

Panduan Pengguna

# AWS Amplify Hosting



# AWS Amplify Hosting: Panduan Pengguna

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan milik dari pemiliknya masing-masing, yang mungkin berafiliasi dengan, terhubung ke, atau disponsori oleh Amazon.

---

# Table of Contents

Apa itu AWS Amplify Hosting? .....	1
Kerangka kerja yang didukung .....	1
Amplify fitur Hosting .....	2
Memulai Amplify Hosting .....	2
Membangun backend .....	3
Harga Amplify Hosting .....	3
Memulai .....	4
Prasyarat .....	4
Langkah 1: Hubungkan repositori .....	4
Langkah 2: Konfirmasikan pengaturan build .....	5
Langkah 3: Deploy aplikasi .....	6
Langkah 4: (Opsional) membersihkan sumber daya .....	7
Menambahkan fitur ke aplikasi Anda .....	7
Rendering sisi server (SSR) .....	9
Definisi rendering sisi server .....	9
Dukungan kerangka kerja SSR .....	10
Menerapkan aplikasi SSR untuk Amplify .....	11
Menggunakan adaptor kerangka kerja .....	12
Menggunakan spesifikasi penerapan .....	13
Spesifikasi penyebaran .....	13
Menyebarkan server Express .....	37
Pengoptimalan gambar untuk aplikasi SSR .....	44
Menggunakan pemuat gambar khusus .....	44
Integrasi optimasi gambar untuk penulis kerangka kerja .....	44
Memahami API Pengoptimalan Gambar .....	45
Dukungan versi Node.js untuk aplikasi Next.js .....	54
Memecahkan masalah penerapan SSR .....	54
Anda menggunakan adaptor kerangka kerja .....	54
Rute Edge API menyebabkan build Next.js Anda gagal .....	55
Regenerasi Statis Incremental On-Demand tidak berfungsi untuk aplikasi Anda .....	55
Output build aplikasi Anda melebihi ukuran maksimum yang diizinkan .....	55
Build Anda gagal dengan kesalahan kehabisan memori .....	57
Ukuran respons HTTP terlalu besar .....	58
Amplify dukungan untuk Next.js .....	58

Dukungan fitur Next.js .....	58
Harga untuk aplikasi Next.js .....	60
Menerapkan aplikasi Next.js dengan Amplify .....	60
Migrasi aplikasi Next.js 11 ke komputasi Amplify Hosting .....	63
Menambahkan fungsionalitas SSR ke aplikasi Next.js statis .....	64
Membuat variabel lingkungan dapat diakses oleh runtime sisi server .....	66
Menerapkan aplikasi Next.js di monorepo .....	69
CloudWatch Log Amazon untuk aplikasi SSR .....	69
Amplify dukungan Next.js 11 .....	70
Menyiapkan domain khusus .....	79
Memahami terminologi dan konsep DNS .....	80
Terminologi DNS .....	80
Verifikasi DNS .....	81
Amplify Hosting proses aktivasi domain kustom .....	81
Menggunakan sertifikat SSL/TLS .....	82
Menambahkan domain kustom yang dikelola Amazon Route 53 .....	83
Menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga .....	84
Memperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy .....	90
Memperbarui data DNS untuk domain yang dikelola oleh Google Domains .....	93
Memperbarui sertifikat SSL/TLS untuk domain .....	96
Mengelola subdomain .....	97
Cara menambahkan subdomain saja .....	97
Cara menambahkan subdomain multilevel .....	98
Cara menambahkan atau mengedit subdomain .....	98
Subdomain wildcard .....	99
Untuk menambah atau menghapus subdomain wildcard .....	99
Mengatur subdomain otomatis untuk domain kustom Amazon Route 53 .....	100
Pratinjau web dengan subdomain .....	101
Memecahkan masalah domain kustom .....	101
Bagaimana cara memverifikasi bahwa CNAME saya berhasil ditetapkan? .....	102
Domain saya yang di-host dengan pihak ketiga terus menampilkan status Menunggu Verifikasi .....	102
Domain saya yang di-host dengan Amazon Route 53 terus menampilkan status Menunggu Verifikasi .....	103
Saya mendapatkan kesalahan CNAME AlreadyExistsException .....	104
Saya mendapatkan kesalahan Verifikasi Tambahan yang Diperlukan .....	105

Saya mendapatkan kesalahan 404 pada URL CloudFront .....	105
Saya mendapatkan sertifikat SSL atau kesalahan HTTPS saat mengunjungi domain saya ..	106
Mengonfigurasi pengaturan build .....	108
Buat perintah dan pengaturan spesifikasi .....	108
Pengaturan build khusus cabang .....	111
Menavigasi ke subfolder .....	112
Menerapkan backend dengan ujung depan untuk aplikasi Gen 1 .....	112
Menyiapkan folder output .....	113
Menginstal paket sebagai bagian dari build .....	113
Menggunakan registri npm privat .....	113
Menginstal paket OS .....	114
Penyimpanan kunci-nilai untuk setiap build .....	114
Melompati build untuk penerapan .....	114
Menonaktifkan build otomatis .....	115
Mengaktifkan atau menonaktifkan build dan deploy frontend berbasis diff .....	115
Mengaktifkan atau menonaktifkan build backend berbasis diff untuk aplikasi Gen 1 .....	116
Pengaturan build monorepo .....	117
Sintaks YAML spesifikasi build monorepo .....	117
Mengatur variabel lingkungan AMPLIFY_MONOREPO_APP_ROOT .....	120
Mengkonfigurasi aplikasi Turborepo dan pnpm monorepo .....	122
Deployment cabang fitur .....	124
Alur kerja tim dengan aplikasi Amplify Gen 2 fullstack .....	125
Alur kerja tim dengan aplikasi Amplify Gen 1 fullstack .....	125
Alur kerja cabang fitur .....	125
GitFlow alur kerja .....	132
Sandbox per developer .....	132
Deployment cabang fitur berbasis pola .....	134
Deployment cabang fitur berbasis pola untuk aplikasi yang terhubung ke domain kustom ....	135
Pembuatan waktu pembuatan otomatis konfigurasi Amplify (hanya aplikasi Gen 1) .....	135
Build backend bersyarat (hanya aplikasi Gen 1) .....	137
Gunakan backend Amplify di seluruh aplikasi (hanya aplikasi Gen 1) .....	138
Menggunakan kembali backend saat membuat aplikasi baru .....	138
Menggunakan kembali backend saat menghubungkan cabang ke aplikasi yang ada .....	139
Mengedit frontend yang ada agar mengarah ke backend berbeda .....	140
Membangun backend .....	141
Buat backend untuk aplikasi Gen 2 .....	141

Buat backend untuk aplikasi Gen 1 .....	141
Prasyarat .....	141
Langkah 1: Menyebarkan frontend .....	142
Langkah 2: Buat backend .....	143
Langkah 3: Hubungkan backend ke frontend .....	144
Langkah selanjutnya .....	146
Deployment manual .....	147
Seret dan lepas penerapan manual .....	147
Penyebaran manual Amazon S3 atau URL .....	148
Memecahkan masalah akses bucket Amazon S3 .....	149
Tombol deploy satu klik .....	150
Tambahkan tombol Deploy to Amplify Hosting ke repositori atau blog .....	150
Menyiapkan GitHub akses .....	151
Menginstal dan mengotorisasi GitHub Aplikasi Amplify untuk penerapan baru .....	151
Memigrasi aplikasi yang ada ke OAuth Aplikasi Amplify GitHub .....	152
Menyiapkan GitHub Aplikasi Amplify untuk AWS CloudFormation, CLI, dan penerapan SDK ....	153
Menyiapkan pratinjau web dengan GitHub Aplikasi Amplify .....	155
Pratinjau permintaan tarik .....	156
Mengaktifkan pratinjau web .....	156
Akses pratinjau web dengan subdomain .....	158
end-to-end Pengujian E .....	159
Tutorial: Mengatur end-to-end tes dengan Cypress .....	159
Tambahkan pengujian ke aplikasi Amplify yang ada .....	159
Menonaktifkan pengujian .....	161
Menggunakan pengalihan .....	163
Jenis pengalihan .....	163
Membuat dan mengedit pengalihan .....	164
Urutan pengalihan .....	165
Parameter kueri .....	166
Pengalihan dan penulisan ulang sederhana .....	166
Pengalihan untuk aplikasi web halaman tunggal (SPA) .....	168
Penulisan ulang proksi balik .....	169
Garis miring di belakang dan URL bersih .....	169
Placeholder .....	170
String kueri dan parameter path .....	170
Pengalihan berbasis wilayah .....	171

