubungkan.
6. Pada pengaturan Build, halaman lakukan hal berikut:
   a. Untuk Nama aplikasi, pilih aplikasi yang akan digunakan untuk menambahkan lingkungan backend. Anda dapat memilih aplikasi saat ini atau aplikasi lain di wilayah saat ini.

b. Untuk Lingkungan, pilih nama lingkungan backend yang akan ditambahkan. Anda dapat menggunakan lingkungan yang sudah ada atau membuat lingkungan baru.

c. Secara default, tumpukan penuh CI/CD is turned off. Turning off full-stack CI/CD menyebabkan aplikasi berjalan dalam mode tarik saja. Pada waktu build, Amplify secara otomatis akan menghasilkan file `aws-exports.js` saja, tanpa memodifikasi lingkungan backend Anda.

d. Pilih peran layanan yang ada untuk memberi Amplify izin yang diperlukan untuk membuat perubahan pada backend aplikasi Anda. Untuk membuat peran layanan, pilih Buat peran baru. Untuk informasi selengkapnya tentang pembuatan peran layanan, lihat Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend.

e. Pilih Berikutnya.

7. Pilih Simpan dan deploy.

Menggunakan kembali backend saat menghubungkan cabang ke aplikasi yang ada

Cara menggunakan kembali backend saat menghubungkan cabang ke aplikasi Amplify yang ada

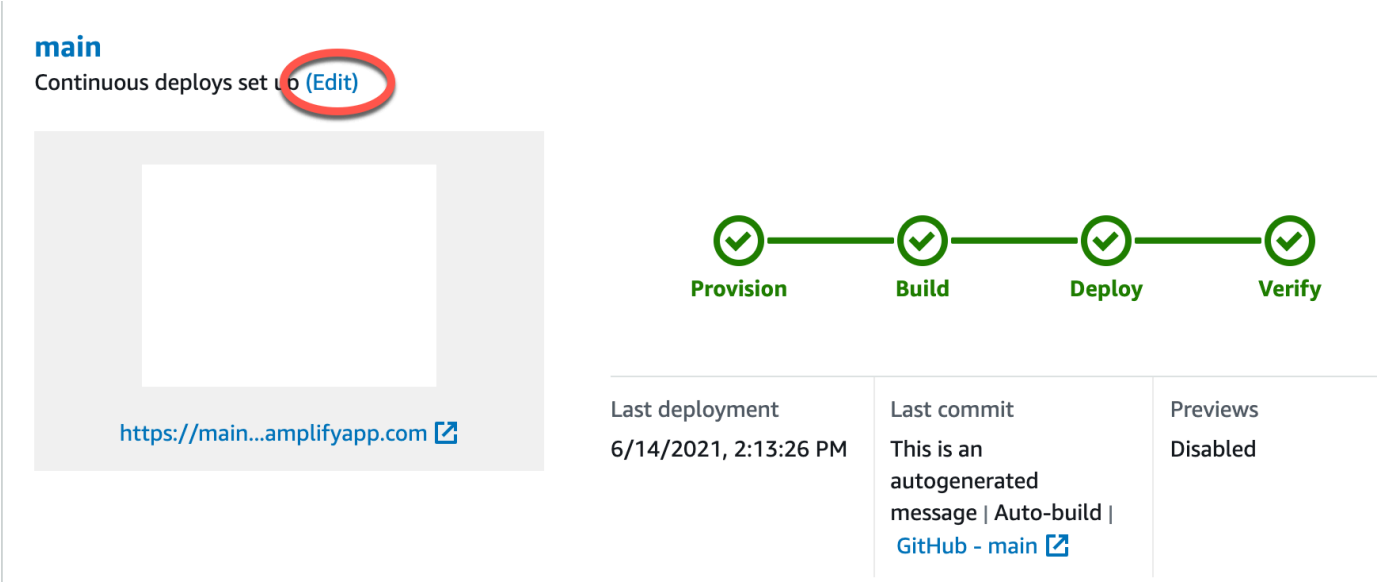
1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang akan dihubungkan ke cabang baru.
3. Di panel navigasi, pilih Pengaturan Aplikasi, Umum.
4. Di bagian Cabang, pilih Hubungkan cabang.
5. Di halaman Tambahkan cabang repositori, untuk Cabang, pilih cabang dari repositori Anda untuk dihubungkan.
6. Untuk Nama aplikasi, pilih aplikasi yang akan digunakan untuk menambahkan lingkungan backend. Anda dapat memilih aplikasi saat ini atau aplikasi lain di wilayah saat ini.
7. Untuk Lingkungan, pilih nama lingkungan backend yang akan ditambahkan. Anda dapat menggunakan lingkungan yang sudah ada atau membuat lingkungan baru.
8. Jika Anda perlu mengatur peran layanan guna memberi Amplify izin yang dibutuhkan untuk membuat perubahan pada backend aplikasi Anda, konsol akan meminta Anda untuk melakukan tugas ini. Untuk informasi selengkapnya tentang pembuatan peran layanan, lihat Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend.

9. Secara default, tumpukan penuh CI/CD is turned off. Turning off full-stack CI/CD menyebabkan aplikasi berjalan dalam mode tarik saja. Pada waktu build, Amplify secara otomatis akan menghasilkan file `aws-exports.js` saja, tanpa memodifikasi lingkungan backend Anda.
10. Pilih Berikutnya.
11. Pilih Simpan dan deploy.

Mengedit frontend yang ada agar mengarah ke backend berbeda

Cara mengedit aplikasi Amplify frontend agar mengarah ke backend berbeda

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi tempat backend akan diedit.
3. Pilih tab Lingkungan hosting.
4. Tentukan cabang yang akan diedit, lalu pilih Edit.



5. Pada Pilih lingkungan backend yang akan digunakan dengan halaman cabang ini, untuk nama Aplikasi, pilih aplikasi frontend yang ingin Anda edit untuk lingkungan backend. Anda dapat memilih aplikasi saat ini atau aplikasi lain di wilayah saat ini.
6. Untuk lingkungan Backend, pilih nama lingkungan backend yang akan ditambahkan.
7. Secara default, tumpukan penuh CI/CD is enabled. Uncheck this option to turn off full-stack CI/CD for this backend. Turning off full-stack CI/CD menyebabkan aplikasi berjalan dalam mode tarik saja. Pada waktu build, Amplify secara otomatis akan menghasilkan file `aws-exports.js` saja, tanpa memodifikasi lingkungan backend.

8. Pilih Simpan. Amplify menerapkan perubahan ini saat berikutnya Anda membuat aplikasi.

Membangun backend untuk aplikasi

Dengan AWS Amplify Anda dapat membangun aplikasi fullstack dengan data, otentikasi, penyimpanan, dan hosting frontend yang digunakan. AWS

AWS Amplify Gen 2 memperkenalkan pengalaman pengembang TypeScript berbasis kode pertama untuk mendefinisikan backend. Untuk mempelajari cara menggunakan Amplify Gen 2 untuk membangun dan menghubungkan backend ke aplikasi Anda, lihat Membangun & menghubungkan backend di dokumen Amplify.

Jika Anda mencari dokumentasi untuk membuat backend untuk aplikasi Gen 1, menggunakan CLI dan Amplify Studio, lihat Membangun & menghubungkan backend di dokumen Gen 1 Amplify.

Topik

Buat backend untuk aplikasi Gen 2

Untuk tutorial yang memandu Anda melalui langkah-langkah untuk membuat aplikasi fullstack Amplify Gen 2 dengan backend TypeScript berbasis, lihat Memulai di dokumen Amplify.

Buat backend untuk aplikasi Gen 1

Dalam tutorial ini, Anda akan mengatur alur kerja CI/CD fullstack dengan Amplify. Anda akan menerapkan aplikasi frontend ke Amplify Hosting. Kemudian Anda akan membuat backend menggunakan Amplify Studio. Terakhir, Anda akan menghubungkan backend cloud ke aplikasi frontend.

Prasyarat

Sebelum Anda memulai tutorial ini, selesaikan prasyarat berikut.

Mendaftar untuk Akun AWS

Jika Anda belum menjadi AWS pelanggan, Anda perlu membuat Akun AWS dengan mengikuti instruksi online. Mendaftar memungkinkan Anda mengakses Amplify dan AWS layanan lain yang dapat Anda gunakan dengan aplikasi Anda.

Buat repositori Git

Amplify mendukung GitHub, Bitbucket,, GitLab dan. AWS CodeCommit Dorong aplikasi Anda ke repositori Git Anda.

Instal Amplify Command Line Interface (CLI)

Untuk petunjuknya, lihat Menginstal Amplify CLI di Dokumentasi Amplify Framework.

Langkah 1: Menyebarkan frontend

Jika Anda memiliki aplikasi frontend yang ada di repositori git yang ingin Anda gunakan untuk contoh ini, Anda dapat melanjutkan ke instruksi untuk menerapkan aplikasi frontend.

Jika Anda perlu membuat aplikasi frontend baru untuk digunakan untuk contoh ini, Anda dapat mengikuti instruksi Create React App dalam dokumentasi Create React App.

Untuk menerapkan aplikasi frontend

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Di halaman Semua aplikasi, pilih Aplikasi baru, lalu Host aplikasi web di sudut kanan atas.
3. Pilih penyedia GitHub, Bitbucket GitLab, atau AWS CodeCommit repositori Anda, lalu pilih Lanjutkan.
4. Amplify mengotorisasi akses ke repositori git Anda. Untuk GitHub repositori, Amplify sekarang menggunakan fitur GitHub Apps untuk mengotorisasi akses Amplify.

   Untuk informasi selengkapnya tentang menginstal dan mengotorisasi GitHub Aplikasi, lihatMenyiapkan akses Amplify ke repositori GitHub.
5. Pada halaman Add repository branch lakukan hal berikut:

   a. Dalam daftar repositori yang baru diperbarui, pilih nama repositori yang akan dihubungkan.

   b. Dalam daftar Branch, pilih nama cabang repositori untuk terhubung.

   c. Pilih Berikutnya.
6. Pada halaman Konfigurasi pengaturan build, pilih Berikutnya.
7. Di halaman Tinjauan, pilih Simpan dan deploy. Saat penerapan selesai, Anda dapat melihat aplikasi di domain `amplifyapp.com` default.

> **Note**
>
> Untuk meningkatkan keamanan aplikasi Amplify Anda, domain amplifyapp.com terdaftar di Daftar Akhiran Publik (PSL). Untuk keamanan lebih lanjut, kami menyarankan Anda menggunakan cookie dengan `__Host-` awalan jika Anda perlu mengatur cookie sensitif di nama domain default untuk aplikasi Amplify Anda. Praktik ini akan membantu mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat halaman Set-Cookie di Jaringan Pengembang Mozilla.

Langkah 2: Buat backend

Sekarang setelah Anda menerapkan aplikasi frontend ke Amplify Hosting, Anda dapat membuat backend. Gunakan petunjuk berikut untuk membuat backend dengan database sederhana dan titik akhir GraphQL API.

Untuk membuat backend

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pada halaman Semua aplikasi, pilih aplikasi yang Anda buat di Langkah 1.
3. Di beranda aplikasi, pilih tab Lingkungan Backend, lalu pilih Memulai. Ini memulai proses penyiapan untuk lingkungan pementasan default.
4. Setelah penyiapan selesai, pilih Launch Studio untuk mengakses lingkungan backend pementasan di Amplify Studio.

Amplify Studio adalah antarmuka visual untuk membuat dan mengelola backend Anda dan mempercepat pengembangan UI frontend Anda. Untuk informasi selengkapnya tentang Amplify Studio, lihat dokumentasi Amplify Studio.

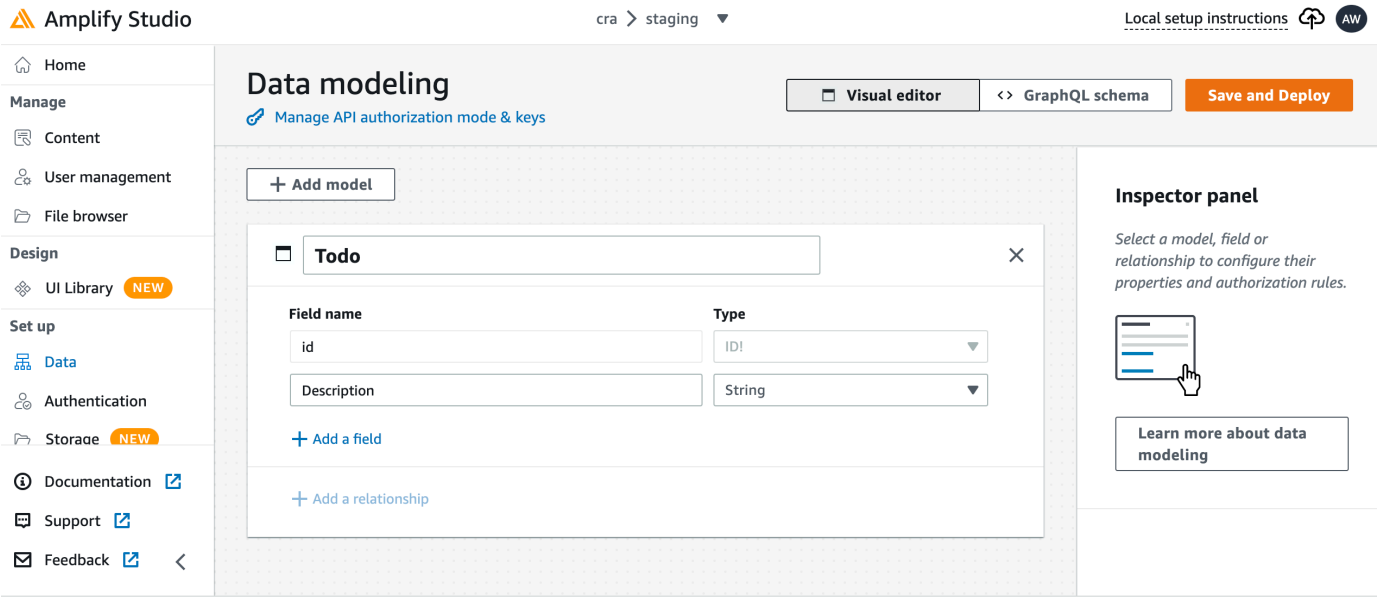
Gunakan petunjuk berikut untuk membuat database sederhana menggunakan antarmuka pembuat backend visual Amplify Studio.

Buat model data

1. Di halaman beranda untuk lingkungan pementasan aplikasi Anda, pilih Buat model data. Ini membuka perancang model data.
2. Pada halaman Pemodelan data, pilih Tambah model.

3. Untuk judul, masukkan**`Todo`**.
4. Pilih Tambahkan bidang.
5. Untuk nama Field, masukkan**`Description`**.

   Screenshot berikut adalah contoh bagaimana model data Anda akan terlihat di desainer.



6. Pilih Simpan dan Terapkan.
7. Kembali ke konsol Amplify Hosting dan penerapan lingkungan pementasan akan berlangsung.

Selama penerapan, Amplify Studio membuat semua resource yang AWS diperlukan di backend, termasuk AWS AppSync GraphQL API untuk mengakses data dan tabel Amazon DynamoDB untuk meng-host item Todo. Amplify digunakan AWS CloudFormation untuk menyebarkan backend Anda, yang memungkinkan Anda menyimpan definisi backend Anda sebagai. infrastructure-as-code

Langkah 3: Hubungkan backend ke frontend

Sekarang Anda telah menerapkan frontend dan membuat backend cloud yang berisi model data, Anda harus menghubungkannya. Gunakan petunjuk berikut untuk menarik definisi backend Anda ke project aplikasi lokal Anda dengan Amplify CLI.

Untuk menghubungkan backend cloud ke frontend lokal

1. Buka jendela terminal dan arahkan ke direktori root proyek lokal Anda.
2. Jalankan perintah berikut di jendela terminal, ganti teks merah dengan ID aplikasi unik dan nama lingkungan backend untuk proyek Anda.

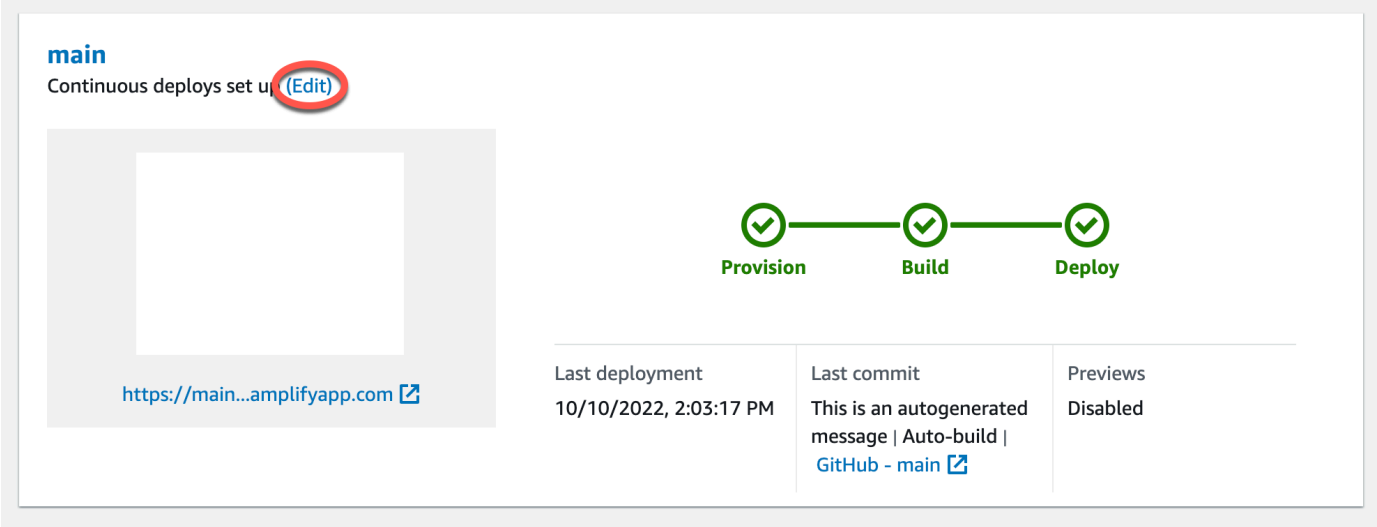
```
amplify pull --appId abcd1234 --envName staging
```

3. Ikuti petunjuk di jendela terminal untuk menyelesaikan pengaturan proyek.

Sekarang Anda dapat mengonfigurasi proses pembuatan untuk menambahkan backend ke alur kerja penerapan berkelanjutan. Gunakan petunjuk berikut untuk menghubungkan cabang frontend dengan backend di konsol Amplify Hosting.

Untuk menghubungkan cabang aplikasi frontend dan backend cloud

1. Di beranda aplikasi, pilih tab Lingkungan hosting.
2. Temukan cabang utama dan pilih Edit.



3. Di jendela Edit backend target, untuk Lingkungan, pilih nama backend yang akan dihubungkan. Dalam contoh ini, pilih backend pementasan yang Anda buat di Langkah 2.

   Secara default, tumpukan penuh CI/CD is enabled. Uncheck this option to turn off full-stack CI/CD for this backend. Turning off full-stack CI/CD menyebabkan aplikasi berjalan dalam mode tarik saja. Pada waktu build, Amplify secara otomatis akan menghasilkan file `aws-exports.js` saja, tanpa memodifikasi lingkungan backend Anda.
4. Selanjutnya, Anda harus menyiapkan peran layanan untuk memberikan Amplify izin yang diperlukan untuk membuat perubahan pada backend aplikasi Anda. Anda dapat menggunakan peran layanan yang ada atau membuat yang baru. Untuk petunjuk, silakan lihat Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend.
5. Setelah menambahkan peran layanan, kembali ke jendela backend Edit target dan pilih Simpan.

6. Untuk menyelesaikan menghubungkan backend pementasan ke cabang utama aplikasi frontend, lakukan build baru proyek Anda.

   Lakukan salah satu hal berikut ini:

   - Dari repositori git Anda, tekan beberapa kode untuk memulai build di konsol Amplify.
   - Di konsol Amplify, navigasikan ke halaman detail build aplikasi dan pilih Redeploy versi ini.

Langkah selanjutnya

Siapkan penerapan cabang fitur

Ikuti rekomendasi alur kerja kami untuk mengatur deployment cabang fitur dengan beberapa lingkungan backend.

Buat UI frontend di Amplify Studio

Gunakan Studio untuk membangun UI frontend Anda dengan satu set komponen ready-to-use UI, lalu sambungkan ke backend aplikasi Anda. Untuk informasi dan tutorial selengkapnya, lihat panduan pengguna untuk Amplify Studio di Dokumentasi Amplify Framework.

Fitur penyebaran lanjutan

Bab ini mencakup fitur penerapan lanjutan yang menyempurnakan alur kerja Amplify Hosting Anda. Fitur-fitur ini memberikan kontrol dan kemampuan tambahan untuk membantu tim mengelola penerapan secara lebih efektif, memastikan kualitas kode, dan menjaga keamanan selama siklus pengembangan.

Pelajari cara melindungi cabang fitur Anda dengan autentikasi kata sandi untuk membatasi akses ke fitur yang belum dirilis. Aktifkan pratinjau web untuk permintaan tarik untuk meninjau perubahan pada pratinjau unik URLs sebelum menggabungkan kode ke cabang produksi. Siapkan end-to-end pengujian menggunakan kerangka Cypress untuk menangkap regresi sebelum mendorong kode ke produksi. Meskipun fitur tombol Deploy to Amplify tidak lagi tersedia, Anda masih dapat dengan mudah menerapkan aplikasi langsung dari repositori menggunakan Amplify Hosting.

Topik

Membatasi akses ke cabang aplikasi Amplify

Jika sedang menyelesaikan fitur yang belum dirilis, Anda dapat menggunakan kata sandi untuk melindungi cabang fitur untuk membatasi akses ke pengguna tertentu. Ketika kontrol akses diatur pada cabang, pengguna akan diminta untuk nama pengguna dan kata sandi ketika mereka mencoba untuk mengakses URL untuk cabang.

Anda dapat mengatur kata sandi yang berlaku untuk cabang individu atau secara global ke semua cabang yang terhubung. Ketika kontrol akses diaktifkan di tingkat cabang dan global, kata sandi tingkat cabang lebih diutamakan daripada kata sandi tingkat global (aplikasi).

Amplify throttle permintaan gagal yang mencoba mengakses sumber daya yang dilindungi kata sandi. Perilaku ini melindungi aplikasi terhadap serangan kamus atau upaya lain untuk membaca data di balik kontrol akses.

Gunakan prosedur berikut untuk menyetel kata sandi guna membatasi akses ke cabang aplikasi Amplify.

Untuk mengatur kata sandi di cabang fitur

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi tempat kata sandi cabang fitur akan diatur.
3. Di panel navigasi, pilih Hosting, lalu pilih Kontrol akses.
4. Di bagian Pengaturan kontrol akses, pilih Kelola akses.
5. Pada halaman Mengelola kontrol akses, lakukan salah satu hal berikut.
   - Untuk mengatur nama pengguna dan kata sandi yang diterapkan ke semua cabang terhubung
     - Aktifkan Kelola akses untuk semua cabang. Misalnya, jika menghubungkan cabang main, dev, dan feature, Anda dapat menerapkan nama pengguna dan kata sandi yang sama untuk semua cabang.
   - Untuk mengatur nama pengguna dan kata sandi yang diterapkan ke cabang individual
     a. Matikan Kelola akses untuk semua cabang.
     b. Temukan cabang yang ingin Anda kelola. Untuk pengaturan Access pilih Restricted- password required.
     c. Untuk Nama Pengguna, masukkan nama pengguna.
     d. Untuk Kata Sandi, masukkan kata sandi.
   - Pilih Simpan.
6. Jika Anda mengelola kontrol akses untuk aplikasi yang dirender sisi server (SSR), terapkan ulang aplikasi dengan melakukan build baru dari repositori Git Anda. Langkah ini diperlukan untuk mengaktifkan Amplify untuk menerapkan pengaturan kontrol akses Anda.

Pratinjau web untuk permintaan tarik

Dengan pratinjau web, tim pengembangan dan penjaminan kualitas (QA) dapat melihat pratinjau perubahan dari permintaan tarik (PRs) sebelum kode digabungkan ke cabang produksi atau integrasi. Permintaan tarik membantu Anda memberi tahu orang lain tentang perubahan yang telah Anda dorong ke cabang di repositori. Setelah permintaan tarik dibuka, Anda dapat mendiskusikan dan meninjau kemungkinan perubahan dengan kolaborator dan menambahkan penerapan tindak lanjut sebelum perubahan digabungkan ke cabang dasar.

Pratinjau web men-deploy setiap permintaan tarik yang dibuat ke repositori Anda ke URL pratinjau unik yang berbeda sepenuhnya dari URL yang digunakan situs utama Anda. Untuk aplikasi dengan lingkungan backend yang disediakan menggunakan CLI Amplify atau Amplify Studio, setiap permintaan tarik (hanya repositori Git privat) akan menghapus sebuah backend sementara yang dihapus ketika PR ditutup.

Saat pratinjau web diaktifkan untuk aplikasi Anda, setiap PR diperhitungkan dalam kuota Amplify 50 cabang per aplikasi. Untuk menghindari melebihi kuota ini, pastikan untuk menutup kuota Anda. PRs Untuk informasi selengkapnya tentang kuota, lihatKuota layanan Amplify Hosting.

> **Note**
>
> Saat ini, variabel `AWS_PULL_REQUEST_ID` lingkungan tidak tersedia saat digunakan AWS CodeCommit sebagai penyedia repositori Anda.

Keamanan pratinjau web

Untuk tujuan keamanan, Anda dapat mengaktifkan pratinjau web di semua aplikasi dengan repositori pribadi, tetapi tidak pada semua aplikasi dengan repositori publik. Jika repositori Git Anda bersifat publik, Anda dapat mengatur pratinjau hanya untuk aplikasi yang tidak memerlukan peran layanan IAM. Misalnya, aplikasi dengan backend dan aplikasi yang digunakan ke platform `WEB_COMPUTE` hosting memerlukan peran layanan IAM. Oleh karena itu, Anda tidak dapat mengaktifkan pratinjau web untuk jenis aplikasi ini jika repositori mereka bersifat publik. Amplify memberlakukan pembatasan ini untuk mencegah pihak ketiga mengirimkan kode arbitrer yang akan berjalan menggunakan izin peran IAM aplikasi Anda.

Saat pratinjau web diaktifkan untuk aplikasi di repositori publik, dengan peran Komputasi SSR, Anda perlu mengelola cabang mana yang dapat mengakses peran dengan hati-hati. Sebaiknya jangan menggunakan peran tingkat aplikasi. Sebagai gantinya, Anda harus melampirkan peran Compute di tingkat cabang. Ini memungkinkan Anda untuk memberikan izin hanya ke cabang yang memerlukan akses ke sumber daya tertentu. Untuk informasi selengkapnya, lihat Menambahkan peran SSR Compute untuk memungkinkan akses ke sumber daya AWS.

Aktifkan pratinjau web untuk permintaan tarik

Untuk aplikasi yang disimpan dalam GitHub repo, pratinjau web menggunakan GitHub Aplikasi Amplify untuk akses repo. Jika Anda mengaktifkan pratinjau web di aplikasi Amplify yang sudah

ada yang sebelumnya Anda gunakan dari GitHub repo OAuth untuk akses, Anda harus terlebih dahulu memigrasikan aplikasi untuk menggunakan Aplikasi Amplify. GitHub Untuk petunjuk migrasi, lihatMemigrasi aplikasi yang ada ke OAuth Aplikasi Amplify GitHub .

Cara mengaktifkan pratinjau web untuk permintaan tarik

1. Pilih Hosting, lalu Pratinjau.

   > **Note**
   >
   > Pratinjau dapat dilihat di menu Pengaturan aplikasi hanya jika aplikasi diatur untuk deployment kontinu dan terhubung ke repositori git. Untuk langkah-langkah seputar jenis deployment ini, lihat Memulai dengan kode yang ada.

2. Hanya untuk GitHub repositori, lakukan hal berikut untuk menginstal dan mengotorisasi Aplikasi Amplify GitHub di akun Anda:

   a. Di jendela Instal GitHub Aplikasi untuk mengaktifkan pratinjau, pilih Instal GitHub aplikasi.
   b. Pilih GitHub akun tempat Anda ingin mengonfigurasi Aplikasi Amplify GitHub.
   c. Halaman terbuka di GitHub.com untuk mengonfigurasi izin repositori untuk akun Anda.
   d. Lakukan salah satu tindakan berikut:
      - Untuk menerapkan instalasi ke semua repositori, pilih Semua repositori.
      - Untuk membatasi instalasi ke repositori tertentu yang Anda pilih, pilih Hanya pilih repositori. Pastikan untuk menyertakan repo untuk aplikasi yang Anda aktifkan pratinjau web di repositori yang Anda pilih.
   e. Pilih Simpan.

3. Setelah Anda mengaktifkan pratinjau untuk repo, kembali ke konsol Amplify untuk mengaktifkan pratinjau untuk cabang tertentu. Pada halaman Pratinjau, pilih cabang dari daftar dan pilih Edit pengaturan.
4. Pada halaman Kelola pengaturan pratinjau, aktifkan Pratinjau permintaan tarik. Kemudian pilih Konfirmasi.
5. Untuk aplikasi fullstack lakukan salah satu hal berikut:
   - Pilih, Buat lingkungan backend baru untuk setiap Permintaan Tarik. Opsi ini memungkinkan Anda untuk menguji perubahan tanpa memengaruhi produksi.
   - Pilih Arahkan semua Permintaan Tarik untuk cabang ini ke lingkungan yang ada.
6. Pilih Konfirmasi.

Selanjutnya, jika Anda mengirimkan permintaan tarik untuk cabang, Amplify akan membangun dan men-deploy PR Anda ke URL pratinjau. Setelah permintaan tarik ditutup, URL pratinjau akan dihapus, dan setiap lingkungan backend sementara yang tertaut ke permintaan tarik akan dihapus. Hanya untuk GitHub repositori, Anda dapat mengakses pratinjau URL langsung dari permintaan tarik di akun Anda GitHub .

Akses pratinjau web dengan subdomain

Pratinjau web untuk permintaan tarik dapat diakses dengan subdomain untuk aplikasi Amplify yang terhubung ke domain kustom yang dikelola oleh Amazon Route 53. Ketika permintaan tarik ditutup, cabang dan subdomain yang terkait dengan permintaan tarik akan dihapus secara otomatis. Ini adalah perilaku default untuk pratinjau web setelah Anda mengatur deployment cabang fitur berbasis pola untuk aplikasi. Untuk langkah-langkah seputar pengaturan subdomain otomatis, lihat Menyiapkan subdomain otomatis untuk domain kustom Amazon Route 53.

Menyiapkan tes end-to-end Cypress untuk aplikasi Amplify Anda

Anda dapat menjalankan pengujian end-to-end (E2E) di tahap pengujian aplikasi Amplify untuk menangkap regresi sebelum mendorong kode ke produksi. Tahap pengujian dapat dikonfigurasi dalam spesifikasi build YAMP. Saat ini, Anda hanya dapat menjalankan kerangka pengujian Cypress selama pembuatan.

Cypress adalah kerangka pengujian JavaScript berbasis yang memungkinkan Anda untuk menjalankan pengujian E2E di peramban. Untuk tutorial yang menunjukkan cara mengatur tes E2E, lihat posting blog Menjalankan tes end-to-end Cypress untuk penerapan CI/CD fullstack Anda dengan Amplify.

Menambahkan tes Cypress ke aplikasi Amplify yang ada

Anda dapat menambahkan pengujian Cypress ke aplikasi yang ada dengan memperbarui pengaturan build aplikasi di konsol Amplify. YAM spesifikasi build berisi serangkaian perintah build dan pengaturan terkait yang digunakan Amplify untuk menjalankan build Anda. Gunakan `test` langkah untuk menjalankan perintah pengujian pada waktu build. Untuk pengujian E2E, Amplify Hosting menawarkan integrasi lebih dalam dengan Cypress yang memungkinkan Anda untuk membuat laporan UI untuk pengujian Anda.

Daftar berikut menjelaskan pengaturan pengujian dan bagaimana penggunaannya.

Tes

Menginstal dependensi yang diperlukan untuk menjalankan pengujian Cypress. Amplify Hosting menggunakan mochawesome untuk membuat laporan untuk melihat hasil pengujian dan wait-on untuk menyiapkan server localhost selama build.

pengujian

Menjalankan perintah cypress untuk melakukan pengujian menggunakan mochawesome.

PostTest

Laporan mochawesome dihasilkan dari JSON output. Perhatikan bahwa jika Anda menggunakan Yarn, Anda harus menjalankan perintah ini dalam mode diam untuk menghasilkan laporan mochawesome. Untuk Yarn, Anda dapat menggunakan perintah berikut.

```
yarn run --silent mochawesome-merge cypress/report/mochawesome-report/
mochawesome*.json > cypress/report/mochawesome.json
```

artifacts>baseDirectory

Direktori tempat pengujian dijalankan.

artefak configFilePath

Data laporan pengujian yang dihasilkan.

artefak

Artefak yang dihasilkan (tangkapan layar dan video) yang tersedia untuk diunduh.

Contoh kutipan berikut dari `amplify.yml` file spesifikasi build menunjukkan cara menambahkan pengujian Cypress ke aplikasi Anda.

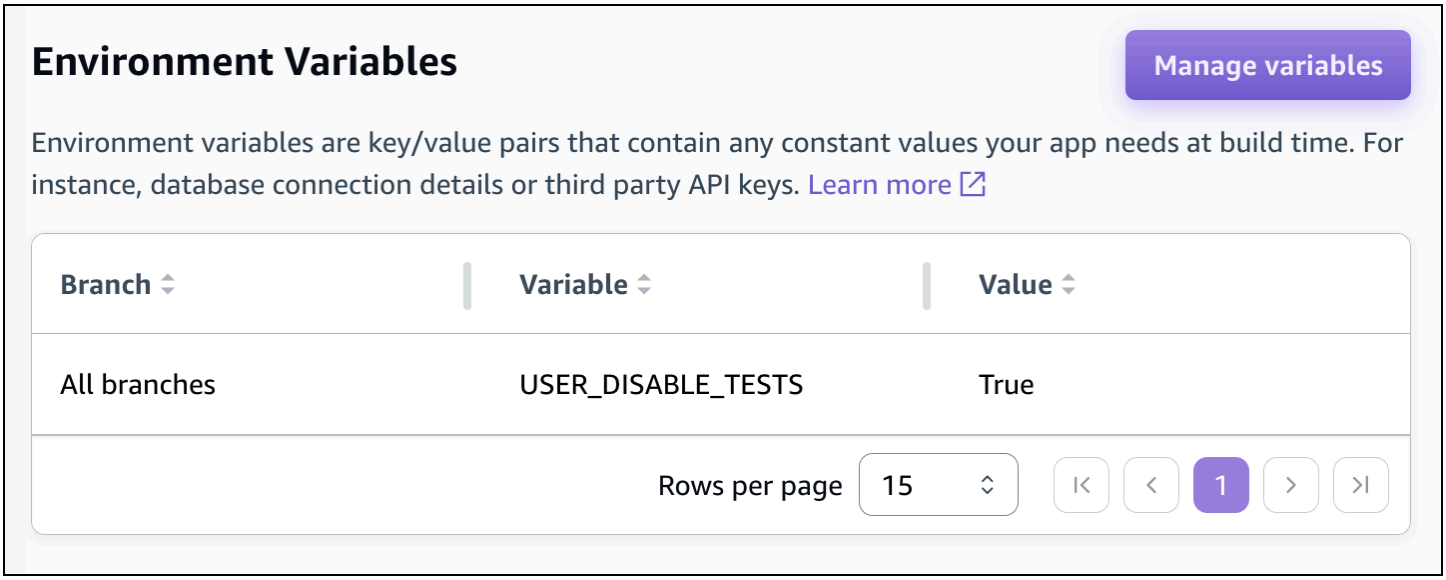
```
test:
  phases:
    preTest:
      commands:
        - npm ci
        - npm install -g pm2
        - npm install -g wait-on
```

```
        - npm install mocha mochawesome mochawesome-merge mochawesome-report-generator
        - pm2 start npm -- start
        - wait-on http://localhost:3000
    test:
      commands:
        - 'npx cypress run --reporter mochawesome --reporter-options
 "reportDir=cypress/report/mochawesome-
report,overwrite=false,html=false,json=true,timestamp=mmddyyyy_HHMMss"'
    postTest:
      commands:
        - npx mochawesome-merge cypress/report/mochawesome-report/mochawesome*.json >
 cypress/report/mochawesome.json
        - pm2 kill
  artifacts:
    baseDirectory: cypress
    configFilePath: '**/mochawesome.json'
    files:
      - '**/*.png'
      - '**/*.mp4'
```

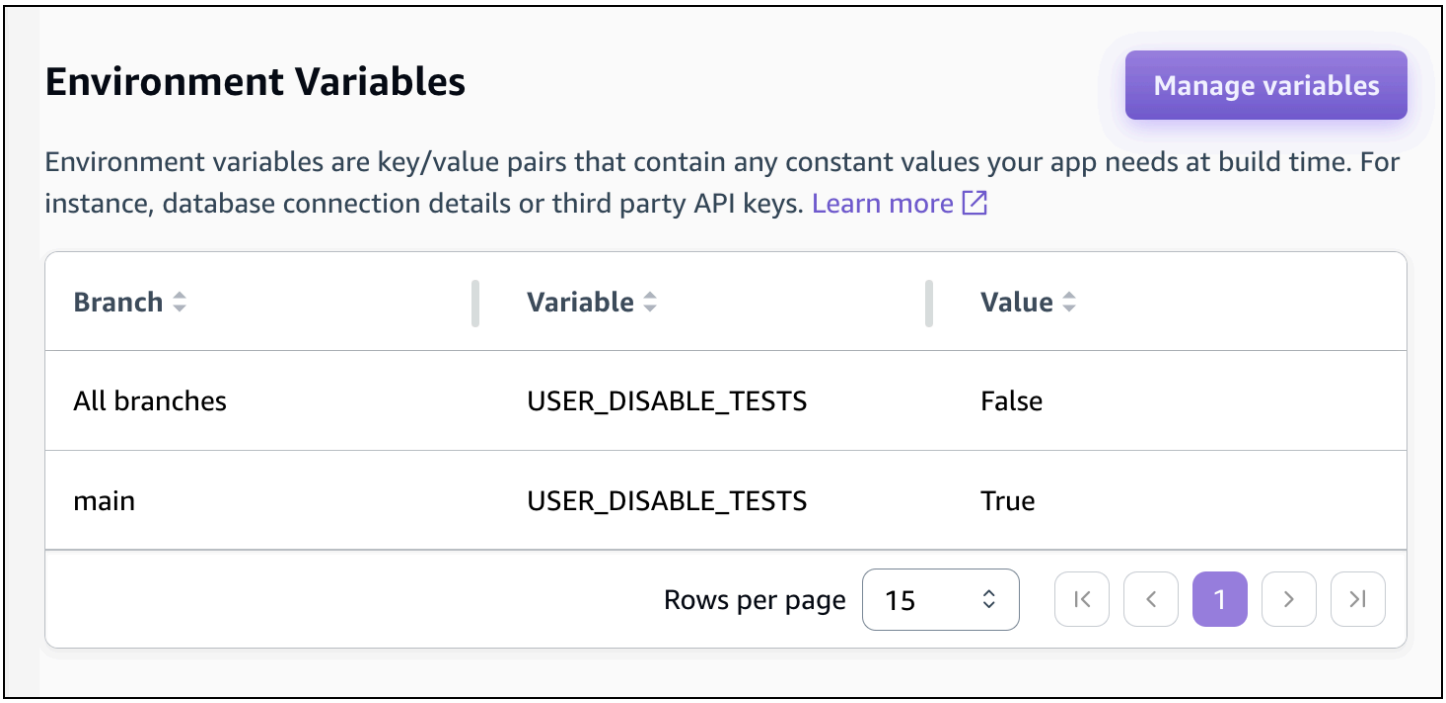
Mematikan pengujian untuk aplikasi atau cabang Amplify

Setelah konfigurasi pengujian ditambahkan ke pengaturan `amplify.yml` build, `test` langkah berjalan untuk setiap build di setiap cabang. Jika ingin menonaktifkan pengujian yang berjalan secara global, atau hanya menjalankan pengujian untuk cabang tertentu, Anda dapat menggunakan variabel USER_DISABLE_TESTS lingkungan tanpa memodifikasi pengaturan build.

Untuk menonaktifkan pengujian secara global untuk semua cabang, tambahkan variabel USER_DISABLE_TESTS lingkungan dengan nilai `true` untuk semua cabang. Tangkapan layar berikut, menunjukkan bagian Variabel lingkungan di konsol Amplify dengan pengujian dinonaktifkan untuk semua cabang.



Untuk menonaktifkan pengujian untuk cabang tertentu, tambahkan variabel USER_DISABLE_TESTS lingkungan dengan nilai `false` untuk semua cabang, kemudian tambahkan override untuk setiap cabang yang akan dinonaktifkan dengan nilai`true`. Pada tangkapan layar berikut, pengujian dinonaktifkan di cabang utama, dan diaktifkan untuk setiap cabang lain.



Menonaktifkan pengujian dengan variabel ini akan menyebabkan langkah pengujian dilewati sepenuhnya selama build. Untuk mengaktifkan kembali pengujian, tetapkan nilai ini ke`false`, atau hapus variabel lingkungan.

Menggunakan tombol Deploy to Amplify untuk berbagi proyek GitHub

> **Important**
>
> Penerapan sekali klik menggunakan tombol Deploy to Amplify Hosting tidak lagi tersedia. Untuk menyebarkan dari repositori, buat aplikasi baru di Amplify Hosting. Untuk petunjuk, lihat Memulai dengan menerapkan aplikasi ke Amplify Hosting.

Dengan tombol Deploy ke Amplify, Anda dapat GitHub berbagi proyek secara publik atau dalam tim Anda. Berikut gambar tombol ini:

Menambahkan tombol Deploy to Amplify Hosting ke repositori atau blog

Tambahkan tombol ke file GitHub README.md, postingan blog, atau halaman markup lain yang menggunakan HTML. Tombol terdiri dari dua komponen berikut:

1. Gambar SVG yang terletak di URL `https://oneclick.amplifyapp.com/button.svg`
2. URL konsol Amplify dengan tautan ke repositori Anda GitHub . Anda dapat menyalin URL repositori Anda, sepertihttps://github.com/username/repository, atau Anda dapat memberikan tautan dalam ke folder tertentu, seperti. `https://github.com/username/repository/tree/branchname/folder` Amplify Hosting akan men-deploy cabang default di repositori Anda. Cabang tambahan dapat dihubungkan setelah aplikasi terhubung.

Gunakan contoh berikut untuk menambahkan tombol ke file penurunan harga, seperti GitHub README.md Anda. Ganti `https://github.com/username/repository` dengan URL ke repositori Anda.

```
[![amplifybutton](https://oneclick.amplifyapp.com/button.svg)](https://
console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/username/
repository)
```

Gunakan contoh berikut untuk menambahkan tombol ke dokumen HTML. Ganti `https://github.com/username/repository` dengan URL ke repositori Anda.

```
<a href="https://console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/
username/repository">
    <img src="https://oneclick.amplifyapp.com/button.svg" alt="Deploy to Amplify
 Hosting">
</a>
```

Menyiapkan pengalihan dan penulisan ulang untuk aplikasi Amplify

Pengalihan memungkinkan server web untuk mengubah rute navigasi dari satu URL ke URL lainnya. Alasan umum untuk menggunakan pengalihan termasuk untuk menyesuaikan tampilan URL, untuk menghindari tautan rusak, untuk memindahkan lokasi hosting aplikasi atau situs tanpa mengubah alamatnya, dan untuk mengubah URL yang diminta ke formulir yang dibutuhkan oleh aplikasi web.

Memahami pengalihan yang didukung Amplify

Amplify mendukung jenis pengalihan berikut di konsol.

Pengalihan permanen (301)

Pengalihan 301 ditujukan untuk perubahan permanen pada tujuan alamat web. Riwayat peringkat mesin pencari alamat asli berlaku untuk alamat tujuan baru. Pengalihan terjadi di sisi klien sehingga bilah navigasi peramban menunjukkan alamat tujuan setelah pengalihan.

Alasan umum penggunaan pengalihan 301 meliputi:

- Untuk menghindari tautan rusak ketika alamat halaman berubah.
- Untuk menghindari tautan rusak ketika pengguna membuat kesalahan ketik yang dapat diprediksi di alamat.

Pengalihan sementara (302)

Pengalihan 302 ditujukan untuk perubahan sementara pada tujuan alamat web. Riwayat peringkat mesin pencari alamat asli tidak berlaku untuk alamat tujuan baru. Pengalihan terjadi di sisi klien sehingga bilah navigasi peramban menunjukkan alamat tujuan setelah pengalihan.

Alasan umum penggunaan pengalihan 302 meliputi:

- Untuk memberikan tujuan detour saat perbaikan dilakukan ke alamat asli.
- Untuk menyediakan halaman uji untuk A/B perbandingan antarmuka pengguna.

> **Note**
>
> Jika aplikasi Anda menampilkan respons 302 yang tidak terduga, kesalahan kemungkinan disebabkan oleh perubahan yang Anda buat pada konfigurasi pengalihan dan header khusus aplikasi Anda. Untuk mengatasi masalah ini, verifikasi bahwa header kustom Anda valid, lalu aktifkan kembali aturan penulisan ulang 404 default untuk aplikasi Anda.

Menulis ulang (200)

Pengalihan 200 (penulisan ulang) ditujukan untuk menampilkan konten dari alamat tujuan seolah-olah ditampilkan dari alamat asli. Riwayat peringkat mesin pencari terus berlaku untuk alamat asli. Pengalihan terjadi di sisi server sehingga bilah navigasi peramban menunjukkan alamat asli setelah pengalihan. Alasan umum penggunaan pengalihan 200 meliputi:

- Untuk mengalihkan seluruh situs ke lokasi hosting baru tanpa mengubah alamat situs.
- Untuk mengalihkan semua lalu lintas ke aplikasi web halaman tunggal (SPA) ke halaman index.html agar ditangani oleh fungsi router sisi klien.

Tidak Ditemukan (404)

Pengalihan 404 terjadi ketika permintaan mengarah ke alamat yang tidak ada. Halaman tujuan 404 ditampilkan, alih-alih halaman yang diminta. Alasan umum terjadinya pengalihan 404 meliputi:

- Untuk menghindari pesan tautan rusak ketika pengguna memasukkan URL buruk.
- Untuk mengarahkan permintaan ke halaman yang tidak ada di aplikasi web ke halaman index.html agar ditangani oleh fungsi router sisi klien.

Memahami urutan pengalihan

Pengalihan diterapkan dari bagian atas daftar ke bawah. Pastikan bahwa urutan yang dibuat memberikan efek yang diinginkan. Sebagai contoh, urutan pengalihan berikut menyebabkan semua permintaan untuk path tertentu di /docs/ melakukan pengalihan ke path yang sama di /documents/, kecuali /docs/specific-filename.html yang melakukan pengalihan ke /documents/different-filename.html:

```
/docs/specific-filename.html /documents/different-filename.html 301
```

```
/docs/<*> /documents/<*>
```

Urutan pengalihan berikut mengabaikan pengalihan specific-filename.html ke different-filename.html:

```
/docs/<*> /documents/<*>
/docs/specific-filename.html /documents/different-filename.html 301
```

Memahami bagaimana Amplify meneruskan parameter kueri

Anda dapat menggunakan parameter kueri untuk kontrol lebih besar atas kecocokan URL Anda. Amplify meneruskan semua parameter kueri ke jalur tujuan untuk pengalihan 301 dan 302, dengan pengecualian berikut:

- Jika alamat asli menyertakan string kueri yang disetel ke nilai tertentu, Amplify tidak meneruskan parameter kueri. Dalam hal ini, pengalihan hanya berlaku untuk permintaan ke URL tujuan dengan nilai kueri yang ditentukan.
- Jika alamat tujuan untuk aturan pencocokan memiliki parameter kueri, parameter kueri tidak diteruskan. Misalnya, jika alamat tujuan untuk pengalihan adalah`https://example-target.com?q=someParam`, parameter kueri tidak akan diteruskan.

Membuat dan mengedit pengalihan di konsol Amplify

Anda dapat membuat dan mengedit pengalihan untuk aplikasi di konsol Amplify. Sebelum memulai, Anda memerlukan informasi berikut tentang bagian-bagian pengalihan.

Alamat asli

Alamat yang diminta pengguna.

Alamat tujuan

Alamat yang benar-benar menyajikan konten yang dilihat pengguna.

Jenis pengalihan

Jenis termasuk pengalihan permanen (301), pengalihan sementara (302), penulisan ulang (200), atau tidak ditemukan (404).

Kode negara dua huruf (opsional)

Nilai yang dapat Anda sertakan untuk mengelompokkan pengalaman pengguna aplikasi berdasarkan wilayah geografis.

Untuk membuat pengalihan di konsol Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang ingin Anda buat pengalihan.
3. Di panel navigasi, pilih Hosting, lalu pilih Menulis ulang dan mengarahkan ulang.
4. Pada halaman Menulis ulang dan mengarahkan ulang, pilih Kelola pengalihan.
5. Tambahkan atau perbarui pengalihan secara manual di Menulis ulang dan mengarahkan editor JSON.
   a. Untuk`source`, tentukan alamat asli yang diminta pengguna.
   b. Untuk`status`, tentukan jenis pengalihan.
   c. Untuk`target`, tentukan alamat tujuan yang merender konten ke pengguna.
   d. (Opsional) Untuk`condition`, masukkan kondisi kode negara dua huruf.
6. Pilih Simpan.

Mengalihkan dan menulis ulang contoh referensi

Bagian ini memberikan contoh untuk berbagai skenario pengalihan umum. Anda dapat menggunakan contoh ini untuk memahami sintaks JSON untuk membuat pengalihan dan penulisan ulang Anda sendiri di editor JSON konsol Amplify.

> **Note**
>
> Pencocokan domain alamat asli tidak peka huruf besar/kecil.

Topik

- Pengalihan dan penulisan ulang sederhana
- Pengalihan untuk aplikasi web halaman tunggal (SPA)
- Penulisan ulang proksi balik

- Trailing garis miring dan bersih URLs
- Placeholder
- String kueri dan parameter path
- Pengalihan berbasis wilayah
- Menggunakan ekspresi wildcard dalam pengalihan dan penulisan ulang

Pengalihan dan penulisan ulang sederhana

Anda dapat menggunakan contoh berikut untuk mengalihkan halaman tertentu secara permanen ke alamat baru.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/original.html`	`/destination.html`	`permanent redirect (301)`	

Format JSON

```
[
  {
    "source": "/original.html",
    "status": "301",
    "target": "/destination.html",
    "condition": null
  }
]
```

Anda dapat menggunakan contoh berikut untuk mengalihkan jalur apa pun di bawah folder ke jalur yang sama di bawah folder yang berbeda.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

Format JSON

```
[
  {
    "source": "/docs/<*>",
    "status": "301",
    "target": "/documents/<*>",
    "condition": null
  }
]
```

Anda dapat menggunakan contoh berikut untuk mengarahkan semua lalu lintas ke index.html sebagai penulisan ulang. Dalam skenario ini, penulisan ulang menampilkan kepada pengguna bahwa pengguna telah sampai di alamat asli.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
/<*>	/index.html	rewrite (200)	

Format JSON

```
[
  {
    "source": "/<*>",
    "status": "200",
    "target": "/index.html",
    "condition": null
  }
]
```

Anda dapat menggunakan contoh berikut untuk menggunakan penulisan ulang untuk mengubah subdomain yang muncul kepada pengguna.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
https://mydomain.com	https://www.mydomain.com	rewrite (200)	

Format JSON

```
[
  {
    "source": "https://mydomain.com",
    "status": "200", "target": "https://www.mydomain.com",
    "condition": null
  }
]
```

Anda dapat menggunakan contoh berikut untuk mengarahkan ke domain yang berbeda dengan awalan jalur.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`https://mydomain.com`	`https://www.mydomain.com/documents`	`temporary redirect (302)`	

Format JSON

```
[
  {
    "source": "https://mydomain.com",
    "status": "302",
    "target": "https://www.mydomain.com/documents/",
    "condition": null
  }
]
```

Anda dapat menggunakan contoh berikut untuk mengalihkan jalur di bawah folder yang tidak dapat ditemukan ke halaman 404 kustom.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/<*>`	`/404.html`	`not found (404)`	

Format JSON

```
[
  {
    "source": "/<*>",
    "status": "404",
    "target": "/404.html",
    "condition": null
  }
]
```

Pengalihan untuk aplikasi web halaman tunggal (SPA)

Sebagian besar kerangka kerja SPA mendukung HTML5 History.pushState () untuk mengubah lokasi browser tanpa memulai permintaan server. Ini dapat digunakan untuk pengguna yang memulai perjalanan dari root (atau /index.html), tetapi tidak dapat digunakan untuk pengguna yang menavigasi langsung ke halaman lain.

Contoh berikut menggunakan ekspresi reguler untuk mengatur penulisan ulang 200 untuk semua file ke index.html, kecuali untuk ekstensi file yang ditentukan dalam ekspresi reguler.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`</^[^.]+$\|\.(?!(css\|gif\|ico\|jpg\|js\|png\|txt\|svg\|woff\|woff2\|ttf\|map\|json\|webp)$)([^.]+$)/>`	`/index.html`	`200`	

Format JSON

```
[
  {
    "source": "</^[^.]+$|\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|
webp)$)([^.]+$)/>",
    "status": "200",
    "target": "/index.html",
    "condition": null
```

```
  }
]
```

Penulisan ulang proksi balik

Contoh berikut menggunakan penulisan ulang ke konten proxy dari lokasi lain sehingga tampak bagi pengguna bahwa domain tidak berubah. HTTPS adalah satu-satunya protokol yang didukung untuk reverse proxy.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/images/<*>`	`https://images.otherdomain.com/<*>`	`rewrite (200)`	

Format JSON

```
[
  {
    "source": "/images/<*>",
    "status": "200",
    "target": "https://images.otherdomain.com/<*>",
    "condition": null
  }
]
```

Trailing garis miring dan bersih URLs

Untuk membuat struktur URL bersih, misalnya about, alih-alih about.html, generator situs statis, seperti Hugo, membuat direktori untuk halaman dengan index.html (/about/index.html). Amplify secara otomatis menciptakan clean URLs dengan menambahkan garis miring saat diperlukan. Tabel berikut menampilkan berbagai skenario:

Input pengguna di peramban	URL di bilah alamat	Dokumen ditampilkan
`/about`	`/about`	`/about.html`

Input pengguna di peramban	URL di bilah alamat	Dokumen ditampilkan
`/about (when about.html returns 404)`	`/about/`	`/about/index.html`
`/about/`	`/about/`	`/about/index.html`

Placeholder

Anda dapat menggunakan contoh berikut untuk mengarahkan jalur dalam struktur folder ke struktur yang cocok di folder lain.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/docs/<year>/<month>/<date>/<itemid>`	`/documents/<year>/<month>/<date>/<itemid>`	`permanent redirect (301)`	

Format JSON

```
[
  {
    "source":  "/docs/<year>/<month>/<date>/<itemid>",
    "status": "301",
    "target": "/documents/<year>/<month>/<date>/<itemid>",
               "condition": null
   }
]
```

String kueri dan parameter path

> **Warning**
>
> Jangan sertakan rahasia, kredensi, atau data sensitif URLs sebagai parameter jalur atau kueri. Nilai-nilai ini dapat dilihat dalam teks biasa di log akses aplikasi Amplify Anda.

Anda dapat menggunakan contoh berikut untuk mengarahkan path ke folder dengan nama yang cocok dengan nilai elemen string kueri di alamat asli:

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/docs?id=<my-blog-id-value`	`/documents/<my-blog-post-id-value>`	`permanent redirect (301)`	

Format JSON

```
[
  {
    "source": "/docs?id=<my-blog-id-value>",
    "status": "301",
    "target": "/documents/<my-blog-id-value>",
    "condition": null
  }
]
```

> **Note**
>
> Amplify meneruskan semua parameter string kueri ke jalur tujuan untuk pengalihan 301 dan 302. Namun, jika alamat asli menyertakan string kueri yang disetel ke nilai tertentu, seperti yang ditunjukkan dalam contoh ini, Amplify tidak meneruskan parameter kueri. Dalam hal ini, pengalihan hanya berlaku untuk permintaan ke alamat tujuan dengan nilai `id` kueri yang ditentukan.

Anda dapat menggunakan contoh berikut untuk mengarahkan semua jalur yang tidak dapat ditemukan pada tingkat tertentu dari struktur folder ke index.html dalam folder tertentu.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/documents/<folder>/<child-folder>/`	`/documents/index.html`	`not found (404)`	

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
<grand-child-folder>			

Format JSON

```
[
  {
    "source": "/documents/<x>/<y>/<z>",
    "status": "404",
    "target": "/documents/index.html",
    "condition": null
  }
]
```

Pengalihan berbasis wilayah

Anda dapat menggunakan contoh berikut untuk mengarahkan permintaan berdasarkan wilayah.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
/documents	/documents/us/	temporary redirect (302)	<US>

Format JSON

```
[
  {
    "source": "/documents",
    "status": "302",
    "target": "/documents/us/",
    "condition": "<US>"
  }
]
```

Menggunakan ekspresi wildcard dalam pengalihan dan penulisan ulang

Anda dapat menggunakan ekspresi wildcard,<*>, di alamat asli untuk pengalihan atau penulisan ulang. Anda harus menempatkan ekspresi di akhir alamat asli, dan itu harus unik. Amplify mengabaikan alamat asli yang menyertakan lebih dari satu ekspresi wildcard, atau menggunakannya dalam penempatan yang berbeda.

Berikut ini adalah contoh pengalihan yang valid dengan ekspresi wildcard.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

Dua contoh berikut menunjukkan pengalihan yang tidak valid dengan ekspresi wildcard.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
`/docs/<*>/content`	`/documents/<*>/content`	`permanent redirect (301)`	
`/docs/<*>/content/<*>`	`/documents/<*>/content/<*>`	`permanent redirect (301)`	

Menggunakan variabel lingkungan dalam aplikasi Amplify

Variabel lingkungan adalah pasangan nilai kunci yang dapat Anda tambahkan ke pengaturan aplikasi Anda untuk membuatnya tersedia untuk Amplify Hosting. Sebagai praktik terbaik, Anda dapat menggunakan variabel lingkungan untuk mengekspos data konfigurasi aplikasi. Semua variabel lingkungan yang Anda tambahkan dienkripsi untuk mencegah akses nakal.

Amplify memberlakukan batasan berikut pada variabel lingkungan yang Anda buat.

- Amplify tidak memungkinkan Anda membuat nama variabel lingkungan dengan awalan. `AWS` Awalan ini dicadangkan untuk Amplify penggunaan internal saja.
- Nilai variabel lingkungan tidak dapat melebihi 5500 karakter.

> **Important**
>
> Jangan gunakan variabel lingkungan untuk menyimpan rahasia. Untuk aplikasi Gen 2, gunakan fitur manajemen Rahasia di konsol Amplify. Untuk informasi selengkapnya, lihat Rahasia dan vars lingkungan di Amplify Documentation. Untuk aplikasi Gen 1, simpan rahasia dalam rahasia lingkungan yang dibuat menggunakan AWS Systems Manager Parameter Store. Untuk informasi selengkapnya, lihat Mengelola rahasia lingkungan.

Amplify referensi variabel lingkungan

Variabel lingkungan berikut dapat diakses secara default dalam konsol Amplify.