Ekspresi wildcard dalam pengalihan dan penulisan ulang .....	172
Membatasi akses .....	173
Variabel-variabel lingkungan .....	174
Amplify variabel lingkungan .....	174
Tetapkan variabel lingkungan .....	180
Akses variabel lingkungan pada waktu pembuatan .....	181
Membuat variabel lingkungan dapat diakses oleh runtime sisi server .....	181
Membuat lingkungan backend baru dengan parameter autentikasi untuk masuk sosial .....	182
Variabel lingkungan kerangka kerja frontend .....	183
Rahasia lingkungan .....	183
Tetapkan rahasia lingkungan .....	184
Akses rahasia lingkungan .....	184
Amplify rahasia lingkungan .....	184
Header kustom .....	186
Format YAML header kustom .....	186
Mengatur header kustom .....	187
Migrasi header kustom .....	189
Header kustom monorepo .....	190
Contoh header keamanan .....	191
Header Cache-Control Kustom .....	191
Webhook masuk .....	193
Pemantauan .....	194
Pemantauan CloudWatch dengan .....	194
Metrik .....	194
Alarm .....	197
CloudWatch Log Amazon untuk aplikasi SSR .....	198
Log akses .....	199
Menganalisis log akses .....	200
Membangun notifikasi .....	201
Mengatur notifikasi email .....	201
Build kustom .....	202
Gambar build kustom .....	202
Ketentuan gambar build kustom .....	202
Mengonfigurasi gambar build kustom .....	203
Pembaruan paket langsung .....	204
Mengonfigurasi pembaruan paket langsung .....	204

Menambahkan peran layanan .....	206
Membuat peran layanan .....	206
Pencegahan Deputi Bingung .....	207
Mengelola performa aplikasi .....	208
Menggunakan header untuk mengendalikan durasi cache .....	208
Menyetel header cache-control untuk meningkatkan performa aplikasi .....	208
Mencatat log panggilan API Amplify dengan AWS CloudTrail .....	210
Informasi Amplify di CloudTrail .....	210
Memahami entri berkas log Amplify .....	211
Keamanan .....	215
Identity and Access Management .....	215
Audiens .....	216
Mengautentikasi dengan identitas .....	217
Mengelola akses menggunakan kebijakan .....	220
Cara Amplify bekerja dengan IAM .....	223
Contoh kebijakan berbasis identitas .....	230
Kebijakan terkelola AWS .....	234
Pemecahan Masalah .....	247
Perlindungan Data .....	249
Enkripsi saat istirahat .....	250
Enkripsi bergerak .....	251
Pengelolaan kunci enkripsi .....	251
Validasi Kepatuhan .....	251
Keamanan Infrastruktur .....	253
Pencatatan dan pemantauan .....	253
Pencegahan confused deputy lintas layanan .....	254
Praktik terbaik keamanan .....	256
Menggunakan cookie dengan domain default Amplify .....	256
Kuota .....	258
Pemecahan Masalah .....	261
Masalah umum .....	261
Kode status HTTP 429 (Terlalu banyak permintaan) .....	261
AL2023 membangun gambar .....	262
Bagaimana cara menjalankan fungsi Amplify dengan runtime Python .....	262
Bagaimana cara menjalankan perintah yang memerlukan hak superuser atau root .....	263
Domain kustom .....	263



---

Rendering sisi server (SSR) .....	263
AWS AmplifyReferensi hosting .....	264
AWS CloudFormationDukungan .....	264
Dukungan AWS Command Line Interface .....	264
Dukungan penandaan sumber daya .....	264
Memperkuat Hosting API .....	264
Riwayat dokumen .....	265
.....	cclxxvii

# Selamat datang di AWS Amplify Hosting

Amplify Hosting menyediakan alur kerja berbasis Git untuk hosting aplikasi web tanpa server full-stack dengan penerapan berkelanjutan. Amplify menerapkan aplikasi Anda ke jaringan pengiriman konten AWS global (CDN). Panduan pengguna ini memberikan informasi yang Anda butuhkan untuk memulai dengan 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
- Astro dengan adaptor komunitas
- SvelteKit Dengan Adaptor Komunitas
- Kerangka SSR apa pun dengan adaptor khusus

### Kerangka kerja SPA

- Bereaksi
- Sudut
- Vue.js
- Ionik
- Ember

### Generator situs statis

- Kesebelas
- Gatsby
- Hugo
- Jekyll

- VuePress

## Amplify fitur Hosting

### [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 E](#)

Tingkatkan kualitas aplikasi Anda 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.

### Penerapan atom

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

## Memulai Amplify Hosting

Untuk memulai dengan Amplify Hosting, lihat tutorialnya. [Memulai dengan Amplify Hosting](#) Setelah menyelesaikan tutorial, Anda akan tahu cara menghubungkan aplikasi web di repositori Git (GitHub,, BitBucket GitLab, atau AWS CodeCommit) dan menyebarkannya 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.

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 Hosting

AWS Amplify adalah bagian dari AWS Tingkat Gratis. Anda dapat memulai secara gratis, lalu membayar saat Anda pergi setelah Anda melebihi batas Tingkat Gratis. [Untuk informasi tentang biaya Amplify Hosting, lihat AWS Amplify Harga.](#)

# Memulai dengan Amplify Hosting

Untuk membantu Anda memahami cara kerja Amplify Hosting, tutorial ini memandu Anda melalui pembuatan dan penerapan aplikasi Next.js dari repositori Git.

Topik

- [Prasyarat](#)
- [Langkah 1: Hubungkan repositori Git](#)
- [Langkah 2: Konfirmasikan pengaturan build](#)
- [Langkah 3: Deploy aplikasi](#)
- [Langkah 4: \(Opsional\) membersihkan sumber daya](#)
- [Menambahkan fitur ke aplikasi Anda](#)

## 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.

Membuat aplikasi

Buat aplikasi Next.js dasar untuk digunakan untuk tutorial ini, menggunakan [create-next-app](#) instruksi 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.
  - a. Pilih Buat aplikasi baru di bagian atas halaman.
  - b. Di bagian bawah halaman, cari bagian Cara memulai, dan pilih Buat aplikasi baru.
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, lihat [Menyiapkan akses Amplify ke GitHub repositori](#).

### 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 Selanjutnya.

## 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 Selanjutnya.

## 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. Di jendela konfirmasi, masukkan **delete**. Kemudian pilih Hapus aplikasi.

## Menambahkan fitur ke aplikasi Anda

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 (PR) 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.

# Terapkan 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.

Untuk mempelajari cara Amplify mendukung SSR, baca topik-topik berikut.

## Topik

- [Definisi rendering sisi server](#)
- [Amplify dukungan untuk kerangka kerja SSR](#)
- [Menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build](#)
- [Pengoptimalan gambar untuk aplikasi SSR](#)
- [Dukungan versi Node.js untuk aplikasi Next.js](#)
- [Memecahkan masalah penerapan SSR](#)
- [Amplify dukungan untuk Next.js](#)

## Definisi rendering sisi server

Amplify mendukung penyebaran dan hosting aplikasi web statis yang dibuat dengan kerangka kerja aplikasi satu halaman (SPA) seperti React, dan aplikasi yang dibuat dengan generator situs statis (SSG) seperti Gatsby. Aplikasi web statis terdiri dari kombinasi file, seperti HTML, CSS, dan JavaScript file, yang disimpan di jaringan pengiriman konten (CDN). Ketika peramban klien membuat permintaan ke situs web, server menampilkan halaman ke klien dengan respons HTTP dan peramban klien menafsirkan konten dan menampilkannya kepada pengguna.