Nama variabel	Deskripsi	Nilai contoh
_BUILD_TIMEOUT	Durasi batas waktu build dalam menit. Nilai minimum adalah 5. Nilai maksimumnya adalah 120.	`30`

Nama variabel	Deskripsi	Nilai contoh
_LIVE_UPDATES	Alat akan ditingkatkan ke versi terbaru.	`[{"name":"Amplify CLI","pkg":"@aws-amplify/cli","type":"npm","version":"latest"}]`
USER_DISABLE_TESTS	Langkah pengujian dilewati selama pembuatan. Anda dapat menonaktifkan pengujian untuk semua cabang atau cabang tertentu di aplikasi. Variabel lingkungan ini digunakan untuk aplikasi yang melakukan pengujian selama fase build. Untuk informasi selengkapnya tentang pengaturan variabel ini, lihatMematikan pengujian untuk aplikasi atau cabang Amplify.	`true`
AWS_APP_ID	ID aplikasi build saat ini	`abcd1234`
AWS_BRANCH	Nama cabang build saat ini	`main`, `develop`, `beta`, `v2.0`
AWS_BRANCH_ARN	Cabang Amazon Resource Name (ARN) dari build saat ini	`aws:arn:amplify:us-west-2:123456789012:appname/branch/...`
AWS_CLONE_URL	URL klon yang digunakan untuk mengambil isi repositori git	`git@github.com:<user-name>/<repo-name>.git`

Nama variabel	Deskripsi	Nilai contoh
AWS_COMMIT_ID	ID komit dari build saat ini “KEPALA” untuk rebuild	`abcd1234`
AWS_JOB_ID	ID tugas build saat ini. ID tugas mencakup beberapa ‘0’ yang ditambahkan sehingga panjangnya selalu sama.	`0000000001`
AWS_PULL_REQUEST_ID	ID permintaan tarik dari build pratinjau web permintaan tarik. Variabel lingkungan ini tidak tersedia saat digunakan AWS CodeCommit sebagai penyedia repositori Anda.	`1`
AWS_PULL_REQUEST_SOURCE_BRANCH	Nama cabang fitur untuk pratinjau permintaan tarik yang dikirimkan ke cabang aplikasi di konsol Amplify.	`featureA`
AWS_PULL_REQUEST_DESTINATION_BRANCH	Nama cabang aplikasi di konsol Amplify tempat permintaan tarik cabang fitur dikirimkan.	`main`
AMPLIFY_AMAZON_CLIENT_ID	ID klien Amazon	`123456`
AMPLIFY_AMAZON_CLIENT_SECRET	Rahasia klien Amazon	`example123456`
AMPLIFY_FACEBOOK_CLIENT_ID	ID klien Facebook	`123456`

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_FACEBOOK_CLIENT_SECRET	Rahasia klien Facebook	`example123456`
AMPLIFY_GOOGLE_CLIENT_ID	ID klien Google	`123456`
AMPLIFY_GOOGLE_CLIENT_SECRET	Rahasia klien Google	`example123456`
AMPLIFY_DIFF_DEPLOY	Mengaktifkan atau menonaktifkan deployment frontend berbasis diff. Untuk informasi selengkapnya, lihat Mengonfigurasi build dan deploy frontend berbasis diff.	`true`
AMPLIFY_DIFF_DEPLOY_ROOT	Path yang digunakan untuk perbandingan deployment frontend berbasis diff, bergantung pada root repositori.	`dist`
AMPLIFY_DIFF_BACKEND	Aktifkan atau nonaktifkan build backend berbasis diff. Ini hanya berlaku untuk aplikasi Gen 1. Untuk informasi selengkapnya, lihat Mengonfigurasi build backend berbasis diff untuk aplikasi Gen 1	`true`

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_BACKEND_PULL_ONLY	Amplify mengelola variabel lingkungan ini. Ini hanya berlaku untuk aplikasi Gen 1. Untuk informasi selengkapnya, lihat Mengedit frontend yang ada agar mengarah ke backend berbeda	`true`
AMPLIFY_BACKEND_APP_ID	Amplify mengelola variabel lingkungan ini. Ini hanya berlaku untuk aplikasi Gen 1. Untuk informasi selengkapnya, lihat Mengedit frontend yang ada agar mengarah ke backend berbeda	`abcd1234`
AMPLIFY_SKIP_BACKEND_BUILD	Jika Anda tidak memiliki bagian backend dalam spesifikasi build dan ingin menonaktifkan build backend, setel variabel lingkungan ini ke. `true` Ini hanya berlaku untuk aplikasi Gen 1.	`true`
AMPLIFY_ENABLE_DEBUG_OUTPUT	Atur variabel ini `true` untuk mencetak jejak tumpukan di log. Ini berguna untuk men-debug kesalahan build backend.	`true`
AMPLIFY_MONOREPO_APP_ROOT	Path yang digunakan untuk menentukan root aplikasi dari aplikasi monorepo, bergantung pada root repositori.	`apps/react-app`

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_USERPOOL_ID	ID untuk kumpulan pengguna Amazon Cognito yang diimpor untuk autentikasi	`us-west-2_example`
AMPLIFY_WEBCLIENT_ID	ID untuk klien aplikasi yang akan digunakan oleh aplikasi web Klien aplikasi harus dikonfigurasi dengan akses ke kumpulan pengguna Amazon Cognito yang ditentukan oleh variabel lingkungan AMPLIFY_USERPOOL_ID.	`123456`
AMPLIFY_NATIVECLIENT_ID	ID untuk klien aplikasi yang akan digunakan oleh aplikasi asli Klien aplikasi harus dikonfigurasi dengan akses ke kumpulan pengguna Amazon Cognito yang ditentukan oleh variabel lingkungan AMPLIFY_USERPOOL_ID.	`123456`
AMPLIFY_IDENTITYPOOL_ID	ID untuk kumpulan identitas Amazon Cognito	`example-identitypool-id`
AMPLIFY_PERMISSIONS_BOUNDARY_ARN	ARN untuk kebijakan IAM untuk digunakan sebagai batas izin yang berlaku untuk semua peran IAM yang dibuat oleh Amplify.	`arn:aws:iam::123456789012:policy/example-policy`

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_DESTRUCTIVE_UPDATES	Setel variabel lingkungan ini ke true untuk memungkinkan GraphQL API diperbarui dengan operasi skema yang berpotensi menyebabkan kehilangan data.	`true`

> **Note**
>
> Variabel `AMPLIFY_AMAZON_CLIENT_ID` dan `AMPLIFY_AMAZON_CLIENT_SECRET` lingkungan adalah OAuth token, bukan kunci AWS akses dan kunci rahasia.

Variabel lingkungan kerangka kerja frontend

Jika Anda mengembangkan aplikasi dengan kerangka kerja frontend yang mendukung variabel lingkungannya sendiri, penting untuk dipahami bahwa ini tidak sama dengan variabel lingkungan yang Anda konfigurasikan di konsol Amplify. Misalnya, React (dengan prefiks REACT_APP) dan Gatsby (dengan prefiks GATSBY) memungkinkan Anda untuk membuat variabel lingkungan waktu aktif yang diintegrasikan secara otomatis ke dalam build produksi frontend Anda oleh kerangka kerja tersebut. Guna memahami efek penggunaan variabel lingkungan ini untuk menyimpan nilai, lihat dokumentasi terkait kerangka kerja frontend yang Anda gunakan.

Menyimpan nilai sensitif, seperti kunci API, di dalam variabel lingkungan dengan prefiks kerangka kerja frontend bukanlah praktik terbaik dan sangat tidak dianjurkan.

Mengatur variabel lingkungan

Gunakan petunjuk berikut untuk mengatur variabel lingkungan untuk aplikasi di konsol Amplify.

> **Note**
>
> Variabel lingkungan terlihat di menu Pengaturan aplikasi konsol Amplify hanya jika aplikasi disiapkan untuk penerapan berkelanjutan dan terhubung ke repositori git. Untuk langkah-langkah seputar jenis deployment ini, lihat Memulai dengan kode yang ada.

Untuk mengatur variabel lingkungan

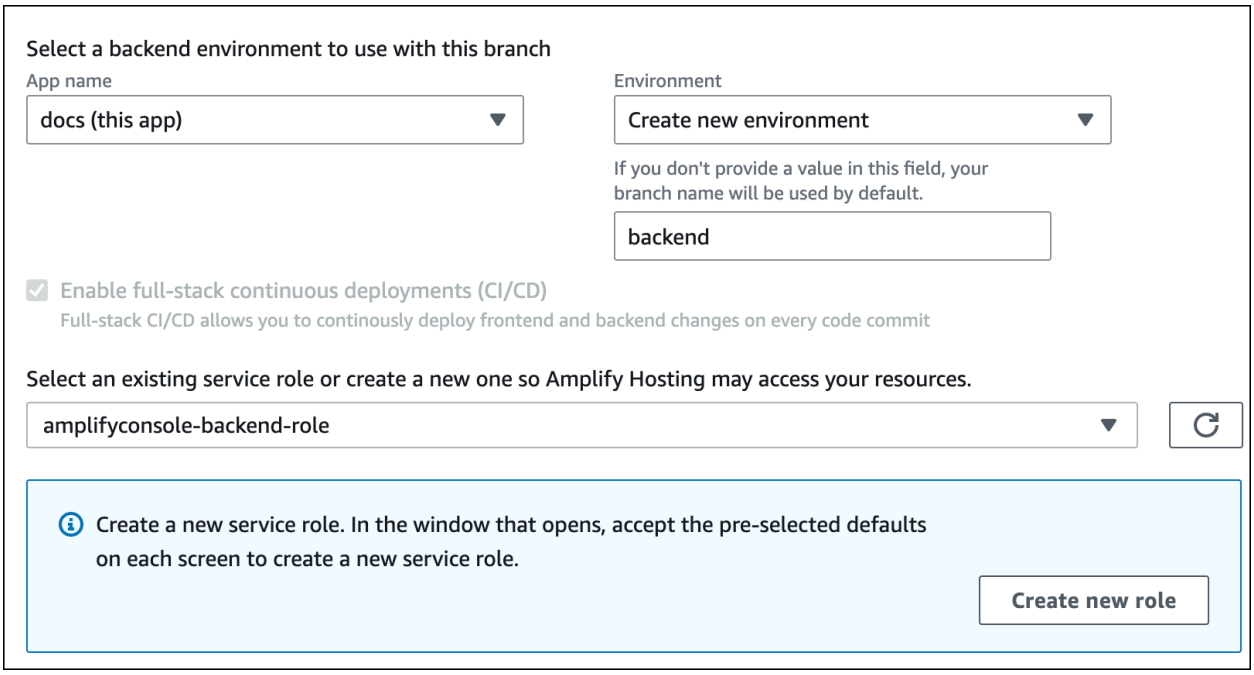
1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Di konsol Amplify, pilih Hosting, lalu pilih variabel Lingkungan.
3. Di halaman Variabel lingkungan, pilih Kelola variabel.
4. Untuk Variable, masukkan kunci Anda. Untuk Nilai, masukkan nilai Anda. Secara default, Amplify menerapkan variabel lingkungan di semua cabang, jadi Anda tidak perlu memasukkan kembali variabel saat menghubungkan cabang baru.
5. (Opsional) Untuk menyesuaikan variabel lingkungan secara khusus untuk sebuah cabang, tambahkan timpaan cabang sebagai berikut:
   a. Pilih Tindakan, lalu pilih Tambahkan timpaan variabel.
   b. Anda sekarang memiliki serangkaian variabel lingkungan khusus untuk cabang Anda.
6. Pilih Simpan.

Membuat lingkungan backend baru dengan parameter autentikasi untuk masuk sosial

Untuk menghubungkan cabang ke aplikasi

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Cara menghubungkan cabang ke aplikasi bervariasi, bergantung pada apakah Anda menghubungkan cabang ke aplikasi baru atau aplikasi yang sudah ada.
   - Menghubungkan cabang ke aplikasi baru
     a. Pada halaman Pengaturan Build, cari bagian Pilih lingkungan backend yang akan digunakan dengan cabang ini. Untuk Lingkungan, pilih Buat lingkungan baru, dan masukkan nama lingkungan backend Anda. Tangkapan layar berikut menunjukkan

Pilih lingkungan backend untuk digunakan dengan cabang ini bagian dari halaman Pengaturan Build dengan **`backend`** dimasukkan untuk nama lingkungan backend.



b. Perluas bagian Pengaturan lanjutan di halaman Pengaturan build dan tambahkan variabel lingkungan untuk kunci masuk sosial. Misalnya, **`AMPLIFY_FACEBOOK_CLIENT_SECRET`** adalah variabel lingkungan yang valid. Untuk daftar variabel lingkungan sistem Amplify yang tersedia secara default, lihat tabel di Amplify referensi variabel lingkungan.

- Menghubungkan cabang ke aplikasi yang sudah ada

  a. Jika Anda menghubungkan cabang baru ke aplikasi yang sudah ada, atur variabel lingkungan masuk sosial sebelum menghubungkan cabang. Di panel navigasi, pilih Pengaturan Aplikasi, lalu Variabel lingkungan.

  b. Di bagian Variabel lingkungan, pilih Kelola variabel.

  c. Di bagian Kelola variabel, pilih Tambahkan variabel.

  d. Untuk Variabel (kunci), masukkan ID klien Anda. Untuk Nilai, masukkan rahasia klien.

  e. Pilih, Simpan.

Mengelola rahasia lingkungan

Dengan dirilisnya Amplify Gen 2, alur kerja untuk rahasia lingkungan disederhanakan untuk memusatkan pengelolaan rahasia dan variabel lingkungan di konsol Amplify. Untuk petunjuk tentang

menyetel dan mengakses rahasia untuk aplikasi Amplify Gen 2, lihat Rahasia dan vars lingkungan di Dokumentasi Amplify.

Rahasia lingkungan untuk aplikasi Gen 1 mirip dengan variabel lingkungan, tetapi mereka adalah pasangan nilai kunci AWS Systems Manager Parameter Store yang dapat dienkripsi. Beberapa nilai harus dienkripsi, seperti kunci pribadi Masuk dengan Apple untuk Amplify.

Menggunakan AWS Systems Manager untuk mengatur rahasia lingkungan untuk aplikasi Amplify Gen 1

Gunakan petunjuk berikut untuk menyetel rahasia lingkungan untuk aplikasi Amplify Gen 1 menggunakan konsol. AWS Systems Manager

Untuk mengatur rahasia lingkungan

1. Masuk ke AWS Management Console dan buka AWS Systems Manager konsol.
2. Di panel navigasi, pilih Manajemen Aplikasi, lalu pilih Parameter Store.
3. Pada halaman AWS Systems Manager Parameter Store, pilih Buat parameter.
4. Pada halaman Create parameter, di bagian Parameter details, lakukan hal berikut:
    a. Untuk Nama, masukkan parameter dalam format**`/amplify/{your_app_id}/{your_backend_environment_name}/{your_parameter_name}`**.
    b. Untuk Jenis, pilih SecureString.
    c. Untuk sumber kunci KMS, pilih Akun saya saat ini untuk menggunakan kunci default untuk akun Anda.
    d. Untuk Nilai, masukkan nilai rahasia Anda untuk mengenkripsi.
5. Pilih, Buat parameter.

> **Note**
>
> Amplify hanya memiliki akses ke kunci di bawah `/amplify/{your_app_id}/{your_backend_environment_name}` untuk build lingkungan tertentu. Anda harus menentukan default AWS KMS key untuk mengizinkan Amplify mendekripsi nilai.

Mengakses rahasia lingkungan untuk aplikasi Gen 1

Rahasia lingkungan untuk aplikasi Gen 1 disimpan `process.env.secrets` sebagai string JSON.

Amplify referensi rahasia lingkungan

Tentukan parameter Systems Manager dalam format`/amplify/{your_app_id}/{your_backend_environment_name}/AMPLIFY_SIWA_CLIENT_ID`.

Anda dapat menggunakan rahasia lingkungan berikut yang dapat diakses secara default dalam konsol Amplify.

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_SIWA_CLIENT_ID	Masuk dengan ID klien Apple	`com.yourapp.auth`
AMPLIFY_SIWA_TEAM_ID	Masuk dengan ID tim Apple	`ABCD123`
AMPLIFY_SIWA_KEY_ID	Masuk dengan ID kunci Apple	`ABCD123`
AMPLIFY_SIWA_PRIVATE_KEY	Masuk dengan kunci pribadi Apple	------ MULAI KUNCI PRIBADI ------ ****..... ------ AKHIRI KUNCI PRIBADI ------

Menyetel header khusus untuk aplikasi Amplify

Dengan header HTTP kustom, Anda dapat menentukan header untuk setiap respons HTTP. Header respons dapat digunakan untuk tujuan debugging, keamanan, dan informasi. Anda dapat menentukan header di konsol Amplify, atau dengan mengunduh dan mengedit file aplikasi `customHttp.yml` dan menyimpannya di direktori root proyek. Untuk prosedur terperinci, lihat Mengatur header kustom.

Sebelumnya, header HTTP kustom ditentukan untuk aplikasi baik dengan mengedit spesifikasi build (buildspec) di konsol Amplify atau dengan mengunduh dan memperbarui `amplify.yml` file dan menyimpannya di direktori root proyek. Kami sangat menyarankan untuk memigrasikan header khusus yang ditentukan dengan cara ini keluar dari buildspec dan file. `amplify.yml` Untuk petunjuk, silakan lihat Memigrasi header kustom keluar dari spesifikasi build dan amplify.yml.

Topik

- Referensi YAMAL header kustom
- Mengatur header kustom
- Memigrasi header kustom keluar dari spesifikasi build dan amplify.yml
- Persyaratan header kustom Monorepo

Referensi YAMAL header kustom

Tentukan header kustom menggunakan format YAML berikut:

```
customHeaders:
  - pattern: '*.json'
    headers:
    - key: 'custom-header-name-1'
      value: 'custom-header-value-1'
    - key: 'custom-header-name-2'
      value: 'custom-header-value-2'
  - pattern: '/path/*'
    headers:
    - key: 'custom-header-name-1'
      value: 'custom-header-value-2'
```

Untuk monorepo, gunakan format YAML berikut:

```
applications:
  - appRoot: app1
    customHeaders:
    - pattern: '**/*'
      headers:
      - key: 'custom-header-name-1'
        value: 'custom-header-value-1'
  - appRoot: app2
    customHeaders:
    - pattern: '/path/*.json'
      headers:
      - key: 'custom-header-name-2'
        value: 'custom-header-value-2'
```

Saat menambahkan header kustom ke aplikasi, Anda akan menentukan sendiri nilai untuk hal berikut:

pola

Header khusus diterapkan ke semua jalur file URL yang cocok dengan pola.

headers

Mendefinisikan header yang sesuai dengan pola file.

kunci

Nama header kustom.

nilai

Nilai header kustom.

Untuk informasi lebih lanjut tentang header HTTP, lihat daftar Mozilla untuk Header HTTP.

Mengatur header kustom

Ada dua cara untuk menentukan header HTTP kustom untuk aplikasi Amplify. Anda dapat menentukan header di konsol Amplify atau Anda dapat menentukan header dengan mengunduh dan mengedit file aplikasi `customHttp.yml` dan menyimpannya di direktori root proyek Anda.

Untuk menyetel header khusus untuk aplikasi dan menyimpannya di konsol

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang header kustomnya akan diatur.
3. Di panel navigasi, pilih Hosting, lalu pilih Header khusus.
4. Pada halaman Custom header, pilih Edit.
5. Di jendela Edit header kustom, masukkan informasi untuk header kustom Anda menggunakan format YAMAL header kustom.
   a. Untuk `pattern`, masukkan pola yang akan dicocokkan.
   b. Untuk `key`, masukkan nama header kustom.
   c. Untuk `value`, masukkan nilai header kustom.
6. Pilih Simpan.
7. Menerapkan ulang aplikasi untuk menerapkan header kustom baru.
   - Untuk aplikasi CI/CD, navigasikan ke cabang untuk menyebarkan dan pilih Redeploy versi ini. Anda juga dapat melakukan build baru dari repositori Git Anda.
   - Untuk aplikasi penerapan manual, terapkan aplikasi lagi di konsol Amplify.

Untuk mengatur header khusus untuk aplikasi dan menyimpannya di root repositori Anda

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang header kustomnya akan diatur.
3. Di panel navigasi, pilih Hosting, lalu pilih Header khusus.
4. Pada halaman Custom header, pilih Download YML.
5. Buka unduhan file `customHttp.yml` di editor kode pilihan Anda, lalu masukkan informasi untuk header kustom Anda menggunakan format YAML header kustom.
   a. Untuk `pattern`, masukkan pola yang akan dicocokkan.
   b. Untuk `key`, masukkan nama header kustom.
   c. Untuk `value`, masukkan nilai header kustom.
6. Simpan file `customHttp.yml` yang telah diedit di direktori root proyek. Jika Anda menggunakan monorepo, simpan file `customHttp.yml` di root repo.
7. Menerapkan ulang aplikasi untuk menerapkan header kustom baru.

- Untuk aplikasi CI/CD, jalankan build baru dari repositori Git yang mencakup file `customHttp.yml` baru.
- Untuk aplikasi penerapan manual, terapkan aplikasi lagi di konsol Amplify dan sertakan `customHttp.yml` file baru dengan artefak yang Anda unggah.

> **Note**
>
> Header khusus yang disetel dalam `customHttp.yml` file dan diterapkan di direktori root aplikasi akan menimpa header kustom yang ditentukan di bagian Header khusus di konsol Amplify.

Contoh header kustom keamanan

Header keamanan kustom memungkinkan penerapan HTTPS, mencegah serangan XSS, dan melindungi peramban dari clickjacking. Gunakan sintaks YAML berikut untuk menerapkan header keamanan kustom ke aplikasi Anda.

```
customHeaders:
  - pattern: '**'
    headers:
      - key: 'Strict-Transport-Security'
        value: 'max-age=31536000; includeSubDomains'
      - key: 'X-Frame-Options'
        value: 'SAMEORIGIN'
      - key: 'X-XSS-Protection'
        value: '1; mode=block'
      - key: 'X-Content-Type-Options'
        value: 'nosniff'
      - key: 'Content-Security-Policy'
        value: "default-src 'self'"
```

Mengatur header kustom Cache-Control

Aplikasi yang dihosting dengan Amplify menghormati `Cache-Control` header yang dikirim oleh asal, kecuali jika Anda menggantinya dengan header khusus yang Anda tentukan. Amplify hanya menerapkan header kustom Cache-Control untuk respons yang berhasil dengan kode status.

`200 OK` Ini mencegah respons kesalahan di-cache dan disajikan ke pengguna lain yang membuat permintaan yang sama.

Anda dapat menyesuaikan perintah `s-maxage` secara manual untuk mendapatkan kendali yang lebih besar atas performa dan ketersediaan deployment aplikasi Anda. Misalnya, untuk menambah durasi konten tetap berada di cache di tepi, Anda dapat meningkatkan waktu untuk tayang (TTL) secara manual dengan memperbarui `s-maxage` ke nilai yang lebih lama dari default 600 detik (10 menit).

Untuk menentukan nilai kustom untuk `s-maxage`, gunakan format YAML berikut. Contoh ini membuat konten terkait tetap berada di cache di tepi selama 3600 detik (satu jam).

```
customHeaders:
  - pattern: '/img/*'
    headers:
      - key: 'Cache-Control'
        value: 's-maxage=3600'
```

Untuk informasi selengkapnya tentang mengontrol performa aplikasi dengan header, lihatMenggunakan header Cache-Control untuk meningkatkan performa aplikasi.

Memigrasi header kustom keluar dari spesifikasi build dan amplify.yml

Sebelumnya, header HTTP kustom ditentukan untuk aplikasi baik dengan mengedit spesifikasi build di konsol Amplify atau dengan mengunduh dan memperbarui `amplify.yml` file dan menyimpannya di direktori root proyek. Sangat disarankan agar Anda memigrasikan header kustom Anda keluar dari spesifikasi build dan file. `amplify.yml`

Tentukan header khusus Anda di bagian Header khusus di konsol Amplify atau dengan mengunduh dan mengedit file. `customHttp.yml`

Untuk memigrasikan header kustom yang disimpan di konsol Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi untuk menjalankan migrasi header kustom.
3. Di panel navigasi, pilih Hosting, Build settings. Di bagian Spesifikasi build aplikasi, Anda dapat meninjau buildspec aplikasi.

4. Pilih Unduh untuk menyimpan salinan buildspec saat ini. Anda dapat menggunakan salinan ini kemudian sebagai referensi untuk memulihkan pengaturan.
5. Setelah pengunduhan selesai, pilih Edit.
6. Perhatikan informasi header kustom di file karena informasi ini akan Anda gunakan di langkah 9. Di jendela Edit, hapus header khusus apa pun dari file dan pilih Simpan.
7. Di panel navigasi, pilih Hosting, Header khusus.
8. Pada halaman Custom header, pilih Edit.
9. Di jendela Edit header khusus, masukkan informasi untuk header kustom Anda yang Anda hapus pada langkah 6.
10. Pilih Simpan.
11. Deploy ulang cabang tempat header kustom baru akan diterapkan.

Memigrasi header kustom dari amplify.yml ke customHttp.yml

1. Navigasi ke file `amplify.yml` yang saat ini di-deploy di direktori root aplikasi.
2. Buka `amplify.yml` di editor kode pilihan Anda.
3. Perhatikan informasi header kustom di file karena informasi ini akan Anda gunakan di langkah 8. Hapus header kustom di file. Simpan dan tutup file .
4. Masuk ke AWS Management Console dan buka konsol Amplify.
5. Pilih aplikasi yang header kustomnya akan diatur.
6. Di panel navigasi, pilih Hosting, Header khusus.
7. Pada halaman Custom header, pilih Download.
8. Buka unduhan file `customHttp.yml` di editor kode pilihan Anda, lalu masukkan informasi untuk header kustom yang Anda hapus dari `amplify.yml` di langkah 3.
9. Simpan file `customHttp.yml` yang telah diedit di direktori root proyek. Jika Anda menggunakan monorepo, simpan file di root repo.
10. Menerapkan ulang aplikasi untuk menerapkan header kustom baru.
    - Untuk aplikasi CI/CD, jalankan build baru dari repositori Git yang mencakup file `customHttp.yml` baru.
    - Untuk aplikasi penerapan manual, terapkan aplikasi lagi di konsol Amplify dan sertakan `customHttp.yml` file baru dengan artefak yang Anda unggah.

> **Note**
>
> Header khusus yang disetel dalam `customHttp.yml` file dan diterapkan di direktori root aplikasi akan menimpa header khusus yang ditentukan di bagian Header khusus di konsol Amplify.

Persyaratan header kustom Monorepo

Saat Anda menentukan header khusus untuk aplikasi di monorepo, perhatikan persyaratan penyiapan berikut:

- Tersedia format YAML khusus untuk monorepo. Untuk sintaks yang benar, lihatReferensi YAMAL header kustom.
- Anda dapat menentukan header kustom untuk aplikasi dalam monorepo menggunakan bagian Custom header dari konsol Amplify. Anda harus menerapkan ulang aplikasi Anda untuk menerapkan header kustom baru.
- Selain menggunakan konsol, Anda dapat menentukan header kustom untuk aplikasi di monorepo dalam file `customHttp.yml`. Anda harus menyimpan file `customHttp.yml` di root repo, kemudian men-deploy kembali aplikasi untuk menerapkan header kustom baru. Header khusus yang ditentukan dalam `customHttp.yml` file akan mengganti header kustom yang ditentukan menggunakan bagian Custom header dari konsol Amplify.

Mengelola konfigurasi cache untuk aplikasi

Amplify menggunakan Amazon CloudFront untuk mengelola konfigurasi caching untuk aplikasi yang dihosting. Konfigurasi cache diterapkan ke setiap aplikasi untuk mengoptimalkan kinerja terbaik.

Pada 13 Agustus 2024, Amplify merilis peningkatan efisiensi caching untuk aplikasi. Untuk informasi selengkapnya, lihat Peningkatan Caching CDN untuk Kinerja Aplikasi yang Lebih Baik dengan AWS Amplify Hosting.

Tabel berikut merangkum dukungan Amplify untuk perilaku caching tertentu sebelum dan sesudah rilis peningkatan caching.

Perilaku caching	Dukungan sebelumnya	Dengan peningkatan caching
Anda dapat menambahkan header khusus untuk aplikasi di konsol Amplify atau dalam `customHeaders.yaml` file. Salah satu header yang dapat Anda ganti adalah. `Cache-Control` Untuk informasi selengkapnya, lihat Menyetel header khusus untuk aplikasi Amplify.	Ya	Ya
Amplify menghormati `Cache-Control` header yang Anda tentukan dalam `customHeaders.yaml` file dan mereka lebih diutamakan daripada pengaturan cache default Amplify.	Ya	Ya
Amplify menghormati `Cache-Control` header yang ditetapkan dalam kerangka kerja aplikasi untuk rute	Ya	Ya

Perilaku caching	Dukungan sebelumnya	Dengan peningkatan caching
dinamis (misalnya, rute SSR Next.js). Jika `Cache-Control` header disetel dalam `customHeaders.yaml` file aplikasi, ini lebih diutamakan daripada setelan dalam file. `next.config.js`		
Setiap penerapan CI/CD aplikasi baru menghapus cache.	Ya	Ya
Anda dapat mengaktifkan mode kinerja untuk aplikasi.	Ya	Tidak Pengaturan mode kinerja tidak lagi tersedia di konsol Amplify. Namun, Anda dapat membuat `Cache-Control` header yang menetapkan `s-maxage` direktif. Untuk petunjuk, lihat Menggunakan header Cache-Control untuk meningkatkan performa aplikasi.

Tabel berikut mencantumkan perubahan pada nilai default untuk pengaturan cache tertentu.

Pengaturan cache	Nilai default sebelumnya	Nilai default dengan peningkatan caching
Durasi cache untuk aset statis	Dua detik	Satu tahun
Durasi cache untuk respons proxy terbalik	Dua detik	Nol detik (tidak ada caching)
Waktu Maks untuk Hidup (TTL)	Sepuluh menit	Satu tahun

Untuk informasi selengkapnya tentang cara Amplify menentukan konfigurasi caching yang akan diterapkan ke aplikasi dan instruksi tentang mengelola konfigurasi kunci cache, lihat topik berikut.

Topik

Bagaimana Amplify menerapkan konfigurasi cache ke aplikasi

Untuk mengelola caching untuk aplikasi Anda, Amplify menentukan jenis konten yang sedang disajikan dengan memeriksa jenis platform aplikasi dan aturan penulisan ulang. Untuk `Compute` aplikasi, Amplify juga memeriksa aturan perutean dalam manifes penerapan.

> **Note**
>
> Jenis platform aplikasi diatur oleh Amplify Hosting selama penerapan. Aplikasi SSG (statis) disetel ke jenis `WEB` platform. Aplikasi SSR (Next.js 12 atau yang lebih baru) diatur ke jenis `WEB_COMPUTE` platform.

Amplify mengidentifikasi empat jenis konten berikut dan menerapkan kebijakan cache terkelola yang ditentukan.

Statis

Konten yang disajikan dari aplikasi dengan `WEB` platform, atau rute statis dalam `WEB_COMPUTE` aplikasi.

Konten ini menggunakan kebijakan Amplify-StaticContent cache.

Optimasi Gambar

Gambar yang disajikan oleh `ImageOptimization` rute dalam `WEB_COMPUTE` aplikasi.

Konten ini menggunakan kebijakan Amplify-ImageOptimization cache.

Komputasi

Konten yang disajikan oleh `Compute` rute dalam `WEB_COMPUTE` aplikasi. Ini termasuk semua konten yang dirender sisi server (SSR).

Konten ini menggunakan kebijakan Amplify-Default atau Amplify-DefaultNoCookies cache tergantung pada nilai `cacheConfig.type` yang disetel pada Amplify App Anda.

Proksi Terbalik

Konten yang disajikan oleh jalur yang cocok dengan aturan kustom penulisan ulang proxy terbalik. Untuk informasi selengkapnya tentang membuat aturan kustom ini, lihat Penulisan ulang proksi balik di bagian Menggunakan pengalihan.

Konten ini menggunakan kebijakan Amplify-Default atau Amplify-DefaultNoCookies cache tergantung pada nilai `cacheConfig.type` yang disetel pada Amplify App Anda.

Memahami kebijakan cache terkelola Amplify

Amplify menggunakan kebijakan cache terkelola yang telah ditentukan sebelumnya untuk mengoptimalkan konfigurasi cache default untuk aplikasi yang dihosting.

- Amplify-Default
- Amplify-DefaultNoCookies
- Amplify-ImageOptimization
- Amplify-StaticContent

Pengaturan kebijakan cache terkelola Amplify-Default

Lihat kebijakan ini di CloudFront konsol

Kebijakan ini dirancang untuk digunakan dengan asal yang merupakan aplikasi AWS Amplifyweb.

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 31536000 detik (satu tahun)
- Default TTL: 0 detik
- Header termasuk dalam kunci cache:
  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`

  - `Host`
- Cookie termasuk dalam kunci cache: Semua cookie disertakan.
- String kueri disertakan dalam kunci cache: Semua string kueri disertakan.
- Pengaturan objek terkompresi cache: Gzip dan Brotli diaktifkan.

Amplify- pengaturan kebijakan cache DefaultNoCookies terkelola

Lihat kebijakan ini di CloudFront konsol

Kebijakan ini dirancang untuk digunakan dengan asal yang merupakan aplikasi AWS Amplifyweb.

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 31536000 detik (satu tahun)
- Default TTL: 0 detik
- Header termasuk dalam kunci cache:
  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`
  - `Host`
- Cookie termasuk dalam kunci cache: Tidak ada cookie yang disertakan.
- String kueri disertakan dalam kunci cache: Semua string kueri disertakan.
- Pengaturan objek terkompresi cache: Gzip dan Brotli diaktifkan.

Amplify- pengaturan kebijakan cache ImageOptimization terkelola

Lihat kebijakan ini di CloudFront konsol

Kebijakan ini dirancang untuk digunakan dengan asal yang merupakan aplikasi AWS Amplifyweb.

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 31536000 detik (satu tahun)

- Default TTL: 0 detik
- Header termasuk dalam kunci cache:
  - `Authorization`
  - `Accept`
  - `Host`
- Cookie termasuk dalam kunci cache: Tidak ada cookie yang disertakan.
- String kueri disertakan dalam kunci cache: Semua string kueri disertakan.
- Pengaturan objek terkompresi cache: Gzip dan Brotli diaktifkan.

Amplify- pengaturan kebijakan cache StaticContent terkelola

Lihat kebijakan ini di CloudFront konsol

Kebijakan ini dirancang untuk digunakan dengan asal yang merupakan aplikasi AWS Amplifyweb.