Amplify juga mendukung aplikasi web dengan rendering sisi server (SSR). Ketika klien mengirimkan permintaan ke halaman SSR, HTML untuk halaman tersebut dibuat di server pada setiap permintaan. SSR membantu developer menyesuaikan situs web per permintaan dan per pengguna. Selain itu, SSR dapat meningkatkan performa dan optimasi mesin pencari (SEO) untuk sebuah situs web.

# Amplify dukungan untuk kerangka kerja SSR

Amplify Hosting mendukung kerangka kerja SSR JavaScript berbasis apa pun dengan bundel penerapan yang sesuai dengan keluaran build yang diharapkan Amplify. Amplify Hosting menyediakan spesifikasi penerapan yang menstandarisasi file dan struktur direktori untuk output build aplikasi untuk kerangka SSR apa pun.

Penulis kerangka kerja dapat menggunakan spesifikasi penerapan berbasis sistem file untuk mengembangkan adaptor build sumber terbuka 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 atau adaptor kerangka kerja, Anda dapat mengembangkan solusi Anda sendiri untuk menghasilkan bundel penerapan yang sesuai dengan struktur direktori yang diharapkan Amplify Hosting.

Amplify Hosting mendukung primitif berikut: Aset statis, Compute, Optimasi gambar, dan aturan Routing. Anda dapat memanfaatkan primitif ini untuk menyebarkan aplikasi dengan fungsionalitas yang lebih kaya. Untuk informasi rinci tentang setiap primitif, lihat [Amplify dukungan primitif SSR](#).

Anda dapat memilih dari skenario berikut untuk memulai penerapan aplikasi SSR ke Amplify.

## Menerapkan aplikasi Next.js

Amplify mendukung aplikasi yang dibuat menggunakan Next.js tanpa perlu adaptor atau konfigurasi manual di konsol. Untuk informasi selengkapnya, lihat [Amplify dukungan untuk Next.js](#).

## Menerapkan aplikasi yang menggunakan adaptor kerangka kerja

Anda dapat mereferensikan adaptor kerangka kerja sumber terbuka yang tersedia untuk menyebarkan aplikasi SSR Anda ke Amplify Hosting. Untuk informasi selengkapnya, lihat [Menggunakan adaptor kerangka kerja](#).

Adaptor tersedia untuk kerangka kerja Nuxt. Untuk informasi selengkapnya tentang penggunaan adaptor ini, lihat dokumentasi [Nuxt](#).

## Bangun adaptor kerangka kerja

Pembuat kerangka kerja yang ingin mengintegrasikan fitur yang disediakan framework, dapat menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build

agar sesuai dengan struktur yang diharapkan Amplify. Untuk informasi selengkapnya, lihat [Menyebarkan server Express menggunakan manifes penerapan](#).

Konfigurasi skrip pembuatan pos

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](#).

## Menerapkan aplikasi SSR untuk Amplify

Anda dapat menggunakan petunjuk dalam topik 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 kerangka kerja](#).

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 Selanjutnya.
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 Selanjutnya.

8. Di halaman Tinjauan, pilih Simpan dan deploy.

## Menggunakan adaptor kerangka kerja

Anda dapat menginstal dan 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` di setel `.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](#)

## Menggunakan spesifikasi penerapan Amplify Hosting untuk mengonfigurasi keluaran build

Gunakan spesifikasi penerapan Amplify untuk mengonfigurasi keluaran build untuk framework SSR yang ingin Anda integrasikan dengan Amplify Hosting. Jika Anda seorang pembuat kerangka kerja, Anda dapat menggunakan spesifikasi penerapan untuk memahami cara menyusun output build yang diharapkan Amplify. Jika Anda tidak menggunakan kerangka kerja, Anda dapat mengembangkan solusi Anda sendiri untuk menghasilkan output build yang diharapkan Amplify.

### Spesifikasi Penerapan Amplify Hosting

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.

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.

## .amplify-hosting/staticDirektori

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 adalah `/folder1/folder2`, maka output build untuk aset statis yang dipanggil `main.css` akan menjadi `.amplify-hosting/static/folder1/folder2/main.css`.

## .amplify-hosting/computeDirektori

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

## File .amplify-hosting/deploy-manifest.json

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

versionAtribut 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

routesAtribut 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. [Menggunakan pengalihan](#)

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.
- `routesArray` 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
<code>path</code>	Tali	Ya	Mendefinisikan pola yang cocok dengan jalur permintaan masuk (tidak termasuk <code>querystring</code> ).

Kunci	Tipe	Diperlukan	Deskripsi
			<p>Panjang jalur maksimum adalah 255 karakter.</p> <p>Jalur harus dimulai dengan garis miring / ke depan.</p> <p>Sebuah jalur dapat berisi salah satu karakter berikut: [A-Z], [a-z], [0-9], [_.*\$/~"@: +].</p> <p>Untuk pencocokan pola, hanya karakter wildcard berikut yang didukung:</p> <ul style="list-style-type: none"><li>• *(cocok dengan 0 karakter atau lebih)</li><li>• /*Pola ini disebut pola catch-all dan akan cocok dengan semua permintaan yang masuk.</li></ul>

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


Kunci	Tipe	Diperlukan	Deskripsi
mundur	Target	Tidak	<p>Objek yang mendefinisikan target untuk mundur jika target asli mengembalikan kesalahan 404.</p> <p>targetJenis 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.</p>

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 <code>Static</code> , <code>Compute</code> , dan <code>ImageOptimization</code> .
src	String	Ya untuk Compute Tidak untuk primitif lainnya	String yang menentukan nama subdirektori dalam bundel deployment yang berisi kode executable primitif. Valid dan diperlukan hanya untuk primitif <code>Compute</code> .  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 <code>default</code> .
CacheControl	String	Tidak	String yang menentukan nilai header <code>Cache-Control</code> untuk diterapkan pada respons. Hanya berlaku untuk

Kunci	Tipe	Diperlukan	Deskripsi
			<p>Statis dan ImageOptimization primitif.</p> <p>Nilai yang ditentukan diganti oleh header khusus. Untuk informasi selengkapnya tentang header pelanggan Amplify Hosting, lihat. <a href="#">Header kustom</a></p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Header Cache-Control ini hanya diterapkan pada respons yang berhasil dengan kode status disetel ke 200 (OK).</p> </div>

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 Anda, lihat [Praktik terbaik untuk mengonfigurasi aturan perutean](#)

## Menggunakan atribut `ComputeResources`

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

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

`imageSettingsAtribut` 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[];
  mininumCacheTTL: 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 menggunakan optimasi gambar. Biarkan array kosong untuk mengizinkan hanya domain penerapan untuk menggunakan optimasi gambar.
RemotePatterns	RemotePattern[]	Ya	Array pola eksternal yang diizinkan yang dapat menggunakan optimasi gambar. Mirip dengan domain, tetapi memberikan kontrol lebih dengan ekspresi reguler (regex).
format	ImageFormat[]	Ya	Array format gambar keluaran yang diizinkan.
MinimumCachettl	Jumlah	Ya	Durasi cache dalam hitungan detik untuk gambar yang dioptimalkan.

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

Definisi objek berikut menunjukkan penggunaan untuk RemotePattern objek.

```
type RemotePattern = {
  protocol?: 'http' | '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.  Nilai-nilai yang valid adalah http atau 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. 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, lihat [Menggunakan 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 pengaturan Express server Anda.
- `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 runtime dan titik Express masuk server Anda.

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 ini.

```
"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.

## 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 mengimplementasikan 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 khusus, lihat [Menggunakan pemuat gambar khusus](#).

### 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](#).

### 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. [Spesifikasi Penerapan Amplify Hosting](#)

## 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.
- HTTP atau HTTPS adalah satu-satunya protokol yang didukung untuk sumber gambar dengan URL jarak jauh.

## 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, protokol http dan https didukung. Nilai harus dikodekan URL.	?url=http%3A%2F%2Fwww.example.com%2Fbuffalo.png
lebar	Jumlah	Ya	Lebar dalam piksel dari gambar yang dioptimalkan.	?width=800
tingginya	Jumlah	Tidak	Ketinggian piksel dari gambar yang dioptimalkan. Jika tidak ditentukan, gambar akan diskalakan secara otomatis agar sesuai dengan lebarnya.	?height=600
cocok	Nilai enum:cover,,cont	Tidak	Bagaimana gambar diubah	?width=800&height=

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
	inside outside		ukurannya agar sesuai dengan lebar dan tinggi yang ditentukan.	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
memangkas	Jumlah	Tidak	Memangkas piksel dari semua tepi yang berisi nilai yang mirip dengan warna latar belakang yang ditentukan dari piksel kiri atas.	?trim=50



Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
perluas	Objek	Tidak	Menambahk an piksel ke tepi gambar menggunak an 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
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

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
format	String	Tidak	Format output yang diinginkan untuk gambar yang dioptimalkan.	?format=webp
kualitas	Jumlah	Tidak	Kualitas gambar, dari 1 hingga 100. Hanya digunakan saat mengonversi format gambar.	?quality=50
merotasi	Jumlah	Tidak	Memutar gambar dengan sudut yang ditentukan dalam jumlah derajat.	?rotate=45
membalik	Boolean	Tidak	Mencerminkan gambar secara vertikal (atas-bawah) pada sumbu x. Ini selalu terjadi sebelum rotasi, jika ada.	?flip

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
gagal	Boolean	Tidak	Mencerminkan gambar secara horizontal (kiri-kanan) pada sumbu y. Ini selalu terjadi sebelum rotasi, jika ada.	?flop
mempertajam	Jumlah	Tidak	Penajaman meningkatkan definisi tepi pada gambar. Nilai yang valid adalah antara 0,000001 dan 10.	?sharpen=1
median	Jumlah	Tidak	Menerapkan filter median. Ini menghilangkan noise atau menghaluskan tepi gambar.	?sharpen=3
mengaburkan	Jumlah	Tidak	Menerapkan blur Gaussian dari sigma yang ditentukan. Nilai yang valid adalah 0,3 hingga 1.000.	?blur=20

Parameter kueri	Tipe	Diperlukan	Deskripsi	Contoh
gama	Jumlah	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	Jumlah	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.

## Pembuatan cache

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 pengoptimalan 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.

## Dukungan versi Node.js untuk aplikasi Next.js

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

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, lihat [Pembaruan paket langsung](#). 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.

## Memecahkan masalah penerapan SSR

Jika Anda mengalami masalah tak terduga saat menerapkan 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

- [Anda menggunakan adaptor kerangka kerja](#)
- [Rute Edge API menyebabkan build Next.js Anda gagal](#)
- [Regenerasi Statis Incremental On-Demand tidak berfungsi untuk aplikasi Anda](#)
- [Output build aplikasi Anda melebihi ukuran maksimum yang diizinkan](#)
- [Build Anda gagal dengan kesalahan kehabisan memori](#)
- [Ukuran respons HTTP terlalu besar](#)

## Anda menggunakan adaptor kerangka kerja

Jika Anda mengalami masalah saat menerapkan aplikasi SSR yang menggunakan adaptor kerangka kerja, lihat. [Amplify dukungan untuk kerangka kerja SSR](#)

## Rute Edge API menyebabkan build Next.js Anda gagal

Saat ini, Amplify tidak mendukung Next.js Edge API Routes. Anda harus menggunakan API dan middleware non-edge saat menghosting aplikasi dengan Amplify.

## Regenerasi Statis Incremental On-Demand tidak berfungsi untuk aplikasi Anda

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 Anda 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 `bagianjob.steps.stepName: "BUILD"`. URL disorot dengan warna merah dalam contoh output 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.zipFile` 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 output dengan informasi ukuran untuk `static` direktori `compute` dan.

```

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 Anda 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 menyetel variabel lingkungan menggunakan konsol Amplify, lihat [Tetapkan 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 terlalu besar

Saat ini, ukuran respons maksimum yang didukung Amplify untuk aplikasi Next.js 12 atau yang lebih baru menggunakan platform Web Compute adalah 5,72 MB. Tanggapan atas batas itu mengembalikan 504 kesalahan tanpa konten ke klien.

## 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 framework React untuk membangun aplikasi web fullstack. Anda dapat menerapkan aplikasi yang dibangun dengan Next.js 14 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, 13, dan 14. Jika Anda menerapkan aplikasi Next.js ke Amplify sebelum rilis komputasi Amplify Hosting, 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)
- 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)
- Deteksi lokal otomatis yang diinternasionalisasi (i18n)
- Next.js Streaming
- Menjalankan middleware pada aset statis dan gambar yang dioptimalkan

## 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.

## Harga untuk aplikasi Next.js

Saat menerapkan aplikasi SSR Next.js 12 atau yang lebih baru, komputasi Amplify Hosting mengelola sumber daya yang diperlukan untuk menerapkan aplikasi SSR untuk Anda. [Untuk informasi tentang biaya komputasi Amplify Hosting, lihat Harga.AWS Amplify](#)

## Menerapkan aplikasi Next.js dengan Amplify

Secara default, Amplify menerapkan aplikasi SSR baru menggunakan layanan komputasi Amplify Hosting dengan dukungan untuk Next.js 12, 13, dan 14. 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 Next.js 11 ke komputasi Amplify Hosting](#).

Gunakan petunjuk berikut untuk menerapkan aplikasi Next.js baru.

Untuk menerapkan aplikasi Next.js 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. Pilih nama repositori untuk terhubung.
  - b. Pilih nama cabang repositori untuk terhubung.
  - c. Pilih Selanjutnya.
5. Aplikasi ini 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 Selanjutnya.
  7. Di halaman Tinjauan, pilih Simpan dan deploy.

## Pengaturan file Package.json

Ketika Anda men-deploy aplikasi Next.js, Amplify memeriksa skrip build aplikasi di file `package.json` untuk mendeteksi apakah aplikasi tersebut SSR atau SSG.

Berikut contoh dari skrip build untuk aplikasi Next.js SSR. Skrip build `"next build"` menunjukkan bahwa aplikasi mendukung halaman SSG dan SSR.

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

Berikut contoh skrip build untuk aplikasi Next.js SSG. 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"  
},
```

## Pengaturan build Amplify

Setelah memeriksa file `package.json` aplikasi Anda untuk menentukan apakah Anda men-deploy aplikasi SSG atau SSR, Amplify akan memeriksa pengaturan build 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](#).

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/**/*
```

Jika Amplify mendeteksi bahwa Anda sedang menerapkan aplikasi SSG, Amplify akan menghasilkan `buildspec` 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.

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

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
```

```
- npm ci
build:
  commands:
    - npm run build
artifacts:
  baseDirectory: out
  files:
    - '**/*'
cache:
  paths:
    - node_modules/**/*
```

## Migrasi aplikasi 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 14.

Konsol Amplify mendeteksi aplikasi di akun Anda yang digunakan sebelum rilis layanan komputasi Amplify Hosting dengan dukungan penuh untuk Next.js versi 12, 13, dan 14. 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.

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, 13, atau 14.

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, 13, atau 14.

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, 13, atau 14 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.

## Perbarui 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 di-deploy 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 pengaturan 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](#).

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. 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 DB_HOST -e DB_USER -e DB_PASS >> .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 DB_HOST -e DB_USER -e DB_PASS >> .env.production
```

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

```
DB_HOST=localhost
DB_USER=myuser
```

```
DB_PASS=mypassword
```

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=abcdefghijkl  
NEXT_PUBLIC_GRAPHQL_ENDPOINT=uowelalsmlsadf  
NEXT_PUBLIC_SEARCH_KEY=asdfiojslf  
NEXT_PUBLIC_SEARCH_ENDPOINT=https://search-url
```

## 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 DB_HOST -e DB_USER -e DB_PASS >> 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 DB_HOST -e DB_USER -e DB_PASS >> 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. Perhatikan bahwa aplikasi pnpm dan Turborepo memerlukan konfigurasi tambahan. Untuk informasi selengkapnya, lihat [Pengaturan build monorepo](#).

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

## CloudWatch Log Amazon untuk aplikasi SSR

Amplify mengirimkan informasi tentang runtime Next.js ke Amazon CloudWatch Logs di file 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](#).