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 31536000 detik (satu tahun)
- Default TTL: 0 detik
- Header termasuk dalam kunci cache:
  - `Authorization`
  - `Host`
- Cookie termasuk dalam kunci cache: Tidak ada cookie yang disertakan.
- String kueri disertakan dalam kunci cache: Tidak ada string kueri yang disertakan.
- Pengaturan objek terkompresi cache: Gzip dan Brotli diaktifkan.

Mengelola cookie kunci cache

Saat menerapkan aplikasi ke Amplify, Anda dapat memilih apakah ingin menyertakan atau mengecualikan cookie di kunci cache. Di konsol Amplify, pengaturan ini ditentukan pada header Kustom dan halaman cache menggunakan sakelar pengaturan kunci Cache. Untuk petunjuk, lihat Menyertakan atau mengecualikan cookie dari kunci cache.

Sertakan cookie di kunci cache

Dengan setelan ini, Amplify secara otomatis memilih konfigurasi cache yang optimal untuk aplikasi Anda berdasarkan jenis konten yang sedang disajikan. Anda harus secara eksplisit memilih jenis konfigurasi cache ini.

Jika Anda menggunakan SDKs atau AWS CLI, pengaturan ini sesuai dengan pengaturan `cacheConfig.type` `AMPLIFY_MANAGED` dengan `CreateApp` atau `UpdateApp` APIs.

Kecualikan cookie dari kunci cache

Ini adalah konfigurasi cache default. Konfigurasi cache ini mirip dengan `AMPLIFY_MANAGED` konfigurasi, kecuali bahwa itu mengecualikan semua cookie dari kunci cache.

Memilih untuk mengecualikan cookie dari kunci cache dapat menghasilkan kinerja cache yang lebih baik. Namun, sebelum Anda memilih konfigurasi cache ini, penting untuk mempertimbangkan apakah aplikasi Anda menggunakan cookie untuk menyajikan konten dinamis.

Jika Anda menggunakan SDKs atau AWS CLI, pengaturan ini sesuai dengan pengaturan `cacheConfig.type` ke `AMPLIFY_MANAGED_NO_COOKIES` dengan `CreateApp` atau `UpdateApp` APIs.

Untuk informasi selengkapnya tentang kunci cache, lihat Memahami kunci cache di Panduan CloudFront Pengembang Amazon;.

Menyertakan atau mengecualikan cookie dari kunci cache

Anda dapat mengatur konfigurasi cookie kunci cache untuk aplikasi di konsol Amplify, SDKs, atau file. AWS CLI

Gunakan prosedur berikut untuk menentukan apakah akan menyertakan atau mengecualikan cookie dari kunci cache saat Anda menerapkan aplikasi baru menggunakan konsol Amplify.

Untuk mengatur konfigurasi cookie kunci cache saat menerapkan aplikasi ke Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pada halaman Semua aplikasi, pilih Buat aplikasi baru.
3. Pada halaman Mulai membangun dengan Amplify, pilih penyedia repositori Git Anda, lalu pilih Berikutnya.

4. Di halaman Tambahkan cabang repositori, lakukan langkah berikut:

   a. Pilih nama repositori untuk terhubung.

   b. Pilih nama cabang repositori untuk terhubung.

   c. Pilih Berikutnya.

5. Jika aplikasi memerlukan peran layanan IAM, Anda dapat mengizinkan komputasi Amplify Hosting untuk secara otomatis membuat peran layanan untuk Anda atau Anda dapat menentukan peran yang telah Anda buat.

   - Untuk mengizinkan Amplify membuat peran secara otomatis dan melampirkannya ke aplikasi Anda:

     - Pilih Buat dan gunakan peran layanan baru.

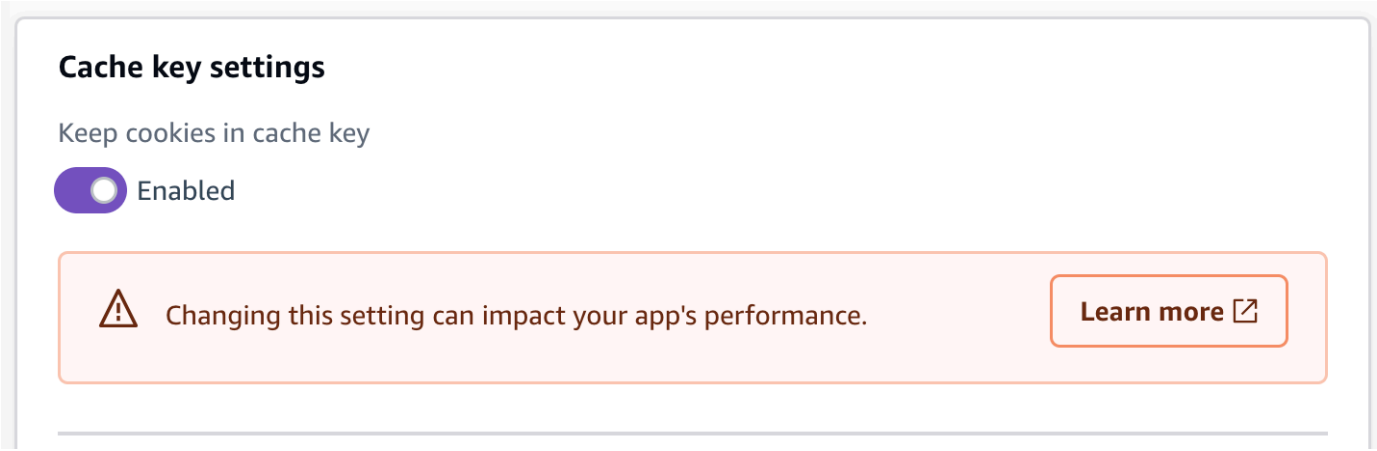
   - Untuk melampirkan peran layanan yang sebelumnya Anda buat:

     a. Pilih Gunakan peran layanan yang ada.

     b. Pilih peran yang akan digunakan dari daftar.

6. Pilih Pengaturan lanjutan, lalu cari bagian Pengaturan kunci cache.

7. Pilih salah satu Simpan cookie di kunci cache atau Hapus cookie dari kunci cache. Tangkapan layar berikut menunjukkan tombol Cache pengaturan toggle di konsol.



8. Pilih Berikutnya.

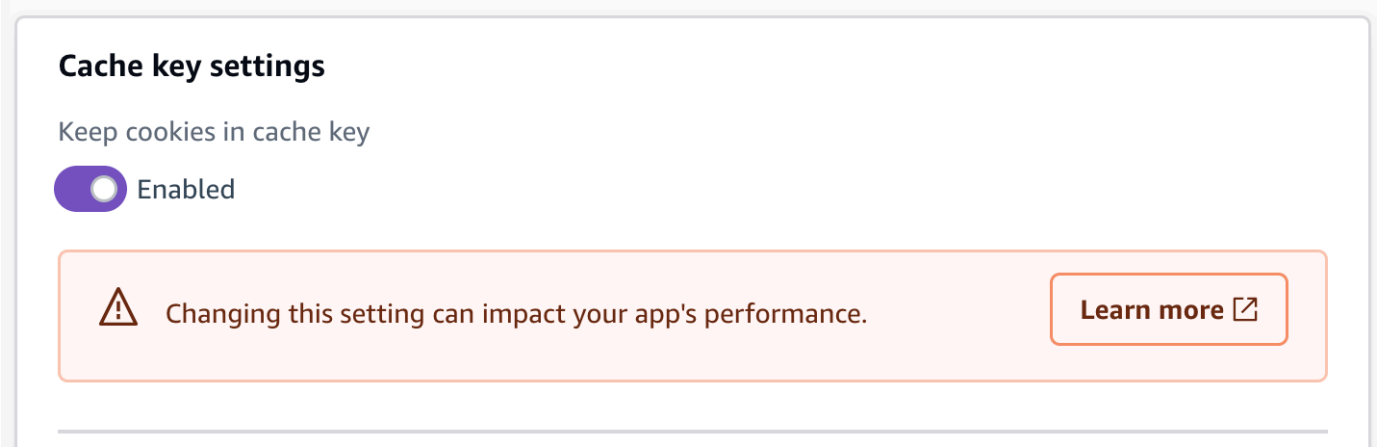
9. Di halaman Tinjauan, pilih Simpan dan deploy.

Mengubah konfigurasi cookie kunci cache untuk aplikasi

Anda dapat mengubah konfigurasi cookie kunci cache untuk aplikasi yang sudah di-deploy ke Amplify. Gunakan prosedur berikut untuk mengubah apakah akan menyertakan atau mengecualikan cookie dari kunci cache untuk aplikasi yang menggunakan konsol Amplify.

Untuk mengubah konfigurasi cookie kunci cache untuk aplikasi yang digunakan

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pada halaman Semua aplikasi, pilih aplikasi yang ingin Anda perbarui.
3. Di panel navigasi, pilih Hosting, lalu pilih Header dan cache khusus.
4. Pada halaman Custom header dan cache, cari bagian Pengaturan kunci cache dan pilih Edit.
5. Pilih salah satu Simpan cookie di kunci cache atau Hapus cookie dari kunci cache. Tangkapan layar berikut menunjukkan tombol Cache pengaturan toggle di konsol.



6. Pilih Simpan.

Menggunakan header Cache-Control untuk meningkatkan performa aplikasi

Arsitektur hosting default Amplify mengoptimalkan keseimbangan antara kinerja hosting dan ketersediaan penerapan. Untuk sebagian besar pelanggan, kami menyarankan Anda menggunakan arsitektur default.

Jika Anda memerlukan kontrol yang lebih baik atas kinerja aplikasi, Anda dapat mengatur `Cache-Control` header HTTP secara manual untuk mengoptimalkan kinerja hosting dengan menjaga konten di-cache di tepi jaringan pengiriman konten (CDN) untuk interval yang lebih lama.

`Cache-Control`Header `max-age` dan `s-maxage` arahan HTTP memengaruhi durasi caching konten untuk aplikasi Anda. Perintah `max-age` memberi tahu peramban durasi (dalam detik) yang Anda inginkan agar konten tetap berada di cache sebelum direfresh dari server asal. Perintah `s-maxage` menimpa `max-age` dan memungkinkan Anda menentukan durasi (dalam detik) yang Anda inginkan agar konten tetap berada di tepi CDN sebelum direfresh dari server asal.

Aplikasi yang dihosting dengan Amplify menghormati `Cache-Control` header yang dikirim oleh asal, kecuali jika Anda menggantinya dengan header khusus yang Anda tentukan. Amplify hanya menerapkan header `Cache-Control` khusus untuk respons yang berhasil dengan kode status. `200 OK` Ini mencegah respons kesalahan di-cache dan disajikan ke pengguna lain yang membuat permintaan yang sama.

Anda dapat menyesuaikan perintah `s-maxage` secara manual untuk mendapatkan kendali yang lebih besar atas performa dan ketersediaan deployment aplikasi Anda. Misalnya, untuk mengubah lamanya waktu konten Anda tetap di-cache di tepi, Anda dapat mengatur waktu untuk hidup (TTL) secara manual dengan memperbarui `s-maxage` ke nilai selain default 31536000 detik (satu tahun).

Anda dapat menentukan header kustom untuk aplikasi di bagian Custom header pada konsol Amplify. Untuk contoh YAML format, lihatMengatur header kustom Cache-Control.

Gunakan prosedur berikut untuk mengatur `s-maxage` arahan agar konten tetap di-cache di tepi CDN selama 24 jam.

Untuk mengatur Cache-Control header kustom

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang header kustomnya akan diatur.
3. Di panel navigasi, pilih Hosting, Header khusus.
4. Pada halaman Custom header, pilih Edit.
5. Di jendela Edit header khusus, masukkan informasi untuk header kustom Anda sebagai berikut:
   a. Untuk`pattern`, masukkan **`**/*`** untuk semua jalur.
   b. Untuk `key`, masukkan **`Cache-Control`**.
   c. Untuk `value`, masukkan **`s-maxage=86400`**.
6. Pilih Simpan.
7. Menerapkan ulang aplikasi untuk menerapkan header kustom baru.

Perlindungan miring untuk penerapan Amplify

Perlindungan kemiringan penerapan tersedia untuk Amplify aplikasi untuk menghilangkan masalah kemiringan versi antara klien dan server dalam aplikasi web. Saat menerapkan perlindungan miring ke aplikasi Amplify, Anda dapat memastikan bahwa klien Anda selalu berinteraksi dengan versi aset sisi server yang benar, terlepas dari kapan penerapan terjadi.

Versi miring adalah tantangan umum bagi pengembang web. Ini terjadi ketika browser web menjalankan versi aplikasi yang sudah ketinggalan zaman dan server menjalankan yang baru. Perbedaan ini dapat menyebabkan perilaku yang tidak terduga, kesalahan, dan pengalaman terdegradasi bagi pengguna aplikasi. Fitur perlindungan miring penerapan Amplify menyematkan klien yang berjalan di browser web ke penerapan tertentu. Ini memastikan bahwa Amplify selalu melayani aset untuk penerapan tertentu, menjaga klien dan server tetap disinkronkan.

Fitur perlindungan miring Amplify dapat mengurangi kesalahan bagi pengguna aplikasi Anda saat Anda merilis penerapan baru. Ini juga dapat meningkatkan pengalaman pengembang dengan mengurangi waktu yang dihabiskan untuk mengelola masalah kompatibilitas mundur dan maju.

Detail fitur perlindungan miring:

Tipe aplikasi yang didukung

- Anda dapat menambahkan perlindungan miring ke aplikasi statis dan SSR yang dibuat dengan kerangka kerja apa pun yang didukung Amplify. Aplikasi dapat digunakan dari repositori Git atau penerapan manual.
- Anda tidak dapat menambahkan perlindungan miring ke aplikasi yang diterapkan ke `WEB_DYNAMIC` platform (Next.js versi 11 atau lebih lama).

Durasi

- Untuk aplikasi statis, Amplify melayani penerapan satu minggu. Untuk aplikasi SSR, kami menjamin perlindungan miring hingga delapan penerapan sebelumnya.

Biaya

- Tidak ada biaya tambahan untuk menambahkan perlindungan skeke ke aplikasi.

Pertimbangan kinerja

Ketika perlindungan miring diaktifkan untuk aplikasi, Amplify harus memperbarui konfigurasi cache CDN-nya. Oleh karena itu, Anda harus mengharapkan penerapan pertama Anda setelah mengaktifkan perlindungan miring memakan waktu hingga sepuluh menit.

Topik

- Mengkonfigurasi proteksi kemiringan penerapan untuk aplikasi Amplify
- Cara kerja perlindungan miring

Mengkonfigurasi proteksi kemiringan penerapan untuk aplikasi Amplify

Anda dapat menambah atau menghapus proteksi kemiringan penerapan untuk aplikasi menggunakan konsol Amplify, the, atau. AWS Command Line Interface SDKs Fitur ini diterapkan di tingkat cabang. Hanya penerapan baru, yang dibuat setelah perlindungan miring diaktifkan untuk cabang, yang akan dilindungi miring.

Untuk menambah atau menghapus perlindungan kemiringan penerapan menggunakan AWS CLI or SDKs, gunakan bidang dan`CreateBranch.enableSkewProtection`. `UpdateBranch.enableSkewProtection` Untuk informasi selengkapnya, lihat CreateBranchdan UpdateBranchdi dokumentasi referensi Amplify API.

Jika Anda ingin menghapus penerapan tertentu sehingga tidak lagi dilayani, gunakan `DeleteJob` API. Untuk informasi selengkapnya, lihat DeleteJobdi dokumentasi referensi Amplify API.

Pada saat ini, Anda hanya dapat mengaktifkan perlindungan miring pada aplikasi yang sudah digunakan untuk Amplify Hosting. Gunakan petunjuk berikut untuk menambahkan perlindungan miring ke cabang menggunakan konsol Amplify.

Aktifkan perlindungan miring untuk cabang aplikasi Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Di halaman Semua aplikasi, pilih nama aplikasi yang diterapkan untuk mengaktifkan perlindungan miring.

3. Di panel navigasi, pilih Pengaturan aplikasi, lalu pilih Pengaturan cabang.
4. Di bagian Cabang, pilih nama cabang yang akan diperbarui.
5. Pada menu Tindakan, pilih Aktifkan perlindungan miring.
6. Di jendela konfirmasi, pilih Konfirmasi. Perlindungan skew sekarang diaktifkan untuk cabang.
7. Menerapkan ulang cabang aplikasi Anda. Hanya penerapan yang dilakukan setelah perlindungan miring diaktifkan yang dilindungi miring.

Gunakan petunjuk berikut untuk menghapus perlindungan miring dari cabang aplikasi menggunakan konsol Amplify.

Hapus perlindungan skedari cabang aplikasi Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Di halaman Semua aplikasi, pilih nama aplikasi yang digunakan untuk menghapus perlindungan miring.
3. Di panel navigasi, pilih Pengaturan aplikasi, lalu pilih Pengaturan cabang.
4. Di bagian Cabang, pilih nama cabang yang akan diperbarui.
5. Pada menu Tindakan, pilih Nonaktifkan perlindungan miring. Perlindungan miring sekarang dinonaktifkan untuk cabang dan hanya konten terbaru yang akan disajikan.

Cara kerja perlindungan miring

Dalam kebanyakan kasus, perilaku default cookie _dpl akan melayani kebutuhan perlindungan miring Anda. Namun, dalam skenario lanjutan berikut, perlindungan miring lebih baik diaktifkan menggunakan parameter `X-Amplify-Dpl` header dan `dpl` kueri.

- Memuat situs web Anda di beberapa tab browser secara bersamaan
- Menggunakan pekerja layanan

Amplify mengevaluasi permintaan yang masuk dalam urutan berikut saat menentukan konten yang akan disajikan kepada klien:

1. **`X-Amplify-Dpl`**header — Aplikasi dapat menggunakan header ini untuk mengarahkan permintaan ke penerapan Amplify tertentu. Header permintaan ini dapat diatur dengan menggunakan nilai`process.env.AWS_AMPLIFY_DEPLOYMENT_ID`.
2. **`dpl`**parameter kueri - Aplikasi Next.js akan secara otomatis mengatur parameter kueri _dpl untuk permintaan ke aset sidik jari (file.js dan.css).
3. _dpl cookie — Ini adalah default untuk semua aplikasi yang dilindungi miring. Untuk browser tertentu, cookie yang sama dikirim untuk setiap tab browser atau instance yang berinteraksi dengan domain.

   Ketahuilah bahwa jika tab browser yang berbeda memiliki versi situs web yang berbeda yang dimuat, cookie _dpl dibagikan oleh semua tab. Dalam skenario ini, tidak mungkin untuk mencapai perlindungan kemiringan total dengan cookie _dpl dan Anda harus mempertimbangkan untuk menggunakan `X-Amplify-Dpl` header untuk perlindungan miring.

X-Amplify-Dpl Contoh header

Contoh berikut menunjukkan kode untuk halaman SSR Next.js yang mengakses perlindungan miring melalui header. `X-Amplify-Dpl` Halaman merender kontennya berdasarkan salah satu rute apinya. Penerapan untuk melayani ke rute api ditentukan dengan menggunakan `X-Amplify-Dpl` header, yang diatur ke nilai. `process.env.AWS_AMPLIFY_DEPLOYMENT_ID`

```
import { useEffect, useState } from 'react';

export default function MyPage({deploymentId}) {
    const [data, setData] = useState(null);

    useEffect(() => {
        fetch('/api/hello', {
            headers: {
                'X-Amplify-Dpl': process.env.AWS_AMPLIFY_DEPLOYMENT_ID
            },
        })
        .then(res => res.json())
        .then(data => setData(data))
        .catch(error => console.error("error", error))
    }, []);

    return <div>
        {data ? JSON.stringify(data) : "Loading ... " }
```

```
    </div>
}
```

Memantau aplikasi Amplify

AWS Amplify menyediakan fitur-fitur berikut untuk memantau aplikasi yang Anda host:

- CloudWatch metrik — Amplify memancarkan metrik melalui CloudWatch Amazon yang dapat Anda gunakan untuk memantau lalu lintas, kesalahan, transfer data, dan latensi untuk aplikasi Anda.
- Access logs — Amplify menyediakan log akses dengan informasi rinci tentang permintaan yang dibuat untuk aplikasi Anda.
- CloudTrail logging — Amplify terintegrasi dengan AWS CloudTrail yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amplify. Anda dapat melihat acara ini di CloudTrail konsol.

Topik

Memantau aplikasi Amplify dengan Amazon CloudWatch

AWS Amplify terintegrasi dengan Amazon CloudWatch, memungkinkan Anda memantau metrik untuk aplikasi Amplify dalam waktu dekat, dan membuat alarm yang mengirim notifikasi saat metrik melebihi ambang batas yang Anda tetapkan. Untuk informasi selengkapnya tentang cara kerja CloudWatch layanan, lihat Panduan CloudWatch Pengguna Amazon.

CloudWatch Metrik yang didukung

Amplify mendukung enam CloudWatch metrik di `AWS/AmplifyHosting` namespace untuk memantau lalu lintas, kesalahan, transfer data, dan latensi untuk aplikasi Anda. Metrik ini dikumpulkan pada interval satu menit. CloudWatch Metrik pemantauan tidak dipungut biaya dan tidak dihitung terhadap kuota CloudWatch layanan.

Tidak semua statistik berlaku untuk setiap metrik. Tabel berikut mencantumkan statistik yang paling relevan dengan deskripsi untuk setiap metrik yang didukung.

Metrik	Deskripsi
Permintaan	Jumlah total permintaan pemirsa yang diterima aplikasi Anda. Statistik paling relevan adalah `Sum`. Gunakan statistik `Sum` untuk mendapatkan jumlah total permintaan.
BytesDownloaded	Jumlah total data yang ditransfer dari aplikasi Anda (diunduh) dalam byte oleh pemirsa untuk permintaan `GET`, `HEAD`, dan `OPTIONS`. Statistik paling relevan adalah `Sum`.
BytesUploaded	Jumlah total data yang ditransfer ke aplikasi Anda (diunggah) dalam byte untuk permintaan apa pun, termasuk header. Amplify tidak mengenakan biaya untuk data yang diunggah di aplikasi Anda. Statistik paling relevan adalah `Sum`.
4xxErrors	Jumlah permintaan yang menampilkan pesan kesalahan dalam kisaran kode status HTTP 400-499. Statistik paling relevan adalah `Sum`. Gunakan statistik `Sum` untuk mendapatkan total kejadian kesalahan ini.
5xxErrors	Jumlah permintaan yang menampilkan pesan kesalahan dalam kisaran kode status HTTP 500-599. Statistik paling relevan adalah `Sum`. Gunakan statistik `Sum` untuk mendapatkan total kejadian kesalahan ini.

Metrik	Deskripsi
Latensi	Waktu ke byte pertama dalam detik. Ini adalah total waktu antara saat Amplify Hosting menerima permintaan dan ketika mengembal ikan respons ke jaringan. Waktu tidak termasuk latensi jaringan yang dialami untuk mengirimk an respons ke perangkat pemirsa. Statistik paling relevan adalah `Average`, `Maximum`, `Minimum`, `p10`, `p50`, `p90`, `p95`, dan `p100`. Gunakan statistik `Average` untuk mengevalu asi latensi yang diharapkan.

Amplify menyediakan dimensi CloudWatch metrik berikut.

Dimensi	Deskripsi
Aplikasi	Data metrik disediakan oleh aplikasi.
Akun AWS	Data metrik disediakan di semua aplikasi di file Akun AWS.

Mengakses metrik CloudWatch

Anda dapat mengakses CloudWatch metrik langsung dari konsol Amplify menggunakan prosedur berikut.

> **Note**
>
> Anda juga dapat mengakses CloudWatch metrik AWS Management Console di https://console.aws.amazon.com/cloudwatch/at.

Untuk mengakses metrik di konsol Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang metriknya ingin Anda lihat.
3. Di panel navigasi, pilih Pemantauan, lalu pilih Metrik.

Membuat CloudWatch alarm

Anda dapat membuat CloudWatch alarm di konsol Amplify yang mengirim notifikasi saat kriteria tertentu terpenuhi. Alarm mengawasi satu CloudWatch metrik dan mengirimkan pemberitahuan Amazon Simple Notification Service ketika metrik melanggar ambang batas untuk jumlah periode evaluasi tertentu.

Anda dapat membuat alarm tingkat lanjut yang menggunakan ekspresi matematika metrik di CloudWatch konsol atau menggunakan. CloudWatch APIs Misalnya, Anda dapat membuat alarm yang memberi tahu Anda ketika persentase 4xxErrors melebihi 15% selama tiga periode berturut-turut. Untuk informasi selengkapnya, lihat Membuat CloudWatch Alarm Berdasarkan Ekspresi Matematika Metrik di Panduan CloudWatch Pengguna Amazon.

CloudWatch Harga standar berlaku untuk alarm. Untuk informasi lebih lanjut, lihat harga Amazon CloudWatch.

Gunakan langkah-langkah berikut untuk membuat alarm di konsol Amplify.

Untuk membuat CloudWatch alarm untuk metrik Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi tempat alarm akan disetel.
3. Di panel navigasi, pilih Pemantauan, lalu pilih Alarm.
4. Pada halaman Alarm, pilih Buat alarm.
5. Di jendela Buat alarm, konfigurasikan alarm sebagai berikut:

    a. Untuk Metrik, pilih nama metrik yang akan dipantau dari daftar.

    b. Untuk Nama alarm, masukkan nama yang sesuai untuk alarm. Misalnya, jika memantau Permintaan, Anda dapat memberikan nama alarm **`HighTraffic`**. Nama harus memiliki karakter ASCII saja.

    c. Untuk Atur notifikasi, lakukan salah satu langkah berikut:

- i. Pilih Baru untuk mengatur topik Amazon SNS baru.
  ii. Untuk Alamat email, masukkan alamat email penerima notifikasi.
  iii. Pilih Tambahkan alamat email baru untuk menambahkan penerima lain.
- i. Pilih Yang ada untuk menggunakan kembali topik Amazon SNS.
  ii. Untuk Topik SNS, pilih nama topik Amazon SNS yang ada dari daftar.

d. Untuk Kapan Saja Statistik Metrik, tetapkan syarat berikut untuk alarm Anda:

i. Tentukan apakah nilai metrik lebih besar dari, kurang dari, atau sama dengan nilai ambang batas.
ii. Tentukan nilai ambang batas.
iii. Tentukan jumlah periode evaluasi berturut-turut yang harus dalam keadaan alarm untuk memanggil alarm.
iv. Tentukan durasi periode evaluasi.

e. Pilih Konfirmasi.

> **Note**
>
> Setiap penerima Amazon SNS yang Anda tentukan akan menerima email konfirmasi dari Notifikasi AWS . Email ini berisi tautan yang harus diikuti oleh penerima untuk mengonfirmasi langganan dan menerima notifikasi.

Mengakses CloudWatch Log untuk aplikasi SSR

Amplify mengirimkan informasi tentang runtime SSR Anda ke CloudWatch Amazon Logs di Anda. Akun AWS Saat Anda menerapkan aplikasi SSR ke komputasi Amplify Hosting, aplikasi memerlukan peran layanan IAM yang diasumsikan Amplify saat memanggil layanan lain atas nama Anda. Anda dapat mengizinkan komputasi Amplify Hosting untuk secara otomatis membuat peran layanan untuk Anda atau Anda dapat menentukan peran yang telah Anda buat.

Jika Anda memilih untuk mengizinkan Amplify membuat peran IAM untuk Anda, peran tersebut sudah memiliki izin untuk membuat Log. CloudWatch Jika membuat peran IAM sendiri, Anda perlu menambahkan izin berikut ke kebijakan agar Amplify dapat mengakses Log Amazon. CloudWatch

```
logs:CreateLogStream
```

```
logs:CreateLogGroup
logs:DescribeLogGroups
logs:PutLogEvents
```

Untuk informasi selengkapnya tentang menambahkan peran layanan, lihatMenambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend. Untuk informasi selengkapnya tentang penerapan aplikasi yang dirender sisi server, lihat. Menyebarkan aplikasi yang dirender sisi server dengan Amplify Hosting

Anda dapat melihat log komputasi Amplify Hosting untuk aplikasi SSR di konsol atau di CloudWatch konsol Amplify. Gunakan petunjuk berikut untuk melihat log di konsol Amplify.

Untuk melihat CloudWatch log untuk aplikasi SSR di konsol Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi SSR untuk melihat CloudWatch log.
3. Di panel navigasi, pilih Monitoring, lalu pilih Hosting compute logs.
4. Pada halaman log komputasi Hosting, cari dan pilih grup CloudWatch log untuk cabang tertentu.

Mengambil dan menganalisis log akses untuk aplikasi Amplify

Amplify menyimpan log akses untuk semua aplikasi yang Anda host di Amplify. Log akses berisi informasi tentang permintaan yang dibuat ke aplikasi yang dihosting. Amplify menyimpan semua log akses untuk aplikasi hingga Anda menghapus aplikasi. Semua log akses untuk aplikasi tersedia di konsol Amplify. Namun, setiap permintaan individu untuk log akses dibatasi untuk periode waktu dua minggu yang Anda tentukan.

> **Warning**
>
> Jangan sertakan rahasia, kredensil, atau data sensitif URLs sebagai parameter jalur atau kueri. Nilai-nilai ini dapat dilihat dalam teks biasa di log akses aplikasi Amplify Anda.

Amplify tidak pernah menggunakan kembali CloudFront distribusi antar pelanggan. Amplify membuat CloudFront distribusi terlebih dahulu sehingga Anda tidak perlu menunggu CloudFront distribusi dibuat saat menerapkan aplikasi baru. Sebelum distribusi ini ditetapkan ke aplikasi Amplify, mereka mungkin menerima lalu lintas dari bot. Namun, mereka dikonfigurasi untuk selalu merespons sebagai

Tidak ditemukan sebelum ditetapkan. Jika log akses aplikasi berisi entri untuk jangka waktu sebelum membuat aplikasi, entri ini terkait dengan aktivitas ini.

> **Important**
>
> Kami menyarankan agar Anda menggunakan log untuk memahami sifat permintaan konten Anda, bukan sebagai akuntasi lengkap atas semua permintaan. Amplify memberikan log akses dengan upaya terbaik. Entri log untuk permintaan tertentu mungkin dikirim dalam waktu lama setelah permintaan diproses secara aktual dan, dalam kasus yang jarang, entri log mungkin tidak dikirimkan sama sekali. Ketika entri log dihilangkan dari log akses, jumlah entri dalam log akses tidak akan cocok dengan penggunaan yang muncul dalam laporan AWS penagihan dan penggunaan.

Mengambil log akses aplikasi

Gunakan prosedur berikut untuk mengambil log akses untuk aplikasi Amplify.

Untuk melihat log akses

1. Masuk ke AWS Management Console dan buka konsol Amplify.
2. Pilih aplikasi yang log aksesnya akan Anda lihat.
3. Di panel navigasi, pilih Monitoring, lalu pilih Access logs.
4. Pilih Edit rentang waktu.
5. Di jendela Edit rentang waktu lakukan hal berikut.
   a. Untuk tanggal Mulai, tentukan hari pertama dari interval dua minggu untuk mengambil log.
   b. Untuk Waktu mulai, pilih waktu pada hari pertama pengambilan log dimulai.
   c. Pilih Konfirmasi.
6. Konsol Amplify menampilkan log untuk rentang waktu yang ditentukan di bagian Access logs. Pilih Unduh untuk menyimpan log dalam format CSV.