## Amplify dukungan 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 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 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. Tanpa izin minimum yang diperlukan, Anda akan mendapatkan 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](#) 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: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
cloudfront>DeleteDistribution
iam:AttachRolePolicy
iam:CreateRole
iam:CreateServiceLinkedRole
iam:GetRole
iam:PutRolePolicy
iam:PassRole
iam:UpdateAssumeRolePolicy
iam>DeleteRolePolicy
lambda:CreateFunction
lambda:EnableReplication
lambda>DeleteFunction
lambda:GetFunction
lambda:GetFunctionConfiguration
```

```
lambda:PublishVersion
lambda:UpdateFunctionCode
lambda:UpdateFunctionConfiguration
lambda:ListTags
lambda:TagResource
lambda:UntagResource
lambda:ListEventSourceMappings
lambda:CreateEventSourceMapping
route53:ChangeResourceRecordSets
route53:ListHostedZonesByName
route53:ListResourceRecordSets
s3:CreateBucket
s3:GetAccelerateConfiguration
s3:GetObject
s3:ListBucket
s3:PutAccelerateConfiguration
s3:PutBucketPolicy
s3:PutObject
s3:PutBucketTagging
s3:GetBucketTagging
sqs:CreateQueue
sqs>DeleteQueue
sqs:GetQueueAttributes
sqs:SetQueueAttributes
amplify:GetApp
amplify:GetBranch
amplify:UpdateApp
amplify:UpdateBranch
```

## Memecahkan masalah penerapan Next.js 11

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

### Topik

- [Direktori output ditimpa](#)
- [Anda menerima pesan kesalahan 404 setelah men-deploy situs SSR](#)
- [Aplikasi Anda tidak memiliki aturan penulisan ulang untuk distribusi CloudFront SSR](#)
- [Ukuran aplikasi yang akan di-deploy terlalu besar](#)
- [Build Anda gagal dengan kesalahan kehabisan memori](#)
- [Aplikasi memiliki cabang SSR dan SSG](#)

- [Aplikasi menyimpan file statis di folder dengan path cadangan](#)
- [Aplikasi Anda telah mencapai CloudFront batas](#)
- [Variabel lingkungan tidak dibawa ke fungsi Lambda](#)
- [Fungsi Lambda @Edge dibuat di Wilayah AS Timur \(Virginia N.\)](#)
- [Aplikasi Next.js Anda menggunakan fitur yang tidak didukung](#)
- [Gambar di aplikasi Next.js Anda tidak dimuat](#)
- [Wilayah yang Tidak Didukung](#)

## Direktori output ditimpa

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](#).

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/**/*
```

## Anda menerima pesan kesalahan 404 setelah men-deploy situs SSR

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 output ditimpa](#).

## Aplikasi Anda 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.

### Rewrites and redirects

Rewrites are a way for a web server to reroute navigation from one URL to another. Support for the following HTTP status codes: 200, 301, 302, 404. [Learn more](#)

Source address	Target address	Type	Country code
/<*>	https:// .cloudfront.net/<*>	200 (Rewrite)	-
/<*>	/index.html	404 (Rewrite)	-

## Ukuran aplikasi yang akan di-deploy terlalu besar

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 Anda 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 menyetel variabel lingkungan menggunakan konsol Amplify, lihat [Tetapkan 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.

### Aplikasi 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 menyimpan file statis di folder dengan path cadangan

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 Anda 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.

## Variabel lingkungan tidak dibawa ke fungsi Lambda

Variabel lingkungan yang Anda tentukan di konsol Amplify untuk aplikasi SSR tidak dibawa ke fungsi aplikasi. AWS Lambda Lihat, [Membuat variabel lingkungan dapat diakses oleh runtime sisi server](#), untuk petunjuk rinci tentang cara menambahkan variabel lingkungan yang dapat Anda referensikan dari fungsi Lambda Anda.

## 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 Anda 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 Next.js versi lama, Anda dapat memigrasikan aplikasi ke penyedia SSR komputasi Amplify Hosting. Untuk petunjuk, lihat [Migrasi aplikasi Next.js 11 ke komputasi Amplify Hosting](#).

## Gambar di aplikasi Next.js Anda 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.

# Menyiapkan domain khusus

Anda dapat menghubungkan aplikasi yang telah digunakan dengan Amplify Hosting ke domain khusus. 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 khusus, pengguna akan melihat bahwa aplikasi Anda di-host di URL khusus, misalnya `https://www.example.com`.

Anda dapat membeli domain khusus 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 untuk domain kapan saja. Untuk informasi rinci tentang mengelola sertifikat, lihat [Menggunakan 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 tentang menyelesaikan langkah ini, lihat [Memulai dengan Amplify Hosting](#).

- Anda memiliki pengetahuan dasar tentang domain dan terminologi DNS.

Untuk informasi lebih lanjut seputar domain dan DNS, lihat [Memahami terminologi dan konsep DNS](#).

## Topik

- [Memahami terminologi dan konsep DNS](#)
- [Menggunakan sertifikat SSL/TLS](#)
- [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](#)
- [Memperbarui data DNS untuk domain yang dikelola oleh Google Domains](#)
- [Memperbarui sertifikat SSL/TLS untuk domain](#)
- [Mengelola subdomain](#)
- [Subdomain wildcard](#)
- [Mengatur 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 menyiapkan 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 memiliki domain kustom Anda. Jika Anda meng-host domain dengan Route 53, verifikasi dilakukan secara otomatis atas nama Anda. Namun, jika Anda meng-host domain Anda dengan penyedia pihak ketiga seperti GoDaddy, Anda harus memperbarui pengaturan DNS domain Anda secara manual dan menambahkan catatan CNAME baru yang disediakan oleh Amplify.

## Amplify Hosting proses aktivasi domain kustom

Saat Anda menambahkan domain khusus dengan Amplify Hosting, ada beberapa langkah yang harus diselesaikan sebelum Anda dapat melihat aplikasi menggunakan domain kustom Anda. Daftar berikut menjelaskan setiap langkah dalam proses pengaturan domain.

### Pembuatan SSL/TLS

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

### Konfigurasi dan verifikasi SSL/TLS

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

## Menggunakan sertifikat SSL/TLS

Sertifikat SSL/TLS adalah dokumen digital yang memungkinkan browser web mengidentifikasi dan membuat koneksi jaringan terenkripsi ke situs web menggunakan protokol SSL/TLS yang aman. 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 terkelola, Amplify mengeluarkan sertifikat SSL/TLS untuk semua domain yang terhubung ke aplikasi Anda sehingga semua lalu lintas diamankan melalui HTTPS/2. Sertifikat default yang dihasilkan oleh AWS Certificate Manager (ACM) berlaku selama 13 bulan dan diperpanjang secara otomatis selama aplikasi Anda di-host 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 khusus, Anda harus mendapatkan sertifikat dari otoritas sertifikat pihak ketiga pilihan Anda. Selanjutnya, impor sertifikat 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 AS Timur (Virginia Utara) (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 khusus Anda dan mengimpornya lagi.

Anda dapat mengubah sertifikat yang digunakan untuk domain kapan saja. 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 khusus yang digunakan ke sertifikat khusus yang berbeda. Untuk petunjuk tentang memperbarui sertifikat, lihat [Memperbarui sertifikat SSL/TLS](#) untuk domain.

## Menambahkan domain kustom yang dikelola Amazon Route 53

Untuk menambahkan domain kustom yang dikelola oleh Route 53


1. Masuk ke AWS Management Console dan buka 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 adalah `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.com` dan `https://example.com` dengan pengalihan diatur dari domain root ke subdomain `www`.

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

8. Pilih sertifikat SSL/TLS yang akan 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 khusus yang dikelola oleh penyedia DNS pihak ketiga ke aplikasi yang digunakan dengan Amplify.

Jika Anda menggunakan GoDaddy atau Google Domains, lihat [the section called “Memperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy”](#) atau [the section called “Memperbarui data DNS untuk domain yang dikelola oleh Google Domains”](#) untuk prosedur khusus untuk penyedia ini.

Cara menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga

1. Masuk ke AWS Management Console dan buka 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 adalah `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 Konfigurasi domain.
    - c. Server nama zona yang dihosting ditampilkan di konsol. Buka situs web penyedia DNS Anda 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 Konfigurasi 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.com` dan `https://example.com` dengan pengalihan diatur dari domain root ke subdomain `www`.

(Opsional) Anda dapat mengubah konfigurasi default jika ingin menambahkan subdomain saja. Untuk mengubah konfigurasi default, pilih Menulis ulang dan mengalihkan dari panel navigasi dan konfigurasi domain Anda.
8. Pilih sertifikat SSL/TLS yang akan 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 Anda memilih Konfigurasi manual, pada langkah enam, Anda harus memperbarui catatan DNS Anda dengan penyedia domain pihak ketiga Anda.

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

### DNS Records

Verify records in your domain registrar match these records.

#### Verification record

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtsbpdnt.acm-validations.aws.</code>

#### Subdomain records

Hostname	Type	Data/URL
<code>@</code>	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
<code>www</code>	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

11. Lakukan salah satu hal berikut ini:
  - Jika Anda menggunakan GoDaddy, pergi ke [Memperbarui catatan DNS untuk domain yang dikelola oleh GoDaddy](#).
  - Jika Anda menggunakan Google Domains, buka [Memperbarui data DNS untuk domain yang dikelola oleh Google Domains](#).
  - 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 data DNS untuk memverifikasi kepemilikan subdomain Anda seperti `_c3e2d7eaf1e656b73f46cd6980fdc0e.example.com`, masukkan hanya untuk nama subdomain rekaman CNAME. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

Tangkapan layar berikut menunjukkan lokasi catatan verifikasi yang akan digunakan.

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtsbpdnt.acm-validations.aws.</code>

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

Jika konsol Amplify menampilkan catatan server validasi ACM seperti `_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`, masukkan untuk nilai catatan CNAME. **`_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`**

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



## DNS Records ×

Verify records in your domain registrar match these records.

### Verification record

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtspbndt.acm-validations.aws.</code>

### Subdomain records

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

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

#### ⓘ Note

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

#### ⚠ Important

Penting bagi Anda untuk melakukan langkah ini segera setelah menambahkan domain khusus Anda di konsol Amplify. ACM 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.

- 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 Anda memiliki lalu lintas produksi, kami sarankan Anda memperbarui catatan CNAME ini setelah status domain Anda ditampilkan sebagai TERSEDIA di konsol Amplify.

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

### DNS Records ×

Verify records in your domain registrar match these records.

#### Verification record

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtsbpdnt.acm-validations.aws.</code>

#### Subdomain records

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgbp.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgbp.cloudfront.net</code>

- Konfigurasi catatan ANAME/ALIAS untuk menunjuk 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

Untuk menambahkan domain kustom yang dikelola oleh GoDaddy

1. Sebelum Anda dapat memperbarui catatan DNS Anda GoDaddy, selesaikan langkah satu hingga sembilan prosedur [the section called “Menambahkan domain kustom yang dikelola penyedia DNS pihak ketiga”](#).
2. Masuk ke GoDaddy akun Anda.
3. Dalam daftar domain Anda, temukan domain yang akan ditambahkan dan 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 hanya subdomain. Misalnya, jika subdomain Anda adalah `www.example.com`, masukkan `www` untuk Nama.
  - d. Untuk Nilai, lihat catatan DNS Anda di konsol Amplify lalu masukkan nilainya. 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.

## DNS Records ×

Verify records in your domain registrar match these records.

### Verification record

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtspbnt.acm-validations.aws.</code>

### Subdomain records

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgbp.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgbp.cloudfront.net</code>

- e. Pilih Simpan.
6. Buat catatan CNAME kedua untuk menunjuk 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 memverifikasi kepemilikan subdomain Anda adalah `_c3e2d7eaf1e656b73f46cd6980fdc0e.example.com`, masukkan hanya untuk Nama. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

Tangkapan layar berikut menunjukkan lokasi catatan verifikasi yang akan digunakan.

**DNS Records**

Verify records in your domain registrar match these records.

**Verification record**

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtspbndt.acm-validations.aws.</code>

**Subdomain records**

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

- c. Untuk Nilai, masukkan sertifikat validasi ACM.

Misalnya, jika server validasi adalah

`_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`, masukkan `_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws` untuk Nilai.

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

**DNS Records**

Verify records in your domain registrar match these records.

**Verification record**

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtspbndt.acm-validations.aws.</code>

**Subdomain records**

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

- d. Pilih Simpan.

**Note**

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

- Langkah ini tidak diperlukan untuk subdomain. GoDaddy tidak mendukung catatan ANAME/ALIAS. Untuk penyedia DNS yang tidak memiliki dukungan ANAME/ALIAS, kami sangat menyarankan untuk memigrasikan 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 siapkan penerusan domain:

- Pada halaman DNS, cari menu di bagian atas halaman dan pilih Penerusan.
- Di bagian Domain, pilih Tambahkan Penerusan.
- Pilih http://, lalu masukkan nama subdomain Anda untuk diteruskan ke (misalnya, www.example.com) untuk URL Tujuan.
- Untuk Tipe Penerusan, pilih Sementara (302).
- Pilih, Simpan.

## Memperbarui data DNS untuk domain yang dikelola oleh Google Domains

Cara menambahkan domain kustom yang dikelola Google Domains

- Sebelum Anda dapat memperbarui data DNS Anda dengan Google Domains, selesaikan langkah satu hingga sembilan prosedur [Untuk menambahkan domain khusus yang dikelola oleh penyedia DNS pihak ketiga](#).
- Masuk ke akun Anda di <https://domains.google.com> dan pilih Domain saya di panel navigasi kiri.
- Dalam daftar domain Anda, temukan domain yang akan ditambahkan dan pilih Kelola.

4. Di panel navigasi kiri, pilih DNS. Google menampilkan catatan Sumber Daya untuk domain Anda. Anda harus menambahkan dua catatan CNAME baru.
5. Buat catatan CNAME pertama untuk mengarahkan semua subdomain Anda ke domain Amplify sebagai berikut:
  - a. Untuk nama Host, masukkan hanya nama subdomain. Misalnya, jika subdomain Anda adalah `www.example.com`, masukkan `www` untuk nama Host.
  - b. Untuk Type, pilih CNAME.
  - c. Untuk Data, masukkan nilai yang tersedia di konsol Amplify.

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

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

### DNS Records ×

Verify records in your domain registrar match these records.

#### Verification record

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtspbnt.acm-validations.aws.</code>

#### Subdomain records

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

6. Buat catatan CNAME kedua untuk menunjuk ke server validasi AWS Certificate Manager (ACM). Satu ACM tervalidasi menghasilkan satu sertifikat SSL/TLS untuk domain Anda.
  - a. Untuk nama Host, masukkan subdomain.

Misalnya, jika catatan DNS di konsol Amplify untuk memverifikasi kepemilikan subdomain Anda adalah `_c3e2d7eaf1e656b73f46cd6980fdc0e.example.com`, masukkan hanya `_c3e2d7eaf1e656b73f46cd6980fdc0e` untuk nama Host.

Tangkapan layar berikut menunjukkan lokasi catatan verifikasi yang akan digunakan.

### DNS Records

Verify records in your domain registrar match these records.

**Verification record**

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtsbpdnt.acm-validations.aws.</code>

**Subdomain records**

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

- b. Untuk Type, pilih CNAME.
- c. Untuk Data, masukkan sertifikat validasi ACM.

Misalnya, jika server validasi adalah

`_cf1z2npwt9vzexample93c1j4xzc92wl.2te3iy6kzr.acm-validations.aws.`, masukkan `_cf1z2npwt9vzexample93c1j4xzc92wl.2te3iy6kzr.acm-validations.aws.` untuk Data.

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

### DNS Records

Verify records in your domain registrar match these records.

**Verification record**

Hostname	Type	Data/URL
<code>_39e1e8d7e0aedc8165cf52a176612124.testexample.com.</code>	CNAME	<code>_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtsbpdnt.acm-validations.aws.</code>

**Subdomain records**

Hostname	Type	Data/URL
@	ANAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>
www	CNAME	<code>d1zp5qtgx0mgpb.cloudfront.net</code>

## 7. Pilih Simpan.



**Note**

Sertifikat Amplify; default yang dihasilkan oleh AWS Certificate Manager (ACM) berlaku selama 13 bulan dan diperpanjang secara otomatis selama aplikasi Anda di-host 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.