Menganalisis log akses

Untuk menganalisis log akses, Anda dapat menyimpan file CSV di bucket Amazon S3. Salah satu cara untuk menganalisis log akses Anda adalah dengan menggunakan Athena. Athena adalah

layanan kueri interaktif yang dapat membantu Anda menganalisis data untuk AWS layanan. Anda dapat mengikuti step-by-step instruksi di sini untuk membuat tabel. Setelah tabel dibuat, Anda dapat membuat kueri data sebagai berikut.

```
SELECT SUM(bytes) AS total_bytes
FROM logs
WHERE "date" BETWEEN DATE '2018-06-09' AND DATE '2018-06-11'
LIMIT 100;
```

Mencatat log panggilan API Amplify dengan AWS CloudTrail

AWS Amplify terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amplify. CloudTrail menangkap semua panggilan API untuk Amplify sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari konsol Amplify dan panggilan kode ke operasi API Amplify. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail peristiwa secara berkelanjutan ke bucket Amazon S3, termasuk peristiwa untuk Amplify. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang CloudTrail dikumpulkan, Anda dapat menentukan permintaan yang dibuat untuk Amplify, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Amplify informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda secara default. Ketika aktivitas terjadi di Amplify, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS . Untuk informasi selengkapnya, lihat Melihat CloudTrail peristiwa dengan riwayat peristiwa di Panduan AWS CloudTrail Pengguna.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Amplify, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah AWS . Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, pelajari topik berikut di Panduan Pengguna AWS CloudTrail :

- Membuat jejak untuk AWS akun Anda
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua operasi Amplify dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API AWS Amplify Konsol, Referensi API UI AWS Amplify Admin, dan Referensi API Amplify UI Builder. Misalnya, panggilan ke`CreateApp`, `DeleteApp` dan `DeleteBackendEnvironment` operasi menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan dibuat dengan kredensial pengguna root atau AWS Identity and Access Management (IAM).
- Permintaan dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Apakah permintaan dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat elemen CloudTrail UserIdentity di AWS CloudTrail Panduan Pengguna.

Memahami entri berkas log Amplify

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `ListApps`operasi Referensi API AWS Amplify Konsol.

```
{
    "eventVersion": "1.08",
```

```
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::444455556666:user/Mary_Major",
        "accountId": "444455556666",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Mary_Major",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-01-12T05:48:10Z"
            }
        }
    },
    "eventTime": "2021-01-12T06:47:29Z",
    "eventSource": "amplify.amazonaws.com",
    "eventName": "ListApps",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.898
 Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.275-b01
 java/1.8.0_275 vendor/Oracle_Corporation",
    "requestParameters": {
        "maxResults": "100"
    },
    "responseElements": null,
    "requestID": "1c026d0b-3397-405a-95aa-aa43aexample",
    "eventID": "c5fca3fb-d148-4fa1-ba22-5fa63example",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "444455556666"
}
```


Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `ListBackendJobs`operasi Referensi API UI AWS Amplify Admin.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::444455556666:user/Mary_Major",
        "accountId": "444455556666",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Mary_Major",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-01-13T00:47:25Z"
            }
        }
    },
    "eventTime": "2021-01-13T01:15:43Z",
    "eventSource": "amplifybackend.amazonaws.com",
    "eventName": "ListBackendJobs",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.898
Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.275-b01
java/1.8.0_275 vendor/Oracle_Corporation",
    "requestParameters": {
        "appId": "d23mv2oexample",
        "backendEnvironmentName": "staging"
    },
    "responseElements": {
        "jobs": [
            {
                "appId": "d23mv2oexample",
                "backendEnvironmentName": "staging",
                "jobId": "ed63e9b2-dd1b-4bf2-895b-3d5dcexample",
                "operation": "CreateBackendAuth",
                "status": "COMPLETED",
                "createTime": "1610499932490",
                "updateTime": "1610500140053"
            },
            {
                "appId": "d23mv2oexample",
                "backendEnvironmentName": "staging",
                "jobId": "06904b10-a795-49c1-92b7-185dfexample",
                "operation": "CreateBackend",
                "status": "COMPLETED",
```

```
                "createTime": "1610499657938",
                "updateTime": "1610499704458"
            }
        ],
        "appId": "d23mv2oexample",
        "backendEnvironmentName": "staging"
    },
    "requestID": "7adfabd6-98d5-4b11-bd39-c7deaexample",
    "eventID": "68769310-c96c-4789-a6bb-68b52example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "444455556666"
}
```

Menggunakan peran IAM dengan aplikasi Amplify

Peran IAM adalah identitas IAM dengan izin tertentu. Izin peran menentukan apa yang dapat dan tidak dapat dilakukan identitas. AWS Anda dapat membuat peran IAM di dalam Akun AWS dan menggunakannya untuk mendelegasikan izin ke Amplify Hosting. Untuk mempelajari lebih lanjut tentang peran, lihat peran IAM di Panduan Pengguna IAM.

Anda dapat menggunakan jenis peran IAM berikut ini untuk memberikan Amplify Hosting izin yang diperlukan untuk melakukan tindakan atas nama Anda atau menjalankan kode komputasi yang mengakses sumber daya lain. AWS

Peran layanan IAM

Amplify mengasumsikan peran ini untuk melakukan tindakan atas nama Anda. Peran ini diperlukan untuk aplikasi dengan sumber daya backend.

IAM SSR Peran komputasi

Memungkinkan aplikasi yang dirender sisi server (SSR) untuk mengakses sumber daya tertentu dengan aman. AWS

Peran IAM SSR Log CloudWatch

Saat Anda menerapkan aplikasi SSR, aplikasi memerlukan peran layanan IAM yang diasumsikan Amplify agar Amplify dapat mengakses Log Amazon. CloudWatch

Topik

- Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend
- Menambahkan peran SSR Compute untuk memungkinkan akses ke sumber daya AWS
- Menambahkan peran layanan dengan izin untuk mengakses CloudWatch Log

Menambahkan peran layanan dengan izin untuk menyebarkan sumber daya backend

Amplify memerlukan izin untuk menerapkan sumber daya backend dengan front end Anda. Anda menggunakan peran layanan untuk mendapatkan izin tersebut. Peran layanan adalah peran AWS

Identity and Access Management (IAM) yang menyediakan Amplify Hosting dengan izin untuk menyebarkan, membuat, dan mengelola backend atas nama Anda.

Saat membuat aplikasi baru yang memerlukan peran layanan IAM, Anda dapat mengizinkan Amplify Hosting untuk secara otomatis membuat peran layanan untuk Anda atau Anda dapat memilih peran IAM yang telah Anda buat. Di bagian ini, Anda akan mempelajari cara membuat peran layanan Amplify yang memiliki izin administratif akun dan eksplisit memungkinkan akses langsung ke sumber daya yang diperlukan aplikasi Amplify untuk menyebarkan, membuat, dan mengelola backend.

Membuat peran layanan Amplify di konsol IAM

Membuat peran layanan

1. Buka konsol IAM dan pilih Peran dari bilah navigasi kiri, lalu pilih Buat peran.
2. Pada halaman Pilih entitas tepercaya, pilih AWS layanan. Untuk kasus Penggunaan, pilih Amplify - Deployment Backend, lalu pilih Berikutnya.
3. Pada halaman Tambahkan izin, pilih Berikutnya.
4. Pada halaman Nama, tampilan, dan buat, untuk nama Peran masukkan nama yang bermakna, seperti**`AmplifyConsoleServiceRole-AmplifyRole`**.
5. Terima semua default dan pilih Buat peran.
6. Kembali ke konsol Amplify untuk melampirkan peran ke aplikasi Anda.
   - Jika Anda sedang dalam proses menerapkan aplikasi baru, lakukan hal berikut:
     a. Segarkan daftar peran layanan.
     b. Pilih peran yang baru saja Anda buat. Untuk contoh ini, seharusnya terlihat seperti AmplifyConsoleServiceRole- AmplifyRole.
     c. Pilih Berikutnya dan ikuti langkah-langkah untuk menyelesaikan penerapan aplikasi Anda.
   - Jika Anda memiliki aplikasi yang sudah ada, lakukan hal berikut:
     a. Di panel navigasi, pilih Pengaturan aplikasi, lalu pilih peran IAM.
     b. Pada halaman peran IAM, di bagian Peran layanan, pilih Edit.
     c. Pada halaman peran Layanan, pilih peran yang baru saja Anda buat dari daftar peran Layanan.
     d. Pilih Simpan.
7. Amplify sekarang memiliki izin untuk menerapkan resource backend untuk aplikasi Anda.

Mengedit kebijakan kepercayaan peran layanan untuk mencegah wakil yang bingung

Masalah "confused deputy" adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang memilik hak akses lebih tinggi untuk melakukan tindakan tersebut. Untuk informasi selengkapnya, lihat Pencegahan "confused deputy" lintas layanan.

Saat ini, kebijakan kepercayaan default untuk peran `Amplify-Backend Deployment` layanan memberlakukan kunci kondisi konteks `aws:SourceAccount` global `aws:SourceArn` dan untuk mencegah deputi yang bingung. Namun, jika sebelumnya Anda membuat `Amplify-Backend Deployment` peran di akun Anda, Anda dapat memperbarui kebijakan kepercayaan peran untuk menambahkan kondisi ini untuk melindungi dari wakil yang bingung.

Gunakan contoh berikut untuk membatasi akses ke aplikasi di akun Anda. Ganti Wilayah dan ID aplikasi dalam contoh dengan informasi Anda sendiri.

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
```

Untuk petunjuk cara mengedit kebijakan kepercayaan untuk peran yang menggunakan AWS Management Console, lihat Memodifikasi peran (konsol) di Panduan Pengguna IAM.

Menambahkan peran SSR Compute untuk memungkinkan akses ke sumber daya AWS

Integrasi ini, memungkinkan Anda menetapkan peran IAM ke layanan Amplify SSR Compute agar aplikasi yang dirender sisi server (SSR) Anda dapat mengakses sumber daya tertentu secara aman berdasarkan izin peran. AWS Misalnya, Anda dapat mengizinkan fungsi komputasi SSR aplikasi Anda mengakses AWS layanan atau sumber daya lain dengan aman, seperti Amazon Bedrock atau bucket Amazon S3, berdasarkan izin yang ditentukan dalam peran IAM yang ditetapkan.

Peran IAM SSR Compute menyediakan kredensil sementara, menghilangkan kebutuhan untuk hardcode kredenal keamanan berumur panjang dalam variabel lingkungan. Menggunakan peran Komputasi SSR IAM sejalan dengan praktik terbaik AWS keamanan dalam memberikan izin hak istimewa paling sedikit dan menggunakan kredensil jangka pendek jika memungkinkan.

Instruksi nanti di bagian ini menjelaskan cara membuat kebijakan dengan izin khusus dan melampirkan kebijakan ke peran. Saat membuat peran, Anda harus melampirkan kebijakan kepercayaan khusus yang memberikan izin Amplify untuk mengambil peran tersebut. Jika hubungan kepercayaan tidak didefinisikan dengan benar, Anda akan mendapatkan kesalahan saat mencoba menambahkan peran. Kebijakan kepercayaan khusus berikut memberikan izin Amplify untuk mengambil peran.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Anda dapat mengaitkan peran IAM Akun AWS dengan aplikasi SSR yang ada menggunakan konsol Amplify,, AWS SDKs atau. AWS CLI Peran yang Anda lampirkan secara otomatis dikaitkan dengan layanan komputasi Amplify SSR, memberikan izin yang Anda tentukan untuk mengakses sumber daya lain. AWS Karena kebutuhan aplikasi Anda berubah dari waktu ke waktu, Anda dapat memodifikasi peran IAM terlampir tanpa memindahkan aplikasi Anda. Ini memberikan fleksibilitas dan mengurangi downtime aplikasi.

> **Important**
>
> Anda bertanggung jawab untuk mengonfigurasi aplikasi Anda untuk memenuhi tujuan keamanan dan kepatuhan Anda. Ini termasuk mengelola peran Komputasi SSR Anda, yang harus dikonfigurasi untuk memiliki set izin minimum yang diperlukan untuk mendukung

kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Mengelola keamanan peran Komputasi SSR IAM.

Membuat peran Komputasi SSR di konsol IAM

Sebelum Anda dapat melampirkan peran Komputasi SSR IAM ke aplikasi Amplify, peran tersebut harus sudah ada di aplikasi Anda. Akun AWS Di bagian ini, Anda akan mempelajari cara membuat kebijakan IAM dan melampirkannya ke peran yang dapat diasumsikan Amplify untuk mengakses AWS sumber daya tertentu.

Kami menyarankan Anda mengikuti praktik AWS terbaik pemberian izin hak istimewa paling sedikit saat membuat peran IAM. Peran IAM SSR Compute dipanggil hanya dari fungsi komputasi SSR dan oleh karena itu seharusnya hanya memberikan izin yang diperlukan untuk menjalankan kode.

Anda dapat menggunakan AWS Management Console, AWS CLI, atau SDKs untuk membuat kebijakan di IAM. Untuk interformasi selengkapnya, lihat Menentukan izin IAM khusus dengan kebijakan yang dikelola pelanggan di Panduan Pengguna IAM.

Petunjuk berikut menunjukkan cara menggunakan konsol IAM untuk membuat kebijakan IAM yang menentukan izin yang akan diberikan ke layanan Amplify Compute.

Untuk menggunakan editor kebijakan JSON konsol IAM untuk membuat kebijakan

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di bagian Editor kebijakan, pilih opsi JSON.
5. Ketik atau tempel dokumen kebijakan JSON.
6. Setelah selesai menambahkan izin ke kebijakan, pilih Berikutnya.
7. Pada halaman Tinjau dan buat, ketik Nama Kebijakan dan Deskripsi (opsional) untuk kebijakan yang Anda buat. Tinjau Izin yang ditentukan dalam kebijakan ini untuk melihat izin yang diberikan oleh kebijakan Anda.
8. Pilih Buat kebijakan untuk menyimpan kebijakan baru Anda.

Setelah Anda membuat kebijakan, gunakan petunjuk berikut untuk melampirkan kebijakan ke peran IAM.

Untuk membuat peran yang memberikan izin Amplify ke sumber daya tertentu AWS

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi konsol, pilih Peran dan kemudian pilih Buat peran.
3. Pilih jenis peran kebijakan kepercayaan kustom.
4. Di bagian Kebijakan kepercayaan khusus, masukkan kebijakan kepercayaan khusus untuk peran tersebut. Kebijakan kepercayaan peran diperlukan dan mendefinisikan prinsip-prinsip yang Anda percayai untuk mengambil peran tersebut.

   Salin dan tempel kebijakan kepercayaan berikut untuk memberikan izin layanan Amplify untuk mengambil peran ini.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "Statement1",
               "Effect": "Allow",
               "Principal": {
                   "Service": [
                       "amplify.amazonaws.com"
                   ]
               },
               "Action": "sts:AssumeRole"
           }
       ]
   }
   ```

5. Selesaikan peringatan keamanan, kesalahan, atau peringatan umum yang dihasilkan selama validasi kebijakan, lalu pilih Berikutnya.
6. Pada halaman Tambahkan izin, cari nama kebijakan yang Anda buat di prosedur sebelumnya dan pilih. Lalu pilih Berikutnya.
7. Untuk Nama peran, masukkan nama peran. Nama peran harus unik di dalam diri Anda Akun AWS. Grup tidak dibedakan berdasarkan huruf besar-kecil. Misalnya, Anda tidak dapat membuat

peran dengan nama **`PRODROLE`** dan **`prodrole`**. Karena AWS sumber daya lain mungkin mereferensikan peran, Anda tidak dapat mengedit nama peran setelah dibuat.

8. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk peran baru ini.
9. (Opsional) Pilih Edit di Langkah 1: Pilih entitas tepercaya atau Langkah 2: Tambahkan izin bagian untuk mengedit kebijakan kustom dan izin untuk peran tersebut.
10. Tinjau peran, lalu pilih Buat peran.

Menambahkan peran Komputasi SSR IAM ke aplikasi Amplify

Setelah membuat peran IAM Akun AWS, Anda dapat mengaitkannya dengan aplikasi di konsol Amplify.

Untuk menambahkan peran Komputasi SSR ke aplikasi di konsol Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Di halaman Semua aplikasi, pilih nama aplikasi untuk menambahkan peran Komputasi.
3. Di panel navigasi, pilih Pengaturan aplikasi, lalu pilih peran IAM.
4. Di bagian Peran komputasi, pilih Edit.
5. Dalam daftar peran default, cari nama peran yang ingin dilampirkan dan pilih. Untuk contoh ini, Anda dapat memilih nama peran yang Anda buat di prosedur sebelumnya. Secara default, peran yang Anda pilih akan dikaitkan dengan semua cabang aplikasi Anda.

   Jika hubungan kepercayaan peran tidak didefinisikan dengan benar, Anda akan mendapatkan kesalahan dan Anda tidak akan dapat menambahkan peran tersebut.
6. (opsional) Jika aplikasi Anda berada dalam repositori publik dan menggunakan pembuatan cabang otomatis atau pratinjau web untuk permintaan tarik diaktifkan, kami tidak menyarankan untuk menggunakan peran tingkat aplikasi. Sebagai gantinya, lampirkan peran Compute hanya ke cabang yang memerlukan akses ke sumber daya tertentu. Untuk mengganti perilaku tingkat aplikasi default dan melampirkan peran ke cabang tertentu, lakukan hal berikut:

   a. Untuk Branch, pilih nama cabang yang akan digunakan.

   b. Untuk peran Compute, pilih nama peran yang akan diasosiasikan dengan cabang.
7. Pilih, Simpan.

Mengelola keamanan peran Komputasi SSR IAM

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Anda bertanggung jawab untuk mengonfigurasi aplikasi Anda untuk memenuhi tujuan keamanan dan kepatuhan Anda. Ini termasuk mengelola peran Komputasi SSR Anda, yang harus dikonfigurasi untuk memiliki set izin minimum yang diperlukan untuk mendukung kasus penggunaan Anda. Kredensil untuk peran Komputasi SSR yang Anda tentukan segera tersedia di runtime fungsi SSR Anda. Jika kode SSR Anda mengekspos kredensil ini, baik secara sengaja, karena bug, atau dengan mengizinkan eksekusi kode jarak jauh (RCE), pengguna yang tidak sah dapat memperoleh akses ke peran SSR dan izinnya.

Saat aplikasi di repositori publik menggunakan peran Komputasi SSR dan pembuatan cabang otomatis atau pratinjau web untuk permintaan tarik, Anda perlu mengelola cabang mana yang dapat mengakses peran dengan hati-hati. Kami menyarankan Anda untuk tidak menggunakan peran tingkat aplikasi. Sebagai gantinya, Anda harus melampirkan peran Compute di tingkat cabang. Ini memungkinkan Anda untuk memberikan izin hanya ke cabang yang memerlukan akses ke sumber daya tertentu.

Jika kredenal peran Anda terekspos, lakukan tindakan berikut untuk menghapus semua akses ke kredenal peran.

1. Cabut semua sesi

   Untuk petunjuk tentang segera mencabut semua izin ke kredenal peran, lihat Mencabut kredenal keamanan sementara peran IAM.
2. Hapus peran dari konsol Amplify

   Tindakan ini segera berlaku. Anda tidak perlu menerapkan ulang aplikasi Anda.

Untuk menghapus peran Compute di konsol Amplify

1. Masuk ke AWS Management Console dan buka konsol Amplify di. https://console.aws.amazon.com/amplify/
2. Di halaman Semua aplikasi, pilih nama aplikasi untuk menghapus peran Compute.
3. Di panel navigasi, pilih Pengaturan aplikasi, lalu pilih peran IAM.
4. Di bagian Peran komputasi, pilih Edit.
5. Untuk menghapus peran Default, pilih X di sebelah kanan nama peran.

6. Pilih Simpan.

Menambahkan peran layanan dengan izin untuk mengakses CloudWatch Log

Amplify mengirimkan informasi tentang runtime SSR Anda ke CloudWatch Amazon Logs di Anda. Akun AWS Saat Anda menerapkan aplikasi SSR, aplikasi memerlukan peran layanan IAM yang diasumsikan Amplify saat memanggil layanan lain atas nama Anda. Anda dapat mengizinkan komputasi Amplify Hosting untuk secara otomatis membuat peran layanan untuk Anda atau Anda dapat menentukan peran yang telah Anda buat.

Jika Anda memilih untuk mengizinkan Amplify membuat peran IAM untuk Anda, peran tersebut sudah memiliki izin untuk membuat Log. CloudWatch Jika membuat peran IAM sendiri, Anda perlu menambahkan izin berikut ke kebijakan agar Amplify dapat mengakses Log Amazon. CloudWatch

```
logs:CreateLogStream
logs:CreateLogGroup
logs:DescribeLogGroups
logs:PutLogEvents
```

Webhook terpadu untuk repositori Git

Amplify Hosting menggunakan webhook untuk memulai build secara otomatis setelah komit baru ke repositori Git Anda. Fitur webhook terpadu meningkatkan integrasi Amplify dengan penyedia Git dan memungkinkan Anda menghubungkan lebih banyak aplikasi Amplify ke satu repositori. Dengan webhook terpadu, Amplify sekarang menggunakan satu webhook per Wilayah untuk semua aplikasi terkait di repositori Anda. Misalnya, jika repositori Anda terhubung ke aplikasi di Wilayah US East (N. Virginia) dan US West (Oregon), Anda akan memiliki dua webhook terpadu.

Sebelum rilis ini, Amplify membuat webhook baru untuk setiap aplikasi yang terkait dengan repositori. Jika Anda memiliki beberapa aplikasi dalam satu repositori, Anda dapat mencapai batas webhook yang diberlakukan oleh penyedia Git individual dan dicegah untuk menambahkan lebih banyak aplikasi. Ini sangat menantang bagi tim yang bekerja di monorepos, di mana beberapa proyek ada dalam satu repositori.

Webhook terpadu memberikan manfaat sebagai berikut:

- Atasi batas webhook penyedia Git: Anda dapat menghubungkan sebanyak mungkin aplikasi Amplify yang Anda butuhkan ke satu repositori.
- Dukungan monorepo yang ditingkatkan: Anda memiliki lebih banyak fleksibilitas dan efisiensi saat bekerja dengan monorepos, di mana beberapa proyek berbagi satu repositori.
- Manajemen yang disederhanakan: Mengelola beberapa aplikasi Amplify dengan satu repositori webhook mengurangi kompleksitas dan potensi titik kegagalan.
- Integrasi alur kerja yang ditingkatkan: Anda dapat menggunakan webhook yang dialokasikan oleh penyedia Git Anda untuk alur kerja penting lainnya dalam proses pengembangan Anda.

Memulai dengan webhook terpadu

Membuat aplikasi baru

Saat Anda menerapkan aplikasi baru ke Amplify Hosting dari repositori Git, fitur webhooks terpadu secara otomatis diimplementasikan untuk repositori Anda. Untuk petunjuk tentang membuat aplikasi baru, lihatMemulai dengan menerapkan aplikasi ke Amplify Hosting.

Memperbarui aplikasi yang ada

Untuk aplikasi Amplify yang ada, Anda harus menghubungkan kembali repositori Git Anda ke aplikasi Anda untuk mengganti webhook yang ada dengan webhook terpadu. Jika Anda telah mencapai

jumlah maksimum webhook yang diizinkan oleh penyedia Git Anda, migrasi ke webhook terpadu mungkin tidak berhasil. Dalam hal ini, hapus setidaknya satu webhook yang ada secara manual sebelum menghubungkan kembali.

Anda dapat memiliki beberapa aplikasi dalam repositori yang digunakan ke Wilayah yang berbeda. AWS Karena operasi Amplify berbasis Wilayah, migrasi ke webhook terpadu hanya terjadi untuk webhook di Wilayah tempat Anda menghubungkan kembali aplikasi Amplify. Akibatnya, Anda mungkin melihat webhook berbasis id aplikasi dan webhook terpadu berbasis Region di repositori Anda.

Gunakan petunjuk berikut untuk memigrasikan aplikasi Amplify yang ada ke webhook terpadu.

Untuk memigrasikan aplikasi Amplify yang ada ke webhook terpadu

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang akan dimigrasikan ke webhook terpadu.
3. Di panel navigasi, pilih Pengaturan aplikasi, lalu pilih Pengaturan cabang.
4. Pada halaman Pengaturan cabang, pilih Hubungkan kembali repositori.
5. Untuk memverifikasi migrasi yang berhasil ke webhook terpadu, navigasikan ke pengaturan webhook di repositori Git Anda. Anda akan melihat URL webhook tunggal dalam format`https://amplify-webhooks.`*`Region`*`.amazonaws.com/`*`git-provider`*.

Keamanan di Amplify

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku AWS Amplify, lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amplify. Topik-topik berikut menjelaskan cara mengonfigurasi Amplify untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amplify Anda.

Topik

Identity and Access Management untuk Amplify

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memperoleh izin) untuk menggunakan sumber daya Amplify. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amplify.

Pengguna layanan – Jika Anda menggunakan layanan Amplify untuk melakukan tugas, administrator Anda akan memberikan kredensial dan izin yang dibutuhkan. Saat menggunakan lebih banyak fitur Amplify untuk melakukan pekerjaan, Anda mungkin memerlukan izin tambahan. Memahami cara pengelolaan akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amplify, lihat Pemecahan masalah identitas dan akses Amplify.

Administrator layanan – Jika bertanggung jawab atas sumber daya Amplify di perusahaan Anda, Anda mungkin memiliki akses penuh ke Amplify. Tugas Anda adalah menentukan fitur dan sumber daya Amplify mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang cara perusahaan Anda dapat menggunakan IAM dengan Amplify, lihat Cara Amplify bekerja dengan IAM.

Administrator IAM – Jika Anda adalah administrator IAM, Anda perlu mempelajari dengan mendetail cara menulis kebijakan untuk mengelola akses ke Amplify. Untuk melihat contoh kebijakan berbasis

identitas Amplify yang dapat Anda gunakan di IAM, lihat Contoh kebijakan berbasis identitas untuk Amplify.

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensil yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut

untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna

memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.

- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.
- Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus

menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.
- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa

memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Cara Amplify bekerja dengan IAM

Sebelum menggunakan IAM untuk mengelola akses ke Amplify, Anda harus memahami berbagai fitur IAM yang mendukung Amplify.

Berbagai fitur IAM yang mendukung Amplify

Fitur IAM	Dukungan Amplify
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya

Fitur IAM	Dukungan Amplify
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Sesi akses teruskan (FAS)	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara Amplify dan layanan AWS lainnya bekerja dengan sebagian besar fitur IAM, AWS lihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Amplify

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan IAM JSON dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amplify

Untuk melihat contoh kebijakan berbasis identitas Amplify, lihat Contoh kebijakan berbasis identitas untuk Amplify.

Kebijakan berbasis sumber daya dalam Amplify

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke principal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Amplify

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk daftar tindakan Amplify, lihat Tindakan yang ditentukan AWS Amplify di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amplify menggunakan prefiks berikut sebelum tindakan:

```
amplify
```

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan setiap tindakan dengan koma.

```
"Action": [
      "amplify:action1",
      "amplify:action2"
         ]
```

Untuk melihat contoh kebijakan berbasis identitas Amplify, lihat Contoh kebijakan berbasis identitas untuk Amplify.

Sumber daya kebijakan untuk Amplify

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk daftar jenis sumber daya Amplify dan jenisnya ARNs, lihat Jenis sumber daya yang ditentukan oleh AWS Amplify dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat menentukan ARN setiap sumber daya, lihat Tindakan yang ditentukan AWS Amplify.

Untuk melihat contoh kebijakan berbasis identitas Amplify, lihat Contoh kebijakan berbasis identitas untuk Amplify.

Kunci syarat kebijakan untuk Amplify

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk daftar kunci syarat Amplify, lihat Kunci syarat yang ditentukan AWS Amplify di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh AWS Amplify.

Untuk melihat contoh kebijakan berbasis identitas Amplify, lihat Contoh kebijakan berbasis identitas untuk Amplify.

Daftar kontrol akses (ACLs) di Amplify

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan Amplify

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial berisi langkah-langkah untuk mengatur ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) di Panduan Pengguna IAM.

Menggunakan kredensial sementara dengan Amplify

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensil sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensil sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensil sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensil sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Teruskan sesi akses untuk Amplify

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

Peran layanan untuk Amplify

Mendukung peran layanan: Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan tertentu dapat merusak fungsionalitas Amplify. Edit peran layanan hanya jika Amplify menyediakan panduan untuk melakukannya.

Peran yang tertaut dengan layanan untuk Amplify

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut-layanan.

Untuk informasi selengkapnya tentang cara membuat atau mengelola peran yang tertaut dengan layanan, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM. Cari layanan dalam tabel yang mencakup `Yes` di kolom Peran yang tertaut dengan layanan. Pilih tautan Ya untuk melihat dokumentasi peran yang tertaut dengan layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk Amplify

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amplify. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amplify, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi AWS Amplify di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol Amplify

- Mengizinkan pengguna melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amplify di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.

- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan konsol Amplify

Untuk mengakses AWS Amplify konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amplify di Anda. Akun AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Dengan dirilisnya Amplify Studio, menghapus aplikasi atau backend memerlukan keduanya dan izin. `amplify` `amplifybackend` Jika kebijakan IAM hanya menyediakan izin `amplify`, pesan kesalahan izin akan muncul saat pengguna mencoba menghapus aplikasi. Jika Anda adalah administrator yang menulis kebijakan, tentukan izin yang benar untuk diberikan kepada pengguna yang perlu melakukan tindakan penghapusan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Amplify, lampirkan juga kebijakan Amplify `ConsoleAccess` atau `ReadOnly` AWS managed ke entitas. Untuk informasi selengkapnya, lihat Menambah izin untuk pengguna dalam Panduan Pengguna IAM.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS kebijakan terkelola untuk AWS Amplify

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan.

AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

Kebijakan terkelola AWS: AdministratorAccess -Amplify

Anda dapat melampirkan kebijakan `AdministratorAccess-Amplify` ke identitas IAM Anda. Amplify juga melampirkan kebijakan ini ke peran layanan yang memungkinkan Amplify melakukan tindakan atas nama Anda.

Saat menerapkan backend di konsol Amplify, Anda harus membuat `Amplify-Backend Deployment` peran layanan yang Amplify gunakan untuk membuat dan mengelola sumber daya. AWS IAM melampirkan kebijakan `AdministratorAccess-Amplify` terkelola ke peran `Amplify-Backend Deployment` layanan.

Kebijakan ini memberikan izin administratif akun sementara secara eksplisit mengizinkan akses langsung ke sumber daya yang diperlukan aplikasi Amplify untuk membuat dan mengelola backend.

Detail izin

Kebijakan ini menyediakan akses ke beberapa AWS layanan, termasuk tindakan IAM. Tindakan ini memungkinkan identitas dengan kebijakan ini digunakan AWS Identity and Access Management untuk membuat identitas lain dengan izin apa pun. Hal ini memungkinkan eskalasi izin dan kebijakan ini harus dianggap sekuat kebijakan. `AdministratorAccess`

Kebijakan ini memberikan izin `iam:PassRole` tindakan untuk semua sumber daya. Ini diperlukan untuk mendukung konfigurasi kumpulan pengguna Amazon Cognito.

Untuk melihat izin kebijakan ini, lihat AdministratorAccess-Amplify di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmplifyBackendDeployFullAccess

Anda dapat melampirkan kebijakan `AmplifyBackendDeployFullAccess` ke identitas IAM Anda.

Kebijakan ini memberikan Amplify izin akses penuh untuk menerapkan sumber daya backend Amplify menggunakan. AWS Cloud Development Kit (AWS CDK) Izin ditangguhkan ke AWS CDK peran yang memiliki izin kebijakan yang diperlukan`AdministratorAccess`.

Detail izin

Kebijakan ini mencakup izin untuk melakukan hal berikut.

- `Amplify`— Ambil metadata tentang aplikasi yang digunakan.
- `AWS CloudFormation`— Buat, perbarui, dan hapus Amplify tumpukan terkelola.
- `SSM`— Buat, perbarui, dan hapus Amplify yang dikelola SSM Parameter Store `String` dan parameter. `SecureString`
- `AWS AppSync`— Perbarui dan ambil AWS AppSync skema, resolver, dan sumber daya fungsi. Tujuannya adalah untuk mendukung fungsionalitas hotswapping kotak pasir Gen 2.
- `Lambda`— Perbarui dan ambil konfigurasi untuk Amplify fungsi yang dikelola. Tujuannya adalah untuk mendukung fungsionalitas hotswapping kotak pasir Gen 2.

  Ambil tag fungsi Lambda. Tujuannya adalah untuk mendukung fungsi Lambda yang ditentukan oleh pelanggan.
- `Amazon S3`— Ambil aset penerapan Amplify.
- `AWS Security Token Service`— Memungkinkan AWS Cloud Development Kit (AWS CDK) CLI untuk mengambil peran penerapan.
- `Amazon RDS`— Baca metadata instans, cluster, dan proxy DB.
- `Amazon EC2`— Baca informasi zona ketersediaan untuk subnet.
- `CloudWatch Logs`— Ambil log untuk fungsi Lambda pelanggan. Tujuannya adalah untuk memungkinkan lingkungan sandbox pengembangan cloud Amplify untuk mengalirkan log fungsi Lambda ke terminal pelanggan.

Untuk melihat izin kebijakan ini, lihat AmplifyBackendDeployFullAccessdi Referensi Kebijakan AWS Terkelola.

Amplify update ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amplify sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan ke umpan RSS pada halaman Riwayat dokumen untuk AWS Amplify.

Perubahan	Deskripsi	Tanggal
AmplifyBackendDeployFullAccess — Permbaruan ke kebijakan yang sudah ada	Tambahkan akses baca ke `logs:FilterLogEvents` sumber daya untuk memungkinkan Amplify mengalirkan log dari fungsi tempat grup log kustom dibuat. Ini adalah perpanjangan dari kemampuan yang ada untuk melakukan streaming log fungsi Lambda.	November 14, 2024
AmplifyBackendDeployFullAccess – Pembaruan ke kebijakan yang ada	Tambahkan akses baca ke `lambda:ListTags` dan `logs:FilterLogEvents` sumber daya untuk mendukung fungsi Lambda yang ditentukan oleh pelanggan. Izin ini memungkinkan lingkungan sandbox pengembangan cloud Amplify untuk mengalirkan log fungsi Lambda ke terminal pelanggan.	Juli 18, 2024
AmplifyBackendDeployFullAccess – Pembaruan ke kebijakan yang ada	Tambahkan akses baca ke `arn:aws:ssm:*:*:parameter/cdk-bootstrap/*` sumber daya untuk memungkinkan Amplify mendeteksi versi bootstrap CDK di akun pelanggan.	31 Mei 2024
AmplifyBackendDeployFullAccess – Pembaruan ke kebijakan yang ada	Tambahkan pernyataan `AmplifyDiscoverRDSVpcConfig` kebijakan baru	April 17, 2024

Perubahan	Deskripsi	Tanggal
	dengan izin EC2 hanya-bac a Amazon RDS dan Amazon yang dicakup oleh kondisi sumber daya dan akun. Izin ini mendukung perintah Amplify Gen `npx amplify generate schema-fr om-database` 2 yang memungkinkan pelanggan menghasilkan skema data TypeScript dari database SQL yang ada. Tambahkan`rds:Descr ibeDBProxies` ,, `rds:DescribeDBInst ances rds:Descr ibeDBClus ters` ,`rds:Descr ibeDBSubnetGroups` , dan `ec2:DescribeSubnet s` izin. `npx amplify generate schema-fr om-database` Perintah ini memerlukan izin ini untuk memeriksa apakah host DB tertentu di-host di Amazon RDS dan secara otomatis menghasilkan konfigurasi VPC Amazon yang diperluka n untuk menyediakan sumber daya lain yang diperluka n untuk menyiapkan AWS AppSync API yang didukung oleh database SQL.	

Perubahan	Deskripsi	Tanggal
AmplifyBackendDeployFullAccess – Pembaruan ke kebijakan yang ada	Tambahkan tindakan `cloudformation:DeleteStack` kebijakan untuk mendukung penghapusan tumpukan saat `DeleteBranch` API dipanggil. Tambahkan tindakan `lambda:GetFunction` kebijakan untuk mendukung fungsi hotswapping. Tambahkan tindakan `lambda:UpdateFunctionConfiguration` kebijakan untuk mendukung pembaruan fungsi Lambda.	April 5, 2024
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan `cloudformation:TagResource` dan `cloudformation:UnTagResource` izin untuk mendukung panggilan ke AWS CloudFormation APIs.	April 4, 2024

Perubahan	Deskripsi	Tanggal
AmplifyBackendDeployFullAccess – Pembaruan ke kebijakan yang ada	Tambahkan tindakan `lambda:InvokeFunction` kebijakan untuk mendukung AWS Cloud Development Kit (AWS CDK) hotswapping. Mereka AWS CDK melakukan panggilan langsung ke fungsi Lambda untuk melakukan hotswapping aset Amazon S3. Tambahkan tindakan `lambda:UpdateFunctionCode` kebijakan untuk mendukung fungsi hotswapping.	Januari 02, 2024
AmplifyBackendDeployFullAccess – Pembaruan ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk mendukung `UpdateApiKey` operasi. Ini diperlukan untuk mengaktifkan penerapan aplikasi yang berhasil setelah keluar dan memulai ulang kotak pasir tanpa menghapus sumber daya.	17 November 2023
AmplifyBackendDeployFullAccess – Pembaruan ke kebijakan yang ada	Tambahkan `amplify:GetBackendEnvironment` izin untuk mendukung penerapan aplikasi Amplify.	6 November 2023

Perubahan	Deskripsi	Tanggal
AmplifyBackendDeployFullAccess – Kebijakan baru	Amplify menambahkan kebijakan baru dengan izin minimum yang diperlukan untuk menerapkan sumber daya backend Amplify.	Oktober 8, 2023
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan `ecr:DescribeRepositories` izin yang diperlukan oleh Amplify Command Line Interface (CLI).	1 Juni 2023

Perubahan	Deskripsi	Tanggal
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk mendukung penghapusan tag dari AWS AppSync sumber daya. Tambahkan tindakan kebijakan untuk mendukung sumber daya Amazon Polly. Tambahkan tindakan kebijakan untuk mendukung pemutakhiran konfigurasi OpenSearch domain. Tambahkan tindakan kebijakan untuk mendukung penghapusan tag dari AWS Identity and Access Management peran. Tambahkan tindakan kebijakan untuk mendukung penghapusan tag dari sumber daya Amazon DynamoDB. Tambahkan `cloudfront:GetCloudFrontOriginAccessIdentityConfig` izin `cloudfront:GetCloudFrontOriginAccessIdentity` dan ke blok `CLISDKCalls` pernyataan untuk mendukung alur kerja publikasi dan hosting Amplify.	Februari 24, 2023

Perubahan	Deskripsi	Tanggal
	Tambahkan `s3:PutBucketPublicAccessBlock` izin ke blok `CLIManageviaCFNPolicy` pernyataan untuk memungkinkan mendukung praktik terbaik keamanan Amazon S3 AWS CLI untuk mengaktifkan fitur Akses Publik Blok Amazon S3 di bucket internal. Tambahkan `cloudformation:DescribeStacks` izin ke blok `CLISDKCalls` pernyataan untuk mendukung pengambilan AWS CloudFormation tumpukan pelanggan saat mencoba ulang di prosesor backend Amplify untuk menghindari duplikasi eksekusi jika tumpukan diperbarui. Tambahkan `cloudformation:ListStacks` izin ke blok `CLICloudformationPolicy` pernyataan. Izin ini diperlukan untuk sepenuhnya mendukung CloudFormation DescribeStacks tindakan tersebut.	

Perubahan	Deskripsi	Tanggal
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk memungkinkan fitur rendering sisi server Amplify mendorong metrik aplikasi ke dalam metrik pelanggan. CloudWatch Akun AWS	30 Agustus 2022
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk memblokir akses publik ke bucket Amplify deployment Amazon S3.	27 April 2022
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan tindakan untuk memungkinkan pelanggan menghapus aplikasi yang dirender sisi server (SSR) mereka. Ini juga memungkinkan CloudFront distribusi yang sesuai untuk dihapus dengan sukses. Tambahkan tindakan untuk memungkinkan pelanggan menentukan fungsi Lambda yang berbeda untuk menangani peristiwa dari sumber peristiwa yang ada menggunakan Amplify CLI. Dengan perubahan ini, AWS Lambda akan dapat melakukan UpdateEventSourceMappingtindakan.	April 17, 2022

Perubahan	Deskripsi	Tanggal
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk mengaktifkan tindakan Amplify UI Builder di semua sumber daya.	2 Desember 2021
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk mendukung fitur autentikasi Amazon Cognito yang menggunakan penyedia identitas sosial. Tambahkan tindakan kebijakan untuk mendukung lapisan Lambda. Tambahkan tindakan kebijakan untuk mendukung kategori Amplify Storage.	November 8, 2021

Perubahan	Deskripsi	Tanggal
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Tambahkan tindakan Amazon Lex untuk mendukung kategori Amplify Interactions. Tambahkan tindakan Amazon Rekognition untuk mendukung kategori Amplify Predictions. Tambahkan tindakan Amazon Cognito untuk mendukung konfigurasi MFA di kumpulan pengguna Amazon Cognito. Tambahkan CloudFormation tindakan untuk mendukung AWS CloudFormation StackSets. Tambahkan tindakan Amazon Location Service untuk mendukung kategori Amplify Geo. Tambahkan tindakan Lambda untuk mendukung lapisan Lambda di Amplify. Tambahkan tindakan CloudWatch Log untuk mendukung CloudWatch Acara. Tambahkan tindakan Amazon S3 untuk mendukung kategori Amplify Storage.	27 September 2021

Perubahan	Deskripsi	Tanggal
	Tambahkan tindakan kebijakan untuk mendukung aplikasi yang dirender sisi server (SSR).	

Perubahan	Deskripsi	Tanggal
AdministratorAccess-Amplify - Perbarui ke kebijakan yang ada	Konsolidasikan semua tindakan Amplify menjadi satu `amplify:*` tindakan. Tambahkan tindakan Amazon S3 untuk mendukung mengenkripsi bucket Amazon S3 pelanggan. Tambahkan tindakan batas izin IAM untuk mendukung Amplify aplikasi yang memiliki batas izin diaktifkan. Tambahkan tindakan Amazon SNS untuk mendukung melihat nomor telepon originasi, serta melihat, membuat, memverifikasi, dan menghapus nomor telepon tujuan. Amplify Studio: Tambahkan Amazon Cognito AWS Lambda, IAM, AWS CloudFormation dan tindakan kebijakan untuk mengaktif kan pengelolaan backend di konsol Amplify dan Amplify Studio. Tambahkan pernyataan kebijakan AWS Systems Manager (SSM) untuk mengelola rahasia lingkungan Amplify.	28 Juli 2021

Perubahan	Deskripsi	Tanggal
	Tambahkan AWS CloudFormation `ListResources` tindakan untuk mendukung lapisan Lambda untuk aplikasi Amplify.	
Amplify mulai melacak perubahan	Amplify mulai melacak perubahan untuk kebijakan AWS terkelolanya.	28 Juli 2021

Pemecahan masalah identitas dan akses Amplify

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan mengatasi masalah umum yang mungkin dialami saat menggunakan Amplify dan IAM.

Topik

- Saya tidak memiliki otorisasi untuk melakukan tindakan di Amplify
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya Amplify saya

Saya tidak memiliki otorisasi untuk melakukan tindakan di Amplify

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `amplify:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplify:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `amplify:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Dengan dirilisnya Amplify Studio, menghapus aplikasi atau backend memerlukan keduanya dan izin. `amplify` `amplifybackend` Jika administrator telah menulis kebijakan IAM yang hanya menyediakan `amplify` izin, Anda akan mendapatkan kesalahan izin saat mencoba menghapus aplikasi.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk menghapus sumber daya *`example-amplify-app`* fiktif, tetapi tidak memiliki izin `amplifybackend:`*`RemoveAllBackends`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplifybackend;:RemoveAllBackends on resource: example-amplify-app
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya agar dia dapat mengakses *`example-amplify-app`* menggunakan `amplifybackend:`*`RemoveAllBackends`* tindakan.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amplify.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi saat pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amplify. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya Amplify saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui Amplify mendukung fitur ini atau tidak, lihat Cara Amplify bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Perlindungan Data di Amplify

AWS Amplify sesuai dengan model tanggung jawab AWS bersama model tanggung , yang mencakup peraturan dan pedoman untuk perlindungan data. AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS layanan. AWS Mempertahankan kontrol atas data yang dihosting di infrastruktur ini, termasuk kontrol konfigurasi keamanan untuk menangani konten pelanggan dan data pribadi. AWS pelanggan dan mitra APN, yang bertindak sebagai pengontrol data atau pengolah data, bertanggung jawab atas data pribadi apa pun yang mereka masukkan ke Cloud. AWS

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara ini, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugas mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut ini:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default dalam AWS layanan.
- Gunakan layanan keamanan terkelola lanjutan seperti Amazon Macie, yang membantu menemukan dan mengamankan data pribadi yang disimpan di Amazon S3.

Sebaiknya jangan pernah memasukkan informasi identitas yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amplify atau AWS layanan lain menggunakan konsol, API AWS CLI, atau. AWS SDKs Data apa pun yang Anda masukkan ke Amplify atau layanan lain mungkin akan diambil dan dimasukkan ke dalam log diagnostik. Saat Anda memberikan URL ke server eksternal, jangan menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Untuk informasi selengkapnya tentang perlindungan data, lihat postingan blog Model Tanggung Jawab Bersama AWS dan GDPR di Blog Keamanan AWS .

Enkripsi saat istirahat

Enkripsi saat istirahat didefinisikan sebagai perlindungan data dari akses tidak sah dengan mengenkripsi data saat disimpan. Amplify mengenkripsi artefak build aplikasi secara default menggunakan Amazon AWS KMS keys S3 yang dikelola oleh file. AWS Key Management Service

Amplify menggunakan Amazon CloudFront untuk menayangkan aplikasi Anda kepada pelanggan Anda. CloudFront menggunakan SSDs yang dienkripsi untuk titik lokasi tepi kehadiran (POPs), dan volume EBS terenkripsi untuk Regional Edge Cache (). RECs Kode fungsi dan konfigurasi dalam CloudFront Fungsi selalu disimpan dalam format terenkripsi pada lokasi tepi terenkripsi POPs, dan SSDs di lokasi penyimpanan lain yang digunakan oleh. CloudFront

Enkripsi bergerak

Enkripsi dalam transit didefinisikan sebagai perlindungan data dari intersepsi saat data ditransfer antar-endpoint komunikasi. Amplify Hosting menyediakan enkripsi untuk data dalam transit secara default. Semua komunikasi antara pelanggan dan Amplify serta antara Amplify dan dependensi hilir dilindungi menggunakan koneksi TLS yang ditandatangani menggunakan proses penandatanganan Signature Version 4. Semua titik akhir Amplify Hosting menggunakan sertifikat SHA-256 yang dikelola oleh. AWS Private Certificate Authority Untuk informasi selengkapnya, lihat Proses penandatanganan Versi Tanda Tangan 4 dan Apa itu AWS Private Certificate Authority.

Pengelolaan kunci enkripsi

AWS Key Management Service (KMS) adalah layanan terkelola untuk membuat dan mengendalikan AWS KMS keys, kunci enkripsi yang digunakan untuk mengenkripsi data pelanggan. AWS Amplify menghasilkan dan mengelola kunci kriptografi untuk mengenkripsi data atas nama pelanggan. Tidak ada kunci enkripsi yang perlu Anda kelola.

Validasi Kepatuhan untuk AWS Amplify

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS Amplify sebagai bagian dari beberapa program AWS kepatuhan. Ini mencakup SOC, PCI, ISO, HIPAA, MTCS, C5, K-ISMS, ENS High, OSPAR, HITRUST CSF, dan FINMA.

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Kepatuhan dan Tata Kelola Keamanan – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- Referensi Layanan yang Memenuhi Syarat HIPAA — Daftar layanan yang memenuhi syarat HIPAA. Tidak semua memenuhi Layanan AWS syarat HIPAA.

- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Keamanan Infrastruktur di AWS Amplify

Sebagai layanan terkelola, AWS Amplify dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amplify melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.

- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Pencatatan dan pemantauan peristiwa keamanan di Amplify

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amplify dan solusi Anda yang lain AWS . AWS menyediakan alat pemantauan berikut untuk menonton Amplify, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau secara real time AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor khusus, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik mencapai ambang batas yang Anda tetapkan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon Elastic Compute Cloud EC2 (Amazon) dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya tentang penggunaan CloudWatch metrik dan alarm dengan Amplify, lihat. Memantau aplikasi Amplify
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari EC2 instans Amazon AWS CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- AWS CloudTrailmenangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon Simple Storage Service (Amazon S3) yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, lihat Mencatat log panggilan API Amplify dengan AWS CloudTrail.
- Amazon EventBridge adalah layanan bus acara tanpa server yang memudahkan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge mengirimkan aliran data real-time dari aplikasi Anda sendiri, aplikasi Software-as-a-Service (SaaS), AWS dan layanan, dan rute data tersebut ke target seperti. AWS Lambda Dengan demikian, Anda dapat

memantau peristiwa yang terjadi dalam layanan dan membangun arsitektur berbasis peristiwa. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

Pencegahan "confused deputy" lintas layanan

Masalah "confused deputy" adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang memilik hak akses lebih tinggi untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan dalam kebijakan sumber daya untuk membatasi izin yang AWS Amplify memberikan layanan lain ke sumber daya. Jika Anda menggunakan kedua kunci konteks kondisi global, `aws:SourceAccount` nilai dan akun dalam `aws:SourceArn` nilai harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Nilai `aws:SourceArn` harus menjadi ARN cabang dari aplikasi Amplify. Tentukan nilai ini dalam format`arn:`*`Partition`*`:amplify:`*`Region`*`:`*`Account`*`:apps/`*`AppId`*`/branches/`*`BranchName`*.

Cara paling efektif untuk melindungi dari masalah "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Contoh berikut menunjukkan kebijakan kepercayaan peran yang dapat Anda terapkan untuk membatasi akses ke aplikasi Amplify apa pun di akun Anda dan mencegah masalah deputi yang membingungkan. Untuk menggunakan kebijakan ini, ganti teks miring merah dalam kebijakan contoh dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": {
```

```
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": [
          "amplify.me-south-1.amazonaws.com",
          "amplify.eu-south-1.amazonaws.com",
          "amplify.ap-east-1.amazonaws.com",
          "amplifybackend.amazonaws.com",
          "amplify.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Contoh berikut menunjukkan kebijakan kepercayaan peran yang dapat Anda terapkan untuk membatasi akses ke aplikasi Amplify tertentu di akun Anda dan mencegah masalah deputi yang membingungkan. Untuk menggunakan kebijakan ini, ganti teks miring merah dalam kebijakan contoh dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": [
          "amplify.me-south-1.amazonaws.com",
          "amplify.eu-south-1.amazonaws.com",
          "amplify.ap-east-1.amazonaws.com",
          "amplifybackend.amazonaws.com",
          "amplify.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole",
```

```
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/d123456789/
branches/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Praktik terbaik keamanan untuk Amplify

Amplify menyediakan sejumlah fitur keamanan untuk dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai rekomendasi yang bermanfaat, bukan sebagai resep.

Menggunakan cookie dengan domain default Amplify

Saat Anda menggunakan Amplify untuk menerapkan aplikasi web, Amplify menghostingnya untuk Anda di domain default. `amplifyapp.com` Anda dapat melihat aplikasi Anda di URL yang diformat sebagai`https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`.

Untuk meningkatkan keamanan aplikasi Amplify Anda, domain amplifyapp.com terdaftar di Daftar Akhiran Publik (PSL). Untuk keamanan lebih lanjut, kami menyarankan Anda menggunakan cookie dengan `__Host-` awalan jika Anda perlu mengatur cookie sensitif di nama domain default untuk aplikasi Amplify Anda. Praktik ini akan membantu mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat halaman Set-Cookie di Jaringan Pengembang Mozilla.

Kuota layanan Amplify Hosting

Berikut ini adalah kuota layanan untuk AWS Amplify Hosting. Kuota layanan (sebelumnya disebut sebagai batas) adalah jumlah maksimum sumber daya layanan atau operasi untuk Anda Akun AWS.

Baru Akun AWS telah mengurangi aplikasi dan kuota pekerjaan bersamaan. AWS meningkatkan kuota ini secara otomatis berdasarkan penggunaan Anda. Anda juga dapat meminta peningkatan kuota.

Konsol Service Quotas memberikan informasi tentang kuota untuk akun Anda. Anda dapat menggunakan konsol Service Quotas untuk melihat kuota default dan meminta peningkatan kuota untuk kuota yang dapat disesuaikan. Untuk informasi selengkapnya, lihat Meminta peningkatan kuota di Panduan Pengguna Service Quotas.

Nama	Default	Dapat disesuaikan	Deskripsi
Aplikasi	Setiap Wilayah yang didukung: 25	Ya	Jumlah maksimum aplikasi yang dapat Anda buat di AWS Amplify Console di akun ini di Wilayah saat ini.
Cabang per aplikasi	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum cabang per aplikasi yang dapat Anda buat di akun ini di Wilayah saat ini.
Ukuran artefak build	Setiap Wilayah yang didukung: 5 Gigabytes	Tidak	Ukuran maksimum (dalam GB) artefak pembuatan aplikasi. Artefak build diterapkan oleh AWS Amplify Console setelah build.

Nama	Default	Dapat disesuaikan	Deskripsi
Ukuran artefak cache	Setiap Wilayah yang didukung: 5 Gigabytes	Tidak	Ukuran maksimum (dalam GB) artefak cache.
Tugas bersamaan	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum pekerjaan bersamaan yang dapat Anda buat di akun ini di Wilayah saat ini.
Domain per aplikasi	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum domain per aplikasi yang dapat Anda buat di akun ini di Wilayah saat ini.
Ukuran artefak cache lingkungan	Setiap Wilayah yang didukung: 5 Gigabytes	Tidak	Ukuran maksimum (dalam GB) artefak cache lingkungan.
Ukuran file ZIP deploy manual	Setiap Wilayah yang didukung: 5 Gigabytes	Tidak	Ukuran maksimum (dalam GB) dari file ZIP penyebaran manual.
Pembuatan aplikasi maksimum per jam	Setiap Wilayah yang didukung: 25	Tidak	Jumlah maksimum aplikasi yang dapat Anda buat di AWS Amplify Console per jam di akun ini di Wilayah saat ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Minta token per detik	Setiap Wilayah yang didukung: 20.000	Ya	Jumlah maksimum token permintaan per detik untuk sebuah aplikasi. Amplify Hosting mengalokasikan token ke permintaan berdasarkan jumlah sumber daya (waktu pemrosesan dan transfer data) yang mereka konsumsi.
Subdomain per domain	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum subdomain per domain yang dapat Anda buat di akun ini di Wilayah saat ini.
Webhooks per aplikasi	Setiap Wilayah yang didukung: 50	Ya	Jumlah maksimum webhook per aplikasi yang dapat Anda buat di akun ini di Wilayah saat ini.

Untuk informasi selengkapnya tentang kuota layanan Amplify, lihat AWS Amplify titik akhir dan kuota di. Referensi Umum AWS

Memecahkan Masalah Amplify

Jika Anda mengalami kesalahan atau masalah deployment saat bekerja dengan Amplify Hosting, periksa topik di bagian ini.

Topik

Memecahkan masalah Amplify umum

Informasi berikut dapat membantu Anda memecahkan masalah pada Amplify Hosting.

Topik

Kode status HTTP 429 (Terlalu banyak permintaan)

Amplify mengontrol jumlah permintaan per detik (RPS) ke situs web Anda berdasarkan waktu pemrosesan dan transfer data yang digunakan permintaan masuk. Jika aplikasi Anda mengembalikan kode status HTTP 429, permintaan masuk melebihi jumlah waktu pemrosesan dan transfer data yang dialokasikan ke aplikasi Anda. Batas aplikasi ini dikelola oleh kuota

`REQUEST_TOKENS_PER_SECOND` layanan Amplify. Untuk informasi selengkapnya tentang kuota, lihatKuota layanan Amplify Hosting.

Untuk memperbaiki masalah ini, kami sarankan mengoptimalkan aplikasi Anda untuk mengurangi durasi permintaan dan transfer data untuk meningkatkan RPS aplikasi. Misalnya, dengan 20.000 token yang sama, halaman SSR yang sangat dioptimalkan yang merespons dalam 100 milidetik dapat mendukung RPS yang lebih tinggi dibandingkan dengan halaman dengan latensi lebih tinggi dari 200 milidetik.

Demikian pula, aplikasi yang mengembalikan ukuran respons 1 MB akan mengkonsumsi lebih banyak token daripada aplikasi yang mengembalikan ukuran respons 250 KB.

Kami juga menyarankan Anda memanfaatkan CloudFront cache Amazon dengan mengonfigurasi Cache-Control header yang memaksimalkan waktu respons yang diberikan disimpan dalam cache. Permintaan yang disajikan dari CloudFront cache tidak dihitung terhadap batas tarif. Setiap CloudFront distribusi dapat menangani hingga 250.000 permintaan per detik, memungkinkan Anda untuk menskalakan aplikasi Anda dengan sangat tinggi menggunakan cache. Untuk informasi selengkapnya tentang CloudFront cache, lihat Mengoptimalkan caching dan ketersediaan di Panduan CloudFront Pengembang Amazon.

Konsol Amplify tidak menampilkan status build dan waktu pembaruan terakhir untuk aplikasi saya

Saat Anda menavigasi ke halaman Semua aplikasi di konsol Amplify, ubin akan ditampilkan untuk setiap aplikasi Anda di Wilayah saat ini. Jika Anda tidak melihat status build, seperti Deployed, dan Waktu pembaruan terakhir ditampilkan untuk aplikasi, aplikasi tidak memiliki cabang `Production` tahap yang terkait dengannya.

Untuk membuat daftar aplikasi di konsol, Amplify menggunakan API. `ListApps` Amplify menggunakan `ProductionBranch.status` atribut untuk menampilkan status build dan `ProductionBranch.lastDeployTime` atribut untuk menampilkan waktu pembaruan terakhir. Untuk informasi selengkapnya tentang API ini, lihat ProductionBranchdi dokumentasi Amplify Hosting API.

Gunakan petunjuk berikut untuk mengaitkan `Production` stage ke cabang aplikasi Anda.

1. Masuk ke konsol Amplify.
2. Di halaman Semua aplikasi, pilih aplikasi yang akan diperbarui.
3. Di panel navigasi pilih Pengaturan aplikasi, lalu Pengaturan cabang.

4. Di bagian Pengaturan cabang, pilih Edit.
5. Untuk cabang Produksi, pilih nama cabang yang ingin Anda gunakan.
6. Pilih Simpan.
7. Kembali ke halaman Semua aplikasi. Status build dan waktu pembaruan terakhir sekarang harus ditampilkan untuk aplikasi Anda.

Pratinjau web tidak dibuat untuk permintaan tarik baru

Fitur pratinjau web memungkinkan Anda untuk melihat pratinjau perubahan dari permintaan tarik sebelum menggabungkannya ke cabang integrasi. Pratinjau web men-deploy setiap permintaan tarik ke repositori Anda ke URL pratinjau unik yang berbeda dari URL yang digunakan situs utama Anda.

Jika Anda telah mengaktifkan pratinjau web untuk aplikasi Anda, tetapi pratinjau tersebut tidak dibuat untuk yang baru PRs, selidiki apakah salah satu dari berikut ini adalah penyebab masalah Anda.

1. Periksa untuk melihat apakah aplikasi Anda telah mencapai kuota `Branches per app` layanan maksimal. Untuk informasi selengkapnya tentang kuota, lihatKuota layanan Amplify Hosting.

   Untuk tetap berada dalam kuota default 50 cabang per aplikasi, pertimbangkan untuk mengaktifkan penghapusan cabang otomatis di aplikasi Anda. Ini akan mencegah Anda mengumpulkan cabang di akun Anda yang tidak lagi ada di repositori Anda.
2. Jika Anda menggunakan GitHub repositori publik dan aplikasi Amplify Anda memiliki peran layanan IAM yang melekat padanya, Amplify tidak membuat pratinjau untuk alasan keamanan. Misalnya, aplikasi dengan backend dan aplikasi yang digunakan ke platform `WEB_COMPUTE` hosting memerlukan peran layanan IAM. Oleh karena itu, Anda tidak dapat mengaktifkan pratinjau web untuk jenis aplikasi ini jika repositori mereka bersifat publik.

   Agar pratinjau web berfungsi untuk aplikasi Anda, Anda dapat memisahkan peran layanan (jika aplikasi tidak memiliki backend atau bukan `WEB_COMPUTE` aplikasi), atau Anda dapat membuat repositori menjadi pribadi. GitHub

Penerapan manual saya macet dengan status tertunda di konsol Amplify

Dengan deployment manual, Anda dapat memublikasikan aplikasi web dengan Amplify Hosting tanpa menghubungkan penyedia Git. Anda dapat menggunakan salah satu dari empat opsi penerapan berikut.

1. Seret dan lepas folder aplikasi Anda di konsol Amplify.
2. Seret dan lepas file.zip (yang berisi artefak build situs Anda) di konsol Amplify.
3. Unggah file.zip (yang berisi artefak build situs Anda) ke bucket Amazon S3 dan sambungkan bucket ke aplikasi di konsol Amplify.
4. Gunakan URL publik yang mengarah ke file.zip (yang berisi artefak build situs Anda) di konsol Amplify.

Kami mengetahui masalah dengan fungsionalitas drag a drop saat menggunakan folder aplikasi untuk penerapan manual di konsol Amplify. Penerapan ini dapat gagal karena alasan berikut.

- Masalah jaringan transien terjadi.
- Ada perubahan lokal pada file saat mengunggah.
- Sesi browser mencoba mengunggah sejumlah besar aset statis secara bersamaan.

Sementara kami berupaya meningkatkan keandalan upload drag and drop kami, kami menyarankan Anda menggunakan file.zip alih-alih menyeret dan menjatuhkan folder aplikasi.

Kami sangat menyarankan untuk mengunggah file.zip ke bucket Amazon S3, karena ini menghindari unggahan file dari konsol Amplify dan memberikan keandalan yang lebih tinggi untuk penerapan manual. Integrasi Amplify dengan Amazon S3 menyederhanakan proses ini. Untuk informasi selengkapnya, lihat Menerapkan situs web statis untuk Amplify dari bucket Amazon S3.

Saya perlu memperbarui versi Node.js aplikasi saya

Amplify dukungan untuk aplikasi yang menggunakan Node.js versi 16 dan 18 berakhir pada 15 September 2025. Aplikasi yang sudah digunakan akan terus berjalan. Namun, setelah tanggal ini, Anda tidak akan dapat menerapkan pembaruan ke aplikasi Anda sampai Anda meningkatkan ke Node.js versi 20 atau yang lebih baru.

Jika Anda menggunakan image build Amazon Linux 2023, Node.js versi 20 didukung secara default. Mulai 15 September 2025, gambar AL2 023 akan secara otomatis mendukung Node.js 22 dan mengubah versi Node.js default dari 18 menjadi 22.

Amazon Linux 2 (AL2) tidak secara otomatis mendukung Node.js versi 20 atau yang lebih baru. Jika saat ini Anda menggunakan AL2, kami sarankan Anda beralih ke AL2 023. Anda dapat mengubah image build di konsol Amplify. Anda juga dapat menggunakan image build kustom yang mendukung versi Node.js yang Anda tentukan.

Sebelum memutakhirkan, kami sarankan Anda menguji aplikasi Anda di cabang baru untuk memverifikasi bahwa itu berfungsi dengan benar.

Opsi peningkatan

Konsol Amplify

Anda dapat menggunakan fitur pembaruan paket langsung di konsol Amplify untuk menentukan versi Node.js yang akan digunakan. Untuk petunjuk, lihat Menggunakan versi paket dan dependensi tertentu dalam image build.

Citra build kustom

Jika Anda menggunakan image build kustom dan NVM diinstal pada gambar Anda, Anda dapat menambahkan `nvm install 20` ke Dockerfile Anda. Untuk mempelajari lebih lanjut tentang persyaratan dan instruksi konfigurasi untuk image build kustom, lihatMenyesuaikan gambar build.

Pengaturan Bangunan

Anda dapat menentukan versi Node.js yang akan digunakan dalam pengaturan `amplify.yml` build aplikasi Anda, dengan menambahkan `nvm use` perintah ke bagian perintah PreBuild. Untuk petunjuk memperbarui pengaturan build aplikasi, lihatMengonfigurasi pengaturan build untuk aplikasi Amplify.

Contoh berikut menunjukkan cara menyesuaikan pengaturan build untuk menyetel versi Node.js default ke Node.js 18 dan meningkatkan ke Node.js versi 20 pada cabang pengujian bernama`node-20`.