8. Dukungan Google Domains untuk catatan ANAME/ALIAS sedang dalam pratinjau. Untuk penyedia DNS yang tidak memiliki dukungan ANAME/ALIAS, kami sangat menyarankan untuk memigrasikan DNS Anda ke Amazon Route 53. Untuk informasi selengkapnya, lihat [Mengonfigurasi Amazon Route 53 sebagai layanan DNS Anda](#). Jika Anda ingin menetapkan Google Domains sebagai penyedia dan memperbarui domain root, atur penerusan domain. Temukan halaman Situs Web untuk domain Google Anda. Kemudian pilih Forward domain dan konfigurasi penerusan Anda di halaman penerusan Web.

**Note**

Penerapan pembaruan pada pengaturan DNS Anda untuk domain Google dapat memakan waktu hingga 48 jam. Untuk bantuan seputar mengatasi kesalahan yang terjadi, lihat [Memecahkan masalah domain kustom](#).

## 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. Anda juga dapat mengubah sertifikat kustom yang digunakan untuk domain. 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 AWS Management Console dan buka konsol [Amplify](#).
2. Pilih aplikasi yang ingin 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 AWS Management Console dan buka 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 Anda dan kemudian pilih Konfigurasi domain. Misalnya, jika nama domain Anda adalah `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 AWS Management Console dan buka 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 Exclude root, dan ubah subdomain untuk menambahkan level baru.

Misalnya, jika Anda memiliki domain bernama alpha.example.com dan Anda ingin membuat subdoman bertingkat beta.alpha.example.com, Anda akan 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 AWS Management Console dan buka 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 Tambah baru.
7. Pilih Simpan.

## Subdomain wildcard

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. Saat 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 wildcard:

- 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.

## Untuk menambah atau menghapus subdomain wildcard

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

1. Masuk ke AWS Management Console dan buka konsol [Amplify Hosting](#).
2. Pilih aplikasi yang ingin Anda kelola subdomain wildcard.
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.
  - Untuk menambahkan subdomain wildcard baru
    - 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.

## Mengatur 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 AWS Management Console dan buka 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 khusus ke aplikasi di AWS Amplify konsol, lihat topik berikut di bagian ini untuk bantuan pemecahan masalah.

Jika Anda tidak melihat solusi untuk masalah Anda di sini, hubungi AWS Support. Untuk informasi selengkapnya, lihat [Membuat kasus dukungan](#) di Panduan AWS Support Pengguna.

### Topik




- [Bagaimana cara memverifikasi bahwa CNAME saya berhasil ditetapkan?](#)
- [Domain saya yang di-host dengan pihak ketiga terus menampilkan status Menunggu Verifikasi](#)
- [Domain saya yang di-host dengan Amazon Route 53 terus menampilkan status Menunggu Verifikasi](#)
- [Saya mendapatkan kesalahan CNAME AlreadyExistsException](#)
- [Saya mendapatkan kesalahan Verifikasi Tambahan yang Diperlukan](#)
- [Saya mendapatkan kesalahan 404 pada URL CloudFront](#)
- [Saya mendapatkan sertifikat SSL atau kesalahan HTTPS saat mengunjungi domain saya](#)

## Bagaimana cara 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](https://www.whatsmydns.net/) untuk memeriksa catatan CNAME Anda untuk domain `www.example.com`.



2. Pilih Cari, lalu [whatsmydns.net](https://www.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`.

 Dallas TX, United States Speakeasy	<code>d1e0xkpcedddpz.cloudfront.net</code> ✓
 Reston VA, United States Sprint	<code>d1e0xkpcedddpz.cloudfront.net</code> ✓
 Atlanta GA, United States Speakeasy	<code>d1e0xkpcedddpz.cloudfront.net</code> ✓

## Domain saya yang di-host dengan pihak ketiga terus menampilkan status Menunggu Verifikasi

1. Jika domain kustom Anda macet dalam status Verifikasi Tertunda, verifikasi bahwa CNAME catatan Anda telah diselesaikan. Lihat topik pemecahan masalah sebelumnya, [Bagaimana cara memverifikasi bahwa CNAME resolusi saya](#), untuk petunjuk tentang melakukan tugas ini.
2. Jika CNAME catatan Anda tidak terselesaikan, konfirmasikan bahwa CNAME entri ada di pengaturan DNS Anda dengan penyedia domain Anda.

### Important

Penting untuk memperbarui CNAME catatan Anda segera setelah Anda membuat domain khusus Anda. Setelah aplikasi Anda dibuat di konsol Amplify, CNAME rekaman

Anda akan diperiksa setiap beberapa menit untuk menentukan apakah itu diselesaikan. Jika catatan tidak berhasil ditetapkan setelah satu jam, pemeriksaan dilakukan setiap beberapa jam, dan dapat mengakibatkan penyiapan domain memerlukan waktu lebih lama. Jika Anda menambahkan atau memperbarui CNAME catatan beberapa jam setelah membuat aplikasi, ini adalah kemungkinan besar aplikasi Anda macet dalam status Verifikasi Tertunda.

3. Jika Anda telah memverifikasi bahwa CNAME catatan itu ada, maka mungkin ada masalah dengan penyedia DNS Anda. Anda dapat menghubungi penyedia DNS untuk mendiagnosis mengapa verifikasi CNAME DNS tidak diselesaikan atau Anda dapat memigrasikan DNS Anda 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 ke Amazon Route 53, kemungkinan domain Anda memiliki server nama yang berbeda dari yang dikeluarkan oleh 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.

The screenshot shows the Amazon Route 53 console interface. On the left, there is a table of hosted zones with columns for Domain Name, Type, Record Set Count, and Comment. The selected zone is highlighted in blue. On the right, the 'Hosted Zone Details' panel is visible, showing the Domain Name, Type (Public Hosted Zone), Hosted Zone ID (Z1NMQLEEGTLCM3), Record Set Count (2), and Comment. The 'Name Servers' field is highlighted with a red box and contains the following list of server names:

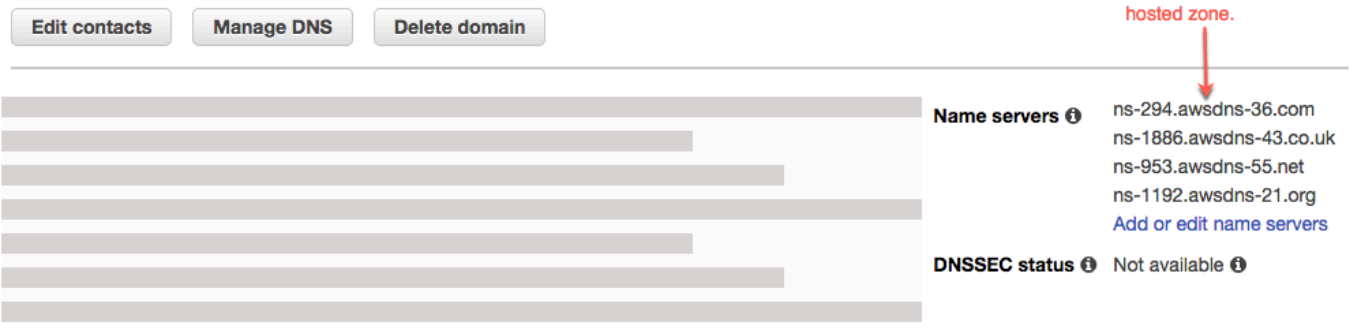
- ns-2003.awsdns-58.co.uk
- ns-70.awsdns-08.com
- ns-1173.awsdns-18.org
- ns-805.awsdns-36.net

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.

## Registered domains > designaws.com



5. Jika ini tidak menyelesaikan masalah, hubungi AWS Support. Untuk informasi selengkapnya, lihat [Membuat kasus dukungan](#) di Panduan AWS Support Pengguna.