```
frontend:
  phases:
    preBuild:
      commands:
        - nvm use 18
        - if [ "${AWS_BRANCH}" = "node-20" ]; then nvm use 20; fi
```

Warning

Ketahuilah bahwa `preBuild` perintah berjalan setelah pembaruan paket langsung. Versi Node.js yang ditentukan oleh `nvm use` perintah akan mengganti versi Node.js yang ditetapkan oleh pembaruan paket langsung.

Memecahkan masalah image build Amazon Linux 2023

Informasi berikut dapat membantu Anda memecahkan masalah pada image build Amazon Linux 2023 (AL2023).

Topik

Saya ingin menjalankan fungsi Amplify dengan runtime Python

Amplify Hosting sekarang menggunakan image build Amazon Linux 2023 secara default saat Anda menerapkan aplikasi baru. AL2023 hadir pra-instal dengan Python versi 3.8, 3.9, 3.10, dan 3.11.

Untuk kompatibilitas mundur dengan image Amazon Linux 2, image build AL2 023 memiliki symlink untuk versi Python yang sudah diinstal sebelumnya.

Secara default, Python versi 3.10 digunakan secara global. Untuk membangun fungsi Anda menggunakan versi Python tertentu, jalankan perintah berikut dalam file spesifikasi build aplikasi Anda.

```
version: 1
backend:
  phases:
    build:
      commands:
        # use a python version globally
        - pyenv global 3.11
        # verify python version
        - python --version
        # install pipenv
        - pip install --user pipenv
        # add to path
        - export PATH=$PATH:/root/.local/bin
        # verify pipenv version
        - pipenv --version
        - amplifyPush --simple
```

Saya ingin menjalankan perintah yang membutuhkan hak superuser atau root

Jika Anda menggunakan image build Amazon Linux 2023 dan mendapatkan kesalahan saat menjalankan perintah sistem yang memerlukan hak superuser atau root, Anda harus menjalankan perintah ini menggunakan perintah Linux. `sudo` Misalnya, jika Anda menjalankan kesalahan`yum install -y gcc`, gunakan`sudo yum install -y gcc`.

Gambar build Amazon Linux 2 menggunakan pengguna root, tetapi gambar AL2 023 Amplify menjalankan kode Anda dengan pengguna khusus`amplify`. Amplify memberikan hak istimewa kepada pengguna ini untuk menjalankan perintah menggunakan perintah Linux. `sudo` Ini adalah praktik terbaik untuk digunakan `sudo` untuk perintah yang membutuhkan hak superuser.

Memecahkan masalah build

Jika Anda mengalami masalah saat membuat atau membangun aplikasi Amplify, konsultasikan topik tersebut di bagian ini untuk bantuan.

Topik

- Komit baru ke repositori saya tidak memicu build Amplify
- Nama repositori saya tidak tercantum di konsol Amplify saat membuat aplikasi baru
- Build saya gagal dengan Cannot find module aws-exports kesalahan (hanya aplikasi Gen 1)
- Saya ingin mengganti batas waktu build

Komit baru ke repositori saya tidak memicu build Amplify

Jika komit baru ke repositori Git Anda tidak memicu build Amplify, verifikasi bahwa webhook Anda masih ada di repositori Anda. Jika ada, periksa riwayat permintaan webhook untuk melihat apakah ada kegagalan. Amplify memiliki batas ukuran payload 256 KB untuk webhook yang masuk. Jika Anda mendorong komit ke repositori yang memiliki banyak file yang diubah, Anda mungkin melebihi batas ini dan menyebabkan build tidak terpicu.

Nama repositori saya tidak tercantum di konsol Amplify saat membuat aplikasi baru

Saat membuat aplikasi baru di konsol Amplify, Anda dapat memilih dari repositori organisasi yang tersedia di halaman Tambah repositori dan cabang. Repositori target Anda mungkin tidak ditampilkan dalam daftar jika belum diperbarui baru-baru ini. Ini mungkin terjadi jika organisasi Anda memiliki sejumlah besar repositori. Untuk mengatasi masalah ini, dorong komit ke repositori, lalu segarkan daftar repositori di konsol. Ini akan menyebabkan repositori ditampilkan.

Build saya gagal dengan **`Cannot find module aws-exports`** kesalahan (hanya aplikasi Gen 1)

Jika aplikasi Anda tidak dapat menemukan `aws-exports.js` file selama build, kesalahan berikut akan ditampilkan.

```
TS2307: Cannot find module 'aws-exports'
```

Antarmuka baris perintah Amplify (CLI) menghasilkan `aws-exports.js` file selama build backend Anda. Untuk mengatasi kesalahan ini, Anda harus membuat `aws-exports.js` file untuk digunakan dalam build. Tambahkan kode berikut ke spesifikasi build Anda untuk membuat file:

```
backend:
  phases:
    build:
      commands:
        - "# Execute Amplify CLI with the helper script"
        - amplifyPush --simple
```

Untuk contoh lengkap setelan spesifikasi build untuk aplikasi Amplify, lihat. Sintaks YAMAL spesifikasi build

Saya ingin mengganti batas waktu build

Batas waktu build default adalah 30 menit. Anda dapat mengganti batas waktu build default menggunakan variabel `_BUILD_TIMEOUT` lingkungan. Waktu habis build minimum adalah 5 menit. Batas waktu build maksimum adalah 120 menit.

Untuk petunjuk tentang menyetel variabel lingkungan untuk aplikasi di konsol Amplify, lihat. Mengatur variabel lingkungan

Memecahkan masalah domain kustom

Jika Anda mengalami masalah saat menghubungkan domain kustom ke aplikasi Amplify, periksa topik di bagian ini untuk bantuan.

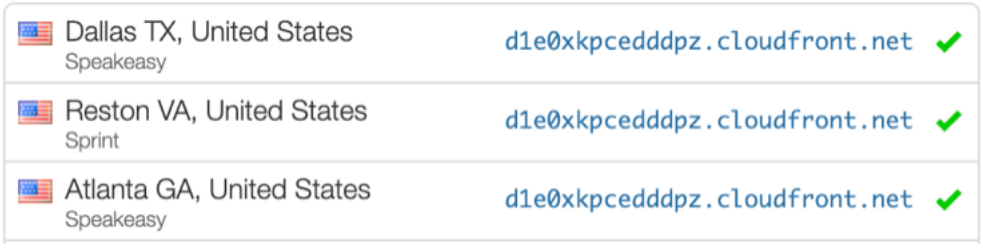
Jika Anda tidak melihat solusi untuk masalah Anda di sini, hubungi Dukungan. Untuk informasi selengkapnya, lihat Membuat kasus dukungan di Panduan AWS Dukungan Pengguna.

Topik

Saya perlu memverifikasi bahwa CNAME saya berhasil ditetapkan

1. Setelah memperbarui catatan DNS dengan penyedia domain pihak ketiga, Anda dapat menggunakan alat, seperti dig atau situs web gratis, seperti https://www.whatsmydns.net/ untuk memverifikasi bahwa catatan CNAME Anda berhasil ditetapkan dengan benar. Tangkapan layar berikut menunjukkan cara menggunakan whatsmydns.net untuk memeriksa catatan CNAME Anda untuk domain www.example.com.

2. Pilih Cari, lalu whatsmydns.net akan menampilkan hasil untuk CNAME Anda. Tangkapan layar berikut adalah contoh daftar hasil yang memverifikasi bahwa CNAME berhasil ditetapkan dengan benar ke URL cloudfront.net.



Domain saya yang di-host dengan pihak ketiga terus menampilkan status Menunggu Verifikasi

1. Menunggu Verifikasi, verifikasi bahwa CNAME catatan Anda berhasil ditetapkan. Lihat topik pemecahan masalah sebelumnya, Bagaimana cara memverifikasi bahwa saya berhasil ditetapkan CNAME, untuk instruksi menjalankan tugas ini.
2. Jika CNAME catatan Anda tidak berhasil ditetapkan, konfirmasi bahwa CNAME entri ada di pengaturan DNS dengan penyedia domain Anda.

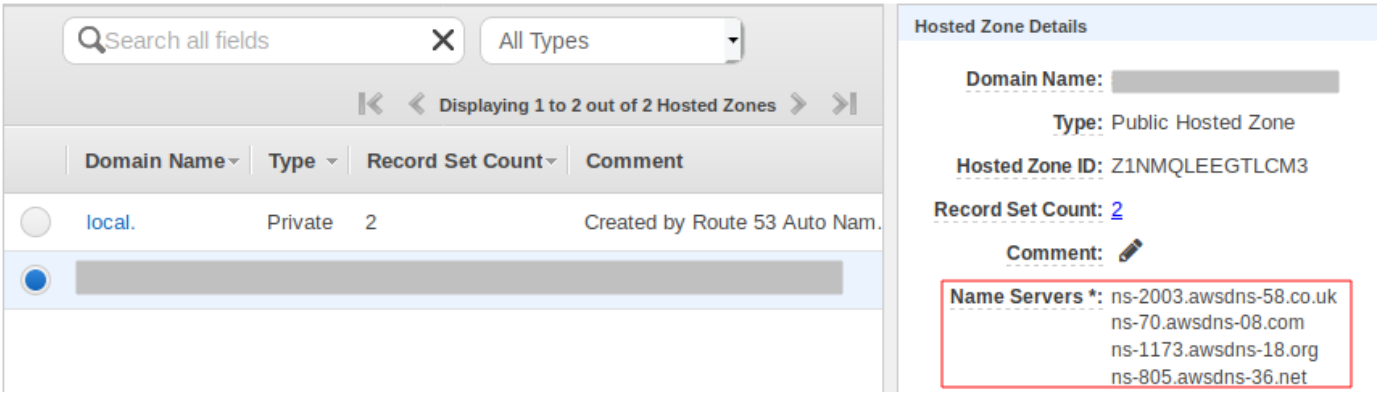
> **Important**
>
> CNAMECatatan harus diperbarui segera setelah membuat domain kustom. Setelah aplikasi dibuat di konsol Amplify, CNAME catatan akan diperiksa setiap beberapa menit untuk mengetahui catatan berhasil ditetapkan atau tidak. Jika catatan tidak berhasil ditetapkan setelah satu jam, pemeriksaan dilakukan setiap beberapa jam, dan dapat mengakibatkan penyiapan domain memerlukan waktu lebih lama. Menambahkan atau memperbarui CNAME catatan beberapa jam setelah membuat aplikasi adalah penyebab paling mungkin mengapa aplikasi Anda terus menampilkan status Menunggu Verifikasi.

3. Jika Anda telah memverifikasi bahwa CNAME catatan ada, masalah mungkin terjadi pada penyedia DNS Anda. Anda dapat menghubungi penyedia DNS untuk mendiagnosis penyebab verifikasi DNS tidak berhasil CNAME ditetapkan atau Anda dapat memigrasi DNS ke Route 53. Untuk informasi selengkapnya, lihat Menetapkan Amazon Route 53 sebagai layanan DNS untuk domain yang ada.

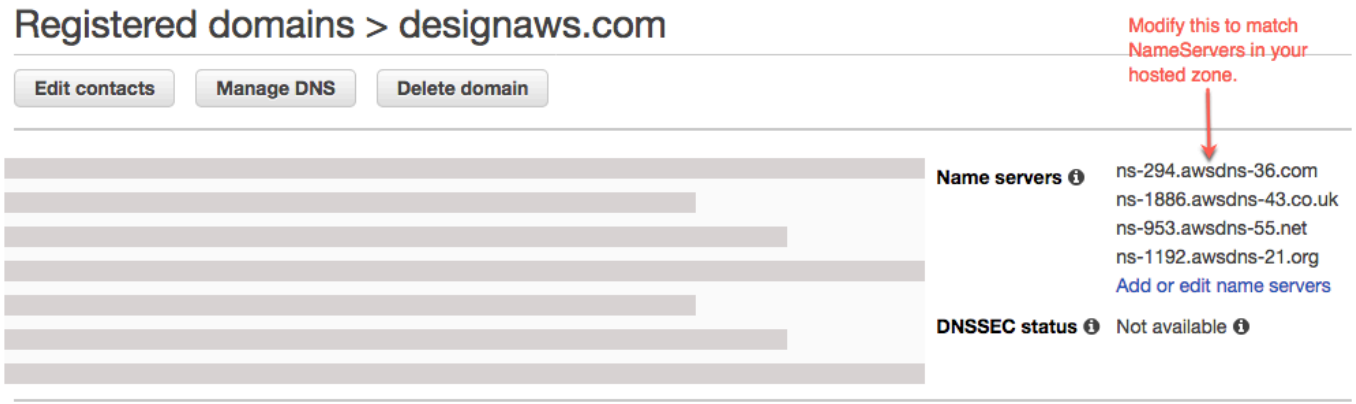
Domain saya yang di-host dengan Amazon Route 53 terus menampilkan status Menunggu Verifikasi

Jika Anda mentransfer domain Anda ke Amazon Route 53, mungkin domain Anda memiliki server nama yang berbeda dari yang diterbitkan Amplify saat aplikasi Anda dibuat. Lakukan langkah-langkah berikut untuk mendiagnosis penyebab kesalahan.

1. Masuk ke konsol Amazon Route 53
2. Di panel navigasi, pilih Zona yang Di-hosting, lalu pilih nama domain yang Anda hubungkan.
3. Catat nilai server nama dari bagian Detail Zona yang Di-hosting. Anda memerlukan nilai-nilai ini untuk menyelesaikan langkah berikutnya. Tangkapan layar berikut dari konsol Route 53 menampilkan lokasi nilai server nama di pojok kanan bawah.



4. Di panel navigasi, pilih Domain terdaftar. Verifikasi bahwa server nama yang ditampilkan di bagian Domain terdaftar sesuai dengan nilai server nama yang Anda catat di langkah sebelumnya dari bagian Detail Zona yang Di-hosting. Jika keduanya tidak cocok, edit nilai server nama agar sesuai dengan nilai di Zona yang Di-hosting. Tangkapan layar berikut dari konsol Route 53 menampilkan lokasi nilai server nama di bagian kanan.



5. Jika ini tidak menyelesaikan masalah, hubungi Dukungan. Untuk informasi selengkapnya, lihat Membuat kasus dukungan di Panduan AWS Dukungan Pengguna.

Aplikasi saya dengan subdomain multi-level terus menampilkan status Menunggu Verifikasi

Jika aplikasi dengan subdomain multi-level macet dalam status Verifikasi Tertunda saat menyambung ke penyedia DNS pihak ketiga, mungkin ada masalah dengan format catatan DNS Anda. Beberapa penyedia DNS secara otomatis menambahkan sufiks domain tingkat kedua (SLD) dan domain tingkat atas (TLD) ke catatan Anda. Jika Anda juga menentukan domain dalam format yang mencakup SLD dan TLD, ini dapat menyebabkan masalah verifikasi domain.

Saat Anda menghubungkan domain, pertama-tama coba tentukan nama domain menggunakan format lengkap yang disediakan oleh Amplify, misalnya. `_hash.docs.backend.example.com` Jika konfigurasi SSL macet dalam status Verifikasi Tertunda, coba hapus TLD dan SLD dari catatan. Misalnya, jika format lengkapnya`_hash.docs.backend.example.com`, tentukan`_hash.docs.backend`. Tunggu 15 hingga 30 menit untuk memungkinkan catatan menyebar. Kemudian gunakan alat seperti MX Toolbox untuk memeriksa apakah proses verifikasi berfungsi.

Penyedia DNS saya tidak mendukung catatan A dengan nama domain yang sepenuhnya memenuhi syarat

Beberapa penyedia DNS tidak mendukung catatan A dengan nama domain yang memenuhi syarat (FQDN), seperti. `example.cloudfront.net` Misalnya, Cloudflare hanya A records dapat menulis IPv4 alamat dan tidak mendukung. FQDNs Untuk mengatasi batasan ini, sebaiknya gunakan CNAME catatan alih-alih A records dalam DNS konfigurasi Anda.

Sebagai contoh, DNS konfigurasi berikut menggunakan fileA record.

```
A     | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Ubah ke DNS konfigurasi berikut untuk menggunakan CNAME catatan saja.

```
CNAME | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Solusi ini memungkinkan Anda mengarahkan domain apex (@ record) dengan benar ke layanan seperti CloudFront, sambil menghindari batasan IPv4 -only di sistem Cloudflare. A records

Saya menerima pesan CNAMEAlready ExistsException kesalahan

Jika Anda menerima pesan CNAMEAlreadyExistsExceptionkesalahan, artinya salah satu nama host yang Anda coba hubungkan (subdomain atau domain puncak) sudah di-deploy ke distribusi Amazon lain. CloudFront Sumber kesalahan Anda tergantung pada penyedia hosting dan DNS Anda saat ini.

CNAMEAlias, seperti `example.com` atau hanya `sub.example.com` dapat dikaitkan dengan CloudFront distribusi tunggal pada satu waktu. CNAMEAlreadyExistsExceptionIni menunjukkan bahwa domain Anda sudah dikaitkan dengan CloudFront distribusi lain, baik dalam yang sama Akun AWS, atau berpotensi di akun yang berbeda. Domain harus dipisahkan dari CloudFront distribusi sebelumnya sebelum distribusi baru yang dibuat oleh Amplify Hosting akan berfungsi. Anda mungkin perlu memeriksa lebih dari satu akun jika Anda atau organisasi Anda memiliki beberapa Akun AWS s.

Lakukan langkah-langkah berikut untuk mendiagnosis penyebab CNAMEAlreadyExistsExceptionkesalahan.

1. Masuk ke CloudFront Konsol Amazon dan verifikasi bahwa Anda tidak men-deploy domain tersebut ke distribusi lain. Satu CNAME catatan dapat dilampirkan ke satu CloudFront distribusi pada satu waktu.
2. Jika sebelumnya men-deploy domain ke CloudFront distribusi, Anda harus menghapusnya.
    a. Pilih Distribusi di menu navigasi kiri.
    b. Pilih nama distribusi yang akan diedit.
    c. Pilih tab Umum. Di bagian Pengaturan, pilih Edit.
    d. Hapus nama domain dari Nama domain alternatif (CNAME). Kemudian pilih, Simpan perubahan.
3. Konfirmasikan bahwa tidak ada CloudFront distribusi lain yang menggunakan domain ini di saat ini Akun AWS atau lainnya Akun AWS. Jika tidak mengganggu layanan yang sedang berjalan, coba hapus dan buat ulang zona yang dihosting.
4. Periksa untuk melihat apakah domain ini terhubung ke aplikasi Amplify lain milik Anda. Jika ya, pastikan Anda tidak mencoba untuk menggunakan kembali nama host tersebut. Jika Anda menggunakan `www.example.com` aplikasi lain, Anda tidak dapat menggunakan aplikasi `www.example.com` yang saat ini Anda hubungkan. Anda dapat menggunakan subdomain lain, seperti. `blog.example.com`

5. Jika domain berhasil terhubung ke aplikasi lain dan kemudian dihapus dalam satu jam terakhir, coba lagi setelah setidaknya satu jam. Jika Anda masih melihat pengecualian ini setelah 6 jam, hubungi Dukungan. Untuk informasi selengkapnya, lihat Membuat kasus dukungan di Panduan AWS Dukungan Pengguna.
6. Jika Anda mengelola domain Anda melalui Route 53, pastikan untuk membersihkan zona CNAME atau ALIAS catatan yang dihosting yang mengarah ke CloudFront distribusi lama.
7. Setelah menyelesaikan langkah-langkah sebelumnya, hapus domain kustom dari Amplify Hosting dan mulai lagi dengan alur kerja untuk menghubungkan domain kustom di konsol Amplify.

Saya mendapatkan kesalahan Verifikasi Tambahan yang Diperlukan

Jika Anda mendapatkan kesalahan Diperlukan Verifikasi Tambahan, ini berarti bahwa AWS Certificate Manager (ACM) memerlukan informasi tambahan untuk memproses permintaan sertifikat ini. Ini dapat terjadi sebagai tindakan perlindungan penipuan, seperti ketika peringkat domain dalam 1000 situs web Alexa teratas. Untuk memberikan informasi yang diperlukan, gunakan Support Center untuk menghubungi Dukungan. Jika Anda tidak memiliki rencana dukungan, posting thread baru di Forum Diskusi ACM.

> **Note**
>
> Anda tidak dapat meminta sertifikat untuk nama domain milik Amazon seperti yang diakhiri dengan amazonaws.com, cloudfront.net, atau elasticbeanstalk.com.

Saya mendapatkan kesalahan 404 pada URL CloudFront

Untuk melayani lalu lintas, Amplify Hosting menunjuk ke CloudFront URL melalui catatan CNAME. Dalam proses menghubungkan aplikasi ke domain kustom, konsol Amplify menampilkan CloudFront URL untuk aplikasi. Namun, Anda tidak dapat mengakses aplikasi Anda secara langsung menggunakan CloudFront URL ini. Ia mengembalikan kesalahan 404. Aplikasi Anda menyelesaikan hanya menggunakan URL aplikasi Amplify (misalnya,`https://main.d5udybEXAMPLE.amplifyapp.com`, atau domain kustom Anda (misalnya). `www.example.com`

Amplify perlu merutekan permintaan ke cabang yang diterapkan dengan benar dan menggunakan nama host untuk melakukan ini. Misalnya, Anda dapat mengonfigurasi domain `www.example.com`

yang mengarah ke cabang utama aplikasi, tetapi juga mengonfigurasi domain `dev.example.com` yang mengarah ke cabang dev aplikasi yang sama. Oleh karena itu, Anda harus mengunjungi aplikasi berdasarkan subdomain yang dikonfigurasi sehingga Amplify dapat merutekan permintaan yang sesuai.

Saya mendapatkan sertifikat SSL atau kesalahan HTTPS saat mengunjungi domain saya

Jika Anda memiliki catatan DNS Otorisasi Otoritas Sertifikat (CAA) yang dikonfigurasi dengan penyedia DNS pihak ketiga Anda, AWS Certificate Manager (ACM) mungkin tidak dapat memperbarui atau menerbitkan ulang sertifikat perantara untuk sertifikat SSL domain kustom Anda. Untuk mengatasinya, Anda perlu menambahkan catatan CAA untuk mempercayai setidaknya salah satu domain otoritas sertifikat Amazon. Prosedur berikut menjelaskan langkah-langkah yang perlu Anda lakukan.

Untuk menambahkan catatan CAA untuk mempercayai otoritas sertifikat Amazon

1. Konfigurasikan data CAA dengan penyedia domain Anda untuk mempercayai setidaknya salah satu domain otoritas sertifikat Amazon. Untuk informasi selengkapnya tentang mengonfigurasi catatan CAA, lihat masalah Otorisasi Otoritas Sertifikasi (CAA) di Panduan Pengguna.AWS Certificate Manager
2. Gunakan salah satu metode berikut untuk memperbarui sertifikat SSL Anda:
   - Perbarui secara manual menggunakan konsol Amplify.

     > **Note**
     >
     > Metode ini akan menyebabkan down time untuk domain kustom Anda.

     a. Masuk ke konsol AWS Management Console Amplify.
     b. Pilih aplikasi tempat data CAA akan ditambahkan.
     c. Di panel navigasi, pilih Pengaturan Aplikasi, Manajemen domain.
     d. Pada halaman manajemen Domain, hapus domain kustom.
     e. Hubungkan aplikasi Anda ke domain kustom lagi. Proses ini mengeluarkan sertifikat SSL baru dan sertifikat perantara sekarang dapat dikelola oleh ACM.

Untuk menyambungkan kembali aplikasi ke domain kustom, gunakan salah satu prosedur berikut yang sesuai dengan penyedia domain yang Anda gunakan.

- Menambahkan domain kustom yang dikelola Amazon Route 53.
- Menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga.
- Memperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy.

- Hubungi Dukungan agar sertifikat SSL Anda diterbitkan kembali.

Memecahkan masalah aplikasi sisi server

Jika Anda mengalami masalah tak terduga saat men-deploy aplikasi SSR dengan komputasi Amplify Hosting, tinjau topik pemecahan masalah berikut. Jika Anda tidak melihat solusi untuk masalah Anda di sini, lihat panduan pemecahan masalah komputasi web SSR di repositori Amplify Hosting Issues. GitHub

Topik

- Saya butuh bantuan menggunakan adaptor kerangka kerja
- Rute Edge API menyebabkan build Next.js saya gagal
- Regenerasi Statis Incremental On-Demand tidak berfungsi untuk aplikasi saya
- Output build aplikasi saya melebihi ukuran maksimum yang diizinkan
- Build saya gagal dengan kesalahan kehabisan memori
- Ukuran respons HTTP aplikasi saya terlalu besar
- Bagaimana cara mengukur waktu mulai aplikasi komputasi saya secara lokal?

Saya butuh bantuan menggunakan adaptor kerangka kerja

Jika Anda mengalami masalah saat menerapkan aplikasi SSR yang menggunakan adaptor kerangka kerja, lihat. Menggunakan adaptor open source untuk kerangka SSR apa pun

Rute Edge API menyebabkan build Next.js saya gagal

Saat ini, Amplify tidak mendukung Rute API Next.js Edge. Anda harus menggunakan non-edge APIs dan middleware saat menghosting aplikasi Anda dengan Amplify.

Regenerasi Statis Incremental On-Demand tidak berfungsi untuk aplikasi saya

Dimulai dengan versi 12.2.0, Next.js mendukung Incremental Static Regeneration (ISR) untuk membersihkan cache Next.js secara manual untuk halaman tertentu. Namun, Amplify saat ini tidak mendukung ISR On-Demand. Jika aplikasi Anda menggunakan validasi ulang sesuai permintaan Next.js, fitur ini tidak akan berfungsi saat Anda menerapkan aplikasi ke Amplify.

Output build aplikasi saya melebihi ukuran maksimum yang diizinkan

Saat ini, ukuran output build maksimum yang didukung Amplify untuk aplikasi SSR adalah 220 MB. Jika Anda mendapatkan pesan kesalahan yang menyatakan bahwa ukuran keluaran build aplikasi melebihi ukuran maksimum yang diizinkan, Anda harus mengambil langkah-langkah untuk menguranginya.

Untuk mengurangi ukuran keluaran build aplikasi, Anda dapat memeriksa artefak build aplikasi dan mengidentifikasi dependensi besar untuk diperbarui atau dihapus. Pertama, unduh artefak build ke komputer lokal Anda. Kemudian, periksa ukuran direktori. Misalnya, `node_modules` direktori mungkin berisi binari seperti `@swc` dan `@esbuild` yang direferensikan oleh file runtime server Next.js. Karena binari ini tidak diperlukan dalam runtime, Anda dapat menghapusnya setelah build.

Gunakan petunjuk berikut untuk mengunduh keluaran build aplikasi dan memeriksa ukuran direktori menggunakan (AWS Command Line Interface CLI).

Untuk mengunduh dan memeriksa keluaran build untuk aplikasi Next.js

1. Buka jendela terminal dan jalankan perintah berikut. Ubah id aplikasi, nama cabang, dan id pekerjaan menjadi informasi Anda sendiri. Untuk id pekerjaan, gunakan nomor build untuk build gagal yang sedang Anda selidiki.

   ```
   aws amplify get-job --app-id abcd1234 --branch-name main --job-id 2
   ```

2. Dalam output terminal, cari URL artefak yang telah ditetapkan sebelumnya di bagian`job`,`steps`,`stepName: "BUILD"`. URL disorot dengan warna merah di output contoh berikut.

   ```
   "job": {
       "summary": {
           "jobArn": "arn:aws:amplify:us-west-2:111122223333:apps/abcd1234/main/
   jobs/0000000002",
   ```

```
        "jobId": "2",
        "commitId": "HEAD",
        "commitTime": "2024-02-08T21:54:42.398000+00:00",
        "startTime": "2024-02-08T21:54:42.674000+00:00",
        "status": "SUCCEED",
        "endTime": "2024-02-08T22:03:58.071000+00:00"
    },
    "steps": [
        {
            "stepName": "BUILD",
            "startTime": "2024-02-08T21:54:42.693000+00:00",
            "status": "SUCCEED",
            "endTime": "2024-02-08T22:03:30.897000+00:00",
            "logUrl": "https://aws-amplify-prod-us-west-2-artifacts.s3.us-
west-2.amazonaws.com/abcd1234/main/0000000002/BUILD/log.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2V...Example
```

3. Salin dan tempel URL ke jendela browser. `artifacts.zip`File diunduh ke komputer lokal Anda. Ini adalah output build Anda.
4. Jalankan perintah penggunaan du disk untuk memeriksa ukuran direktori. Contoh perintah berikut mengembalikan ukuran `compute` dan `static` direktori.

```
du -csh compute static
```

Berikut ini adalah contoh dari output dengan informasi ukuran untuk `compute` dan `static` direktori.

```
 29M    compute
3.8M    static
 33M    total
```

5. Buka `compute` direktori, dan cari `node_modules` foldernya. Tinjau dependensi Anda untuk file yang dapat Anda perbarui atau hapus untuk mengurangi ukuran folder.
6. Jika aplikasi Anda menyertakan binari yang tidak diperlukan dalam runtime, hapus binari setelah build dengan menambahkan perintah berikut ke bagian build file aplikasi Anda. `amplify.yml`

```
- rm -f node_modules/@swc/core-linux-x64-gnu/swc.linux-x64-gnu.node
- rm -f node_modules/@swc/core-linux-x64-musl/swc.linux-x64-musl.node
```

Berikut ini adalah contoh bagian perintah build dari `amplify.yml` file dengan perintah ini ditambahkan setelah menjalankan build produksi.

```
frontend:
  phases:
    build:
      commands:
         -npm run build

         // After running a production build, delete the files
         - rm -f node_modules/@swc/core-linux-x64-gnu/swc.linux-x64-gnu.node
         - rm -f node_modules/@swc/core-linux-x64-musl/swc.linux-x64-musl.node
```

Build saya gagal dengan kesalahan kehabisan memori

Next.js memungkinkan Anda untuk menyimpan artefak build cache untuk meningkatkan kinerja pada build berikutnya. Selain itu, AWS CodeBuild container Amplify mengompres dan mengunggah cache ini ke Amazon S3, atas nama Anda, untuk meningkatkan kinerja build berikutnya. Ini dapat menyebabkan build Anda gagal dengan kesalahan kehabisan memori.

Lakukan tindakan berikut untuk mencegah aplikasi Anda melebihi batas memori selama fase build. Pertama, hapus `.next/cache/**/*` dari bagian cache.paths dari pengaturan build Anda. Selanjutnya, hapus variabel `NODE_OPTIONS` lingkungan dari file setelan build Anda. Sebagai gantinya, atur variabel `NODE_OPTIONS` lingkungan di konsol Amplify untuk menentukan batas memori maksimum Node. Untuk informasi selengkapnya tentang pengaturan variabel lingkungan menggunakan konsol Amplify, lihat. Mengatur variabel lingkungan

Setelah melakukan perubahan ini, coba build Anda lagi. Jika berhasil, tambahkan `.next/cache/**/*` kembali ke bagian cache.paths dari file pengaturan build Anda.

Untuk informasi selengkapnya tentang konfigurasi cache Next.js guna meningkatkan kinerja build, lihat AWS CodeBuild di situs web Next.js.

Ukuran respons HTTP aplikasi saya terlalu besar

Saat ini, ukuran respons maksimum yang didukung Amplify untuk Next.js 12 dan aplikasi yang lebih baru menggunakan platform Web Compute adalah 5,72 MB. Tanggapan atas batas itu mengembalikan 504 kesalahan tanpa konten ke klien.

Bagaimana cara mengukur waktu mulai aplikasi komputasi saya secara lokal?

Gunakan petunjuk berikut untuk menentukan waktu inisialisasi/mulai lokal untuk Next.js 12 atau aplikasi Compute yang lebih baru. Anda dapat membandingkan kinerja aplikasi secara lokal vs. di Amplify Hosting dan menggunakan hasilnya untuk meningkatkan kinerja aplikasi Anda.

Untuk mengukur waktu inisialisasi aplikasi Next.js Compute secara lokal

1. Buka `next.config.js` file aplikasi dan atur `output` opsi menjadi `standalone` sebagai berikut.

```
** @type {import('next').NextConfig} */
const nextConfig = {
  // Other options
  output: "standalone",
};

module.exports = nextConfig;
```

2. Buka jendela terminal dan jalankan perintah berikut untuk membangun aplikasi.

```
next build
```

3. Jalankan perintah berikut untuk menyalin `.next/static` folder ke`.next/standalone/.next/static`.

```
cp -r .next/static .next/standalone/.next/static
```

4. Jalankan perintah berikut untuk menyalin `public` folder ke`.next/standalone/public`.

```
cp -r public .next/standalone/public
```

5. Jalankan perintah berikut untuk memulai server Next.js.

```
node .next/standalone/server.js
```

6. Perhatikan berapa lama waktu yang dibutuhkan antara menjalankan perintah di langkah 5 dan server mulai. Ketika server mendengarkan pada port, itu harus mencetak pesan berikut.

```
Listening on port 3000
```

7. Perhatikan berapa lama waktu yang dibutuhkan modul lain untuk memuat setelah dimulainya server pada langkah 6. Misalnya, perpustakaan seperti `bugsnag` membutuhkan waktu 10-12 detik untuk memuat. Setelah dimuat, itu akan menampilkan pesan konfirmasi`[bugsnag] loaded`.
8. Tambahkan durasi waktu dari langkah 6 dan langkah 7 bersama-sama. Hasil ini adalah waktu inisialisasi/mulai lokal aplikasi Compute Anda.

Memecahkan masalah pengalihan dan penulisan ulang

Jika Anda mengalami masalah saat mengatur pengalihan dan penulisan ulang untuk aplikasi Amplify, periksa topik di bagian ini untuk bantuan.

Topik

Akses ditolak untuk rute tertentu bahkan dengan aturan pengalihan SPA.

Jika Anda mendapatkan kesalahan akses ditolak untuk rute tertentu dengan aturan pengalihan SPA, `baseDirectory` mungkin tidak disetel dengan benar di pengaturan build aplikasi. Misalnya, jika frontend aplikasi Anda dibuat ke `build` direktori, setelan build Anda juga harus mengarah ke `build` direktori. Contoh spesifikasi build berikut menunjukkan pengaturan ini.

```
frontend:
  artifacts:
    baseDirectory: build
    files:
        - "**/*"
```

Untuk contoh lengkap setelan spesifikasi build untuk aplikasi Amplify, lihat Sintaks YAMAL spesifikasi build

Saya ingin menyiapkan proxy terbalik ke API

Anda dapat menggunakan JSON berikut untuk mengatur proxy terbalik ke titik akhir dinamis.

```
[
  {
    "source": "/documents/<*>",
    "target": "https://otherdomain/resource/<*>",
    "status": "200",
    "condition": null
  }
]
```

Untuk contoh dasar membuat proxy terbalik untuk aplikasi Amplify Anda ke API pihak ketiga, lihat. Penulisan ulang proksi balik

Memecahkan masalah caching

Jika Anda mengalami masalah caching untuk aplikasi Amplify, konsultasikan topik tersebut di bagian ini untuk bantuan.

Topik

- Saya ingin mengurangi ukuran cache untuk aplikasi
- Saya ingin menonaktifkan membaca dari cache untuk suatu aplikasi

Saya ingin mengurangi ukuran cache untuk aplikasi

Jika Anda menggunakan cache, Anda mungkin menyimpan file perantara yang tidak dibersihkan di antara build. Caching file yang jarang digunakan ini akan meningkatkan ukuran cache Anda. Untuk mencegah hal ini, Anda dapat mengecualikan folder tertentu agar tidak di-cache dengan menggunakan `!` arahan di `cache` bagian spesifikasi build aplikasi Anda.

Contoh pengaturan build berikut menunjukkan cara menggunakan `!` direktif untuk menentukan folder yang tidak ingin Anda cache.

```
cache:
  paths:
    - node_modules/**/*
    - "!node_modules/path/not/to/cache"
```

Ketika Anda cache `node_modules` folder, `node_modules/.cache` dihilangkan secara default.

Untuk contoh lengkap setelan spesifikasi build untuk aplikasi Amplify, lihat Sintaks YAMAL spesifikasi build

Saya ingin menonaktifkan membaca dari cache untuk suatu aplikasi

Jika Anda ingin menonaktifkan pembacaan dari cache untuk aplikasi, hapus bagian cache dari spesifikasi build aplikasi Anda.

Menyiapkan akses Amplify ke repositori GitHub

Amplify sekarang menggunakan fitur GitHub Apps untuk mengotorisasi akses hanya-baca Amplify ke repositori. GitHub Dengan GitHub Aplikasi Amplify, izin lebih disesuaikan, memungkinkan Anda memberikan Amplify akses hanya ke repositori yang Anda tentukan. Untuk mempelajari lebih lanjut tentang GitHub Aplikasi, lihat Tentang GitHub Aplikasi di GitHub situs web.

Saat Anda menghubungkan aplikasi baru yang disimpan dalam GitHub repo, secara default Amplify menggunakan GitHub Aplikasi untuk mengakses repo. Namun, aplikasi Amplify yang sudah ada yang sebelumnya Anda sambungkan dari GitHub repo digunakan untuk OAuth akses. CI/CD akan terus berfungsi untuk aplikasi ini, tetapi kami sangat menyarankan Anda memigrasikannya untuk menggunakan Aplikasi Amplify yang baru. GitHub

Saat menerapkan aplikasi baru atau memigrasikan aplikasi yang sudah ada menggunakan konsol Amplify, Anda akan secara otomatis diarahkan ke lokasi penginstalan untuk Aplikasi Amplify. GitHub Untuk mengakses halaman arahan instalasi aplikasi secara manual, buka browser web dan navigasikan ke aplikasi berdasarkan wilayah. Gunakan format`https://github.com/apps/aws-amplify-`*`REGION`*, ganti *`REGION`* dengan wilayah tempat Anda akan menerapkan aplikasi Amplify. Misalnya, untuk menginstal GitHub Aplikasi Amplify di wilayah AS Barat (Oregon), navigasikan ke -2. https://github.com/apps/ aws-amplify-us-west

Topik

- Menginstal dan mengotorisasi Aplikasi GitHub Amplify untuk penerapan baru
- Memigrasi aplikasi yang ada ke OAuth Aplikasi Amplify GitHub
- Menyiapkan GitHub Aplikasi Amplify untuk penerapan, AWS CloudFormation CLI, dan SDK
- Menyiapkan pratinjau web dengan Aplikasi Amplify GitHub

Menginstal dan mengotorisasi Aplikasi GitHub Amplify untuk penerapan baru

Saat Anda menerapkan aplikasi baru ke Amplify dari kode yang ada di GitHub repo, gunakan petunjuk berikut untuk menginstal dan mengotorisasi Aplikasi. GitHub

Untuk menginstal dan mengotorisasi Aplikasi Amplify GitHub

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Dari halaman Semua aplikasi, pilih Aplikasi baru, lalu Host aplikasi web.
3. Pada halaman Memulai dengan Amplify Hosting, pilih GitHub, lalu pilih Lanjutkan.
4. Jika ini adalah pertama kalinya menghubungkan GitHub repositori, Halaman baru terbuka di browser Anda di GitHub .com, meminta izin untuk mengotorisasi AWS Amplify di akun Anda. GitHub Pilih Izinkan.
5. Selanjutnya, Anda harus menginstal GitHub Aplikasi Amplify di akun Anda GitHub . Halaman terbuka di GitHub.com meminta izin untuk menginstal dan mengotorisasi AWS Amplify di akun Anda. GitHub
6. Pilih GitHub akun tempat Anda ingin menginstal Aplikasi Amplify GitHub .
7. Lakukan salah satu tindakan berikut:
   - Untuk menerapkan instalasi ke semua repositori, pilih Semua repositori.
   - Untuk membatasi instalasi ke repositori tertentu yang Anda pilih, pilih Hanya pilih repositori. Pastikan untuk menyertakan repo untuk aplikasi yang Anda migrasi di repo yang Anda pilih.
8. Pilih Instal & Otorisasi.
9. Anda diarahkan ke halaman cabang Add repositori untuk aplikasi Anda di konsol Amplify.
10. Dalam daftar repositori yang baru diperbarui, pilih nama repositori yang akan dihubungkan.
11. Di daftar Cabang, pilih nama cabang repositori yang akan dihubungkan.
12. Pilih Berikutnya.
13. Pada halaman Konfigurasi pengaturan build, pilih Berikutnya.
14. Di halaman Tinjauan, pilih Simpan dan deploy.

Memigrasi aplikasi yang ada ke OAuth Aplikasi Amplify GitHub

Aplikasi Amplify yang ada yang sebelumnya Anda sambungkan dari GitHub repositori digunakan OAuth untuk akses repo. Kami sangat menyarankan Anda memigrasikan aplikasi ini untuk menggunakan Aplikasi GitHub Amplify.

Gunakan petunjuk berikut untuk memigrasikan aplikasi dan menghapus OAuth webhook yang sesuai di akun Anda GitHub . Perhatikan bahwa prosedur migrasi bervariasi tergantung pada apakah aplikasi GitHub Amplify sudah diinstal. Setelah memigrasikan aplikasi pertama dan menginstal serta mengotorisasi GitHub Aplikasi, Anda hanya perlu memperbarui izin repositori untuk migrasi aplikasi berikutnya.

Untuk memigrasikan aplikasi dari OAuth ke Aplikasi GitHub

1. Masuk ke, lalu buka AWS Management Console Konsol Amplify.
2. Pilih aplikasi yang akan di-migrasikan.
3. Di halaman informasi aplikasi, cari pesan biru Migrasi ke GitHub Aplikasi kami dan pilih Mulai migrasi.
4. Pada halaman Instal dan otorisasi GitHub Aplikasi, pilih Konfigurasi GitHub Aplikasi.
5. Halaman baru terbuka di browser Anda di GitHub .com, meminta izin untuk mengotorisasi AWS Amplify di akun Anda GitHub . Pilih Izinkan.
6. Pilih GitHub akun tempat Anda ingin menginstal Aplikasi Amplify GitHub .
7. Lakukan salah satu tindakan berikut:
   - Untuk menerapkan instalasi ke semua repositori, pilih Semua repositori.
   - Untuk membatasi instalasi ke repositori tertentu yang Anda pilih, pilih Hanya pilih repositori. Pastikan untuk menyertakan repo untuk aplikasi yang Anda migrasi di repositori yang Anda pilih.
8. Pilih Instal & Otorisasi.
9. Anda diarahkan ke halaman Instal dan otorisasi GitHub Aplikasi untuk aplikasi Anda di konsol Amplify. Jika GitHub otorisasi berhasil, Anda akan melihat pesan sukses. Pilih, Berikutnya.
10. Pada halaman Instalasi lengkap, pilih Instalasi lengkap. Langkah ini menghapus webhook yang ada, membuat yang baru, dan menyelesaikan migrasi.

Menyiapkan GitHub Aplikasi Amplify untuk penerapan, AWS CloudFormation CLI, dan SDK

Aplikasi Amplify yang ada yang sebelumnya Anda sambungkan dari GitHub repositori digunakan OAuth untuk akses repo. Ini dapat mencakup aplikasi yang Anda gunakan menggunakan Amplify Command Line Interface (CLI) AWS CloudFormation,, atau file. SDKs Kami sangat menyarankan Anda memigrasikan aplikasi ini untuk menggunakan Aplikasi GitHub Amplify yang baru. Migrasi harus dilakukan di konsol Amplify di. AWS Management Console Untuk petunjuk, lihat Memigrasi aplikasi yang ada ke OAuth Aplikasi Amplify GitHub .

Anda dapat menggunakan AWS CloudFormation, Amplify CLI, dan SDKs untuk menerapkan aplikasi Amplify baru yang menggunakan Aplikasi untuk akses repo. GitHub Proses ini mengharuskan Anda menginstal GitHub Aplikasi Amplify terlebih dahulu di akun Anda GitHub . Selanjutnya, Anda perlu membuat token akses pribadi di GitHub akun Anda. Terakhir, men-deploy aplikasi dan tentukan token akses pribadi.

Instal GitHub Aplikasi Amplify di akun Anda

1. Buka browser web dan arahkan ke lokasi penginstalan untuk GitHub Aplikasi Amplify di AWS Wilayah tempat Anda akan menerapkan aplikasi.

   Gunakan format`https://github.com/apps/aws-amplify-REGION/installations/new`, ganti *REGION* dengan input Anda sendiri. Misalnya, jika Anda menginstal aplikasi di wilayah US West (Oregon), tentukan`https://github.com/apps/aws-amplify-us-west-2/installations/new`.
2. Pilih GitHub akun tempat Anda ingin menginstal aplikasi Amplify GitHub .
3. Lakukan salah satu tindakan berikut:
   - Untuk menerapkan instalasi ke semua repositori, pilih Semua repositori.
   - Untuk membatasi instalasi ke repositori tertentu yang Anda pilih, pilih Hanya pilih repositori. Pastikan untuk menyertakan repo untuk aplikasi yang Anda migrasi di repo yang Anda pilih.
4. Pilih Instal.

Buat token akses pribadi di GitHub akun Anda

1. Masuk ke GitHub akun Anda.
2. Di sudut kanan atas, cari foto profil Anda dan pilih Pengaturan dari menu.

3. Di menu navigasi kiri, pilih Pengaturan pengembang.
4. Pada halaman GitHub Aplikasi, di menu navigasi kiri, pilih Token akses pribadi.
5. Pada halaman Token akses pribadi, pilih Hasilkan token baru.
6. Pada halaman token akses pribadi baru, untuk Catatan masukkan nama deskriptif untuk token.
7. Di bagian Pilih cakupan, pilih admin:repo_hook.
8. Pilih Hasilkan token.
9. Salin dan simpan token akses pribadi. Anda harus menyediakannya saat Anda menerapkan aplikasi Amplify dengan CLI AWS CloudFormation,, atau. SDKs

Setelah GitHub aplikasi Amplify diinstal di GitHub akun Anda dan Anda telah membuat token akses pribadi, Anda dapat menerapkan aplikasi baru dengan Amplify CLI,, atau. AWS CloudFormation SDKs Gunakan `accessToken` bidang ini untuk menentukan token akses pribadi yang Anda buat di prosedur sebelumnya. Untuk informasi selengkapnya, lihat CreateAppdi referensi Amplify API dan AWS::Amplify::Appdi AWS CloudFormation Panduan Pengguna.

Perintah CLI berikut menerapkan aplikasi Amplify baru yang menggunakan Aplikasi untuk akses repositori. GitHub Ganti *myapp-using-githubapphttps://github.com/Myaccount/react-app*,, dan *MY_TOKEN* dengan informasi Anda sendiri.

```
aws amplify create-app --name myapp-using-githubapp --repository https://github.com/
Myaccount/react-app --access-token MY_TOKEN
```

Menyiapkan pratinjau web dengan Aplikasi Amplify GitHub

Pratinjau web men-deploy setiap permintaan tarik (PR) pada GitHub repositori Anda ke URL pratinjau unik. Pratinjau sekarang menggunakan Aplikasi GitHub Amplify untuk akses ke GitHub repo Anda. Untuk petunjuk tentang menginstal dan mengotorisasi GitHub App untuk pratinjau web, lihat. Aktifkan pratinjau web untuk permintaan tarik

AWS Amplify Referensi hosting

Gunakan topik di bagian ini untuk menemukan bahan referensi terperinci untuk AWS Amplify.

Topik

AWS CloudFormation dukungan

Gunakan AWS CloudFormation template untuk menyediakan sumber daya Amplify, memungkinkan penerapan aplikasi web yang dapat berulang dan andal. AWS CloudFormation menyediakan bahasa umum bagi Anda untuk mendeskripsikan dan menyediakan semua sumber daya infrastruktur di lingkungan cloud Anda dan menyederhanakan peluncuran di beberapa AWS akun dan/atau wilayah hanya dengan beberapa klik.

Untuk Amplify Hosting, lihat dokumentasi Amplify. CloudFormation Untuk Amplify Studio, lihat dokumentasi Amplify UI Builder. CloudFormation

AWS Command Line Interface dukungan

Gunakan AWS Command Line Interface untuk membuat aplikasi Amplify secara terprogram dari baris perintah. Untuk informasi, lihat AWS CLI dokumentasi.

Dukungan penandaan sumber daya

Anda dapat menggunakan AWS Command Line Interface untuk menandai sumber daya Amplify. Untuk informasi lebih lanjut, lihat dokumentasi sumber daya tanda AWS CLI.

Amplify Hosting API

Referensi ini memberikan deskripsi tindakan dan tipe data untuk Amplify Hosting API. Untuk informasi selengkapnya, lihat dokumentasi referensi Amplify API.

Riwayat dokumen untuk AWS Amplify

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak perilisan terakhir AWS Amplify.

- Pembaruan dokumentasi terbaru: 28 Mei 2025

Perubahan	Deskripsi	Tanggal
Pengaturan Build dan konfigurasi chapter yang diperbarui	Memperbarui Mengelola konfigurasi build untuk aplikasi Amplify chapter untuk menjelaskan fitur tipe instans build baru yang dapat dikonfigurasi yang memungkinkan Anda memilih tipe instans yang menyediakan sumber daya CPU, memori, dan ruang disk yang dibutuhkan aplikasi Anda.	28 Mei 2025
Pembaruan Bab Firewall	Memperbarui Dukungan firewall untuk Amplify situs yang dihosting chapter untuk menjelaskan ketersediaan umum (GA) integrasi Amplify dengan AWS WAF, termasuk fungsionalitas GA dan struktur harga.	26 Maret 2025
Bab perlindungan Skew baru	Menambahkan Perlindungan miring untuk penerapan Amplify chapter untuk menjelaskan fitur perlindungan miring yang menghilangkan masalah kemiringan	10 Maret 2025

Perubahan	Deskripsi	Tanggal
	versi antara klien dan server di aplikasi web Amplify.	
Pembaruan Bab Webhooks	Menambahkan Webhook terpadu untuk repositori Git topik untuk menjelaskan fitur webhook terpadu yang menggunakan satu webhook komprehensif untuk semua aplikasi Amplify yang terkait dengan satu repositori Git.	10 Maret 2025
Baru Menambahkan peran Komputasi SSR untuk memungkinkan akses ke AWS topik sumber daya	Menambahkan Menambahkan peran SSR Compute untuk memungkinkan akses ke sumber daya AWS topik untuk menjelaskan cara membuat dan mengaitkan peran Amplify SSR Compute dengan aplikasi untuk memberikan akses layanan Amplify Compute ke resource. AWS	17 Februari 2025
Penggunaan Baru AWS WAF untuk melindungi bagian aplikasi Amplify Anda	Menambahkan Dukungan firewall untuk Amplify situs yang dihosting chapter untuk menjelaskan integrasi Amplify dengan AWS WAF (dalam pratinjau) yang memungkinkan Anda untuk melindungi aplikasi web Anda dengan daftar kontrol akses web (web ACL).	18 Desember 2024

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	14 November 2024
Dukungan Amplify untuk topik Next.js	Memperbarui Amplify dukungan untuk Next.js topik untuk menjelaskan dukungan Amplify untuk Next.js versi 15.	6 November 2024
Baru Menerapkan situs web statis untuk Amplify dari Amazon S3 chapter	Menambahkan Menerapkan situs web statis untuk Amplify dari bucket Amazon S3 chapter untuk menjelaskan integrasi baru Amplify dengan Amazon S3 yang memungkinkan Anda meng-host konten situs web statis yang disimpan S3 hanya dengan beberapa klik.	16 Oktober 2024
Bab konfigurasi cache Mengelola Baru	Menambahkan Mengelola konfigurasi cache untuk aplikasi chapter untuk menjelaskan perilaku caching default Amplify dan bagaimana menerapkan kebijakan cache terkelola ke konten.	13 Agustus 2024

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	18 Juli 2024
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	31 Mei 2024
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	17 April 2024
Pembaruan Bab memulai	Diperbarui Memulai dengan menerapkan aplikasi ke Amplify Hosting chapter untuk menggunakan contoh aplikasi Next.js dalam tutorial.	12 April 2024
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	5 April 2024

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	4 April 2024
Bab Penyelesaian Masalah Baru	Menambahkan Memecahkan Masalah Amplify chapter untuk menjelaskan cara memperbaiki masalah yang Anda temui dengan aplikasi yang digunakan untuk Amplify Hosting.	2 April 2024
Dukungan baru untuk sertifikat SSL/TLS kustom	Menambahkan Menggunakan sertifikat SSL/TLS topik ke Menghubungkan domain kustom chapter untuk menjelaskan dukungan Amplify untuk sertifikat SSL/TLS kustom saat menghubungkan aplikasi ke domain kustom.	20 Februari 2024
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	2 Januari 2024

Perubahan	Deskripsi	Tanggal
Dukungan baru untuk kerangka SSR	Memperbarui Menyebarkan aplikasi yang dirender sisi server dengan Amplify Hosting topik untuk menjelaskan dukungan Amplify untuk kerangka kerja SSR berbasis JavaScript apa pun dengan adaptor sumber terbuka.	19 November 2023
Dukungan baru untuk peluncuran fitur pengoptimalan gambar	Menambahkan Pengoptimalan gambar untuk aplikasi SSR topik untuk menjelaskan dukungan bawaan untuk pengoptimalan gambar untuk aplikasi yang dirender sisi server.	19 November 2023
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	17 November 2023
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	6 November 2023

Perubahan	Deskripsi	Tanggal
Topik subdomain wildcard baru	Menambahkan Menyiapkan subdomain topik untuk menjelaskan dukungan untuk subdomain wildcard pada domain kustom.	6 November 2023
Kebijakan terkelola baru	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan kebijakan AmplifyBackendDeployFullAccess AWS terkelola baru untuk Amplify.	8 Oktober 2023
Dukungan baru untuk peluncuran fitur kerangka monorepo	Memperbarui Mengonfigurasi pengaturan build monorepo topik untuk menjelaskan dukungan penerapan aplikasi di monorepos yang dibuat menggunakan npm workspace, pnpm workspace, Yarn workspace, Nx, dan Turborepo.	19 Juni 2023
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	1 Juni 2023

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	24 Februari 2023
Pembaruan Bab rendering sisi server	Memperbarui Menyebarkan aplikasi yang dirender sisi server dengan Amplify Hosting chapter untuk menjelask an perubahan terbaru pada dukungan Amplify untuk Next.js versi 12 dan 13.	17 November 2022
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	30 Agustus 2022
Topik kebijakan terkelola diperbarui	Memperbarui Membangun backend untuk aplikasi topik untuk menjelaskan cara menerapkan backend menggunakan Amplify Studio.	23 Agustus 2022
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	27 April 2022

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	17 April 2022
Peluncuran fitur GitHub Aplikasi Baru	Menambahkan Menyiapkan akses Amplify ke repositori GitHub topik untuk menjelask an GitHub Aplikasi baru untuk mengotorisasi akses Amplify ke GitHub repositori Anda.	5 April 2022
Peluncuran fitur Amplify Studio baru	Memperbarui Selamat datang di AWS Amplify Hosting topik untuk menjelaskan pembaruan pada Amplify Studio yang menyediakan desainer visual untuk membuat komponen UI yang dapat Anda sambungkan ke data backend Anda.	2 Desember 2021
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify guna mendukung Amplify Studio.	2 Desember 2021

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	8 November 2021
Topik kebijakan terkelola diperbarui	Memperbarui AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	27 September 2021
Topik kebijakan terkelola baru	Penambahan AWS kebijakan terkelola untuk AWS Amplify topik untuk menjelaskan kebijakan AWS terkelola untuk Amplify dan perubahan terbaru terhadap kebijakan tersebut.	28 Juli 2021
Pembaruan Bab rendering sisi server	Pembaruan bab Menyebarkan aplikasi yang dirender sisi server dengan Amplify Hosting untuk menjelaskan dukungan baru untuk Next.js versi 10.x.x dan Next.js versi 11.	22 Juli 2021

Perubahan	Deskripsi	Tanggal
Pembaruan Bab konfigurasi pengaturan build	Penambahan topik Mengonfigurasi pengaturan build monorepo untuk menjelaskan cara mengonfigurasi pengaturan build dan variabel lingkungan `AMPLIFY_MONOREPO_APP_ROOT` baru ketika men-deploy aplikasi monorepo dengan Amplify.	20 Juli 2021

Perubahan	Deskripsi	Tanggal
Pembaruan Bab deployment cabang fitur	Penambahan topik Pembuatan waktu pembuatan otomatis konfigurasi Amplify (hanya aplikasi Gen 1) untuk menjelaskan cara menghasilkan secara otomatis file `aws-exports.js` pada waktu build. Penambahan topik Build backend bersyarat (hanya aplikasi Gen 1) untuk menjelaskan cara mengaktifkan build backend bersyarat. Penambahan topik Gunakan backend Amplify di seluruh aplikasi (hanya aplikasi Gen 1) untuk menjelaskan cara menggunakan kembali backend yang ada ketika Anda membuat aplikasi baru, menghubungkan cabang baru ke aplikasi yang ada, atau memperbarui frontend yang ada agar mengarah ke lingkungan backend yang berbeda.	30 Juni 2021
Pembaruan Bab keamanan	Penambahan topik Perlindungan Data di Amplify untuk menjelaskan cara menerapkan model tanggung jawab bersama dan cara Amplify menggunakan enkripsi untuk melindungi data at rest dan in transit.	3 Juni 2021

Perubahan	Deskripsi	Tanggal
Dukungan baru untuk peluncuran fitur SSR	Penambahan bab Menyebarkan aplikasi yang dirender sisi server dengan Amplify Hosting untuk menjelaskan dukungan Amplify untuk aplikasi web yang menggunakan rendering sisi server (SSR) dan dibuat dengan Next.js.	18 Mei 2021
Bab keamanan baru	Penambahan bab Keamanan di Amplify untuk menjelaskan cara menerapkan model tanggung jawab bersama saat menggunakan Amplify dan cara mengonfigurasi Amplify untuk memenuhi tujuan keamanan dan kepatuhan Anda.	26 Maret 2021
Pembaruan topik build kustom	Pembaruan topik Gambar build kustom dan pembaruan paket langsung untuk menjelaskan cara mengonfigurasi gambar build kustom yang di-hosting di Amazon Elastic Container Registry Public.	12 Maret 2021
Pembaruan topik pemantauan	Pembaruan topik Pemantauan untuk menjelaskan cara mengakses data CloudWatch metrik Amazon dan mengatur alarm.	2 Februari 2021

Perubahan	Deskripsi	Tanggal
Topik CloudTrail pencatatan log baru	Penambahan AWS CloudTrail topik Mencatat log panggilan API Amplify menggunakan untuk menjelaskan cara AWS CloudTrail menangkap dan mencatat semua tindakan API untuk Referensi API Konsol dan Referensi API AWS Amplify Konsol dan Referensi API UI AWS Amplify Admin.	2 Februari 2021
Peluncuran fitur UI Admin baru	Pembaruan topik Selamat datang di AWS Amplify Hosting untuk menjelaskan UI Admin baru yang menyediakan antarmuka visual bagi developer seluler dan web frontend untuk membuat dan mengelola backend aplikasi di luar AWS Management Console.	1 Desember 2020
Peluncuran fitur mode kinerja baru	Pembaruan topik Mengelola kinerja aplikasi topik untuk menjelaskan cara mengaktifkan mode kinerja guna mengoptimalkan kinerja hosting yang lebih cepat.	4 November 2020
Pembaruan topik header kustom	Pembaruan topik Header kustom untuk menjelaskan cara menentukan header kustom untuk aplikasi Amplify menggunakan konsol atau dengan mengedit file YML.	28 Oktober 2020

Perubahan	Deskripsi	Tanggal
Peluncuran fitur subdomain otomatis baru	Penambahan topik Mengatur subdomain otomatis untuk domain kustom Route 53 untuk menjelaskan cara menggunakan deploymen t cabang fitur berbasis pola untuk aplikasi yang terhubung ke domain kustom Amazon Route 53. Penambahan topik Akses pratinjau web dengan subdomain untuk menjelask an cara mengatur pratinjau web dari permintaan tarik agar dapat diakses dengan subdomain.	20 Juni 2020
Topik notifikasi baru	Penambahan topik Notifikas i untuk menjelaskan cara mengatur notifikasi email untuk aplikasi Amplify untuk memberi tahu pemangku kepentingan atau anggota tim ketika build berhasil atau gagal.	20 Juni 2020

Perubahan	Deskripsi	Tanggal
Pembaruan topik domain kustom	Pembaruan Menghubungkan domain kustom topik untuk meningkatkan prosedur penambahan domain kustom di Amazon Route 53 GoDaddy, dan Google Domains. Pembaruan ini juga mencakup informasi pemecahan masalah baru untuk menyiapkan domain kustom.	12 Mei 2020
AWS Amplify rilis	Rilis ini memperkenalkan Amplify.	26 November 2018

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.