## Saya mendapatkan kesalahan CNAME AlreadyExistsException

Jika Anda mendapatkan AlreadyExistsException kesalahan CNAME, ini berarti bahwa salah satu nama host yang Anda coba sambungkan (subdomain, atau domain apex) sudah digunakan ke distribusi Amazon lainnya. CloudFront Lakukan langkah-langkah berikut untuk mendiagnosis penyebab kesalahan.

1. Masuk ke [CloudFrontkonsol Amazon](#) dan verifikasi bahwa domain ini tidak diterapkan ke distribusi lain. Satu CNAME catatan dapat dilampirkan ke satu CloudFront distribusi pada satu waktu.
2. Jika sebelumnya Anda menerapkan 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. 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 menggunakan `www.example.com` untuk aplikasi lain, Anda tidak dapat menggunakan

`www.example.com` dengan aplikasi yang saat ini Anda hubungkan. Anda dapat menggunakan subdomain lain, seperti `blog.example.com`.

4. 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 AWS Support. Untuk informasi selengkapnya, lihat [Membuat kasus dukungan](#) di Panduan AWS Support Pengguna.

## 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 AWS Support. 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 khusus, konsol Amplify menampilkan CloudFront URL untuk aplikasi tersebut. 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. Konfigurasi 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 AWS Management Console dan buka konsol [Amplify](#).
- b. Pilih aplikasi yang ingin Anda tambahkan catatan CAA.
- 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.](#)
  - [Memperbarui data DNS untuk domain yang dikelola oleh Google Domains.](#)
- Hubungi AWS Support agar sertifikat SSL Anda diterbitkan kembali.

## Mengonfigurasi pengaturan build

Saat Anda menerapkan aplikasi dengan Amplify Hosting, aplikasi ini secara otomatis mendeteksi kerangka kerja front end dan pengaturan build terkait dengan memeriksa file `package.json` di repositori Anda. Anda memiliki opsi berikut untuk menyimpan pengaturan build aplikasi:

- Simpan pengaturan build di konsol Amplify - Konsol Amplify mendeteksi otomatis pengaturan build dan menyimpannya sehingga dapat diakses melalui 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.

Anda dapat mengedit setelan build aplikasi di Amplify console dengan memilih Hosting, lalu Build settings di panel navigasi. Pengaturan build diterapkan ke semua cabang di aplikasi Anda, kecuali untuk cabang dengan file `amplify.yml` yang disimpan di repositori.

### Note

Pengaturan build terlihat di menu Hosting konsol Amplify hanya jika aplikasi disiapkan untuk penerapan berkelanjutan dan terhubung ke repositori git. Untuk petunjuk tentang jenis penerapan ini, lihat [Memulai](#).

## Buat perintah dan pengaturan spesifikasi

Spesifikasi build YAMAL berisi kumpulan perintah build dan setelan terkait yang Amplify gunakan untuk menjalankan build Anda. Daftar berikut menjelaskan pengaturan ini dan bagaimana mereka digunakan.

versi

Nomor versi Amplify YAMAL.

appRoot

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

## env

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

## backend

Jalankan perintah Amplify CLI untuk menyediakan backend, memperbarui fungsi Lambda, atau skema GraphQL sebagai bagian dari penerapan berkelanjutan.

## frontend

Jalankan perintah build frontend.

## pengujian

Jalankan perintah selama fase pengujian. Pelajari cara [menambahkan tes ke aplikasi](#).

## membangun fase

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 pasca-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, misalnya. apps/app

## artefak>direktori dasar

Direktori tempat artefak build Anda ada.

## artefak>file

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

## cache

Bidang cache buildspec digunakan untuk menyimpan dependensi waktu pembuatan cache seperti folder `node_modules`, dan secara otomatis disarankan berdasarkan manajer paket dan kerangka kerja tempat aplikasi pelanggan dibangun. Selama build pertama, setiap path di sini disimpan di cache, dan pada build berikutnya kita kembali mengisi cache dan menggunakan dependensi yang disimpan di cache, jika mungkin, untuk mempercepat waktu build.

Contoh spesifikasi build berikut menunjukkan sintaks YAML dasar:

## Sintaks YAML spesifikasi build

```
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
    - path
test:
  phases:
    preTest:
      commands:
        - *enter command*
    test:
      commands:
        - *enter command*
    postTest:
      commands:
        - *enter command*
artifacts:
  files:
    - location
    - location
  configFilePath: *location*
  baseDirectory: *location*
```

## Pengaturan build khusus cabang

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
```



## Menavigasi ke subfolder

Untuk monorepos, pengguna ingin dapat `cd` masuk ke folder untuk menjalankan build. Setelah Anda menjalankan `cd` perintah, itu berlaku untuk semua tahapan build Anda sehingga Anda tidak perlu mengulangi perintah dalam fase terpisah.

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

## Menerapkan backend dengan ujung depan 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).

`amplifyPushPerintah` adalah skrip pembantu yang membantu Anda dengan penerapan 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 pembuatan.

```
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 AL2023 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>
```

## 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.

## Menonaktifkan build otomatis

Anda dapat mengonfigurasi Amplify untuk menonaktifkan build otomatis pada setiap komit kode. Untuk mengatur, pilih Pengaturan aplikasi, Pengaturan cabang, lalu cari bagian Cabang yang mencantumkan cabang yang terhubung. Pilih cabang, lalu pilih Tindakan, Nonaktifkan pembuatan otomatis. Komit baru ke cabang itu tidak akan lagi memulai build baru.

## Mengaktifkan atau menonaktifkan build dan deploy frontend berbasis diff

Anda dapat mengonfigurasi Amplify untuk menggunakan build frontend berbasis diff. Jika diaktifkan, pada awal setiap build Amplify mencoba menjalankan diff di folder `AndaappRoot`, atau `/src/` folder secara default. 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 AWS Management Console dan buka 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`.

## Mengaktifkan atau menonaktifkan 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 Hosting untuk menggunakan build backend berbasis diff menggunakan variabel lingkungan. `AMPLIFY_DIFF_BACKEND` Saat Anda mengaktifkan build backend berbasis diff, pada awal setiap build Amplify mencoba menjalankan diff pada folder di 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 setelah build di `amplify.yml` file aplikasi.

Untuk mengonfigurasi build backend berbasis diff

1. Masuk ke AWS Management Console dan buka 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.

## Pengaturan build monorepo

Monorepo adalah tempat untuk menyimpan beberapa proyek atau layanan mikro dalam satu repositori. Anda dapat menggunakan Amplify Hosting untuk menyebarkan aplikasi di 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. Saat menerapkan 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 Anda dapat 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. Saat `amplify.yml` file ada, pengaturannya akan mengganti setelan build apa pun yang disimpan di konsol Amplify.

## Sintaks YAML 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 deklarasikan dalam spesifikasi build monorepo Anda:

### `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.

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
      configFile: *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
      configFile: *location*
      baseDirectory: *location*
```

## 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 menyetel variabel `AMPLIFY_MONOREPO_APP_ROOT` lingkungan secara manual di bagian Variabel Lingkungan di 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 menerapkan aplikasi monorepo dengan konsol Amplify

1. Masuk ke AWS Management Console dan buka 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 Selanjutnya.
5. Di halaman Pengaturan aplikasi, Anda dapat menggunakan setelan default atau menyesuaikan setelan build untuk aplikasi Anda. Di bagian variabel Lingkungan, Amplify menetapkan `AMPLIFY_MONOREPO_APP_ROOT` ke jalur yang Anda tentukan di langkah 4d.
6. Pilih Selanjutnya.

7. Di halaman Tinjauan, pilih Simpan dan deploy.

## Mengatur variabel lingkungan AMPLIFY\_MONOREPO\_APP\_ROOT untuk aplikasi yang ada

Gunakan petunjuk berikut untuk menyetel variabel AMPLIFY\_MONOREPO\_APP\_ROOT lingkungan secara manual untuk aplikasi yang sudah di-deploy ke Amplify, atau telah dibuat menggunakan CloudFormation

Untuk mengatur variabel lingkungan AMPLIFY\_MONOREPO\_APP\_ROOT untuk aplikasi yang ada

1. Masuk ke AWS Management Console dan buka 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 kehoisted. 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
```

# 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 <https://main.appid.amplifyapp.com> dan <https://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 pengujian 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.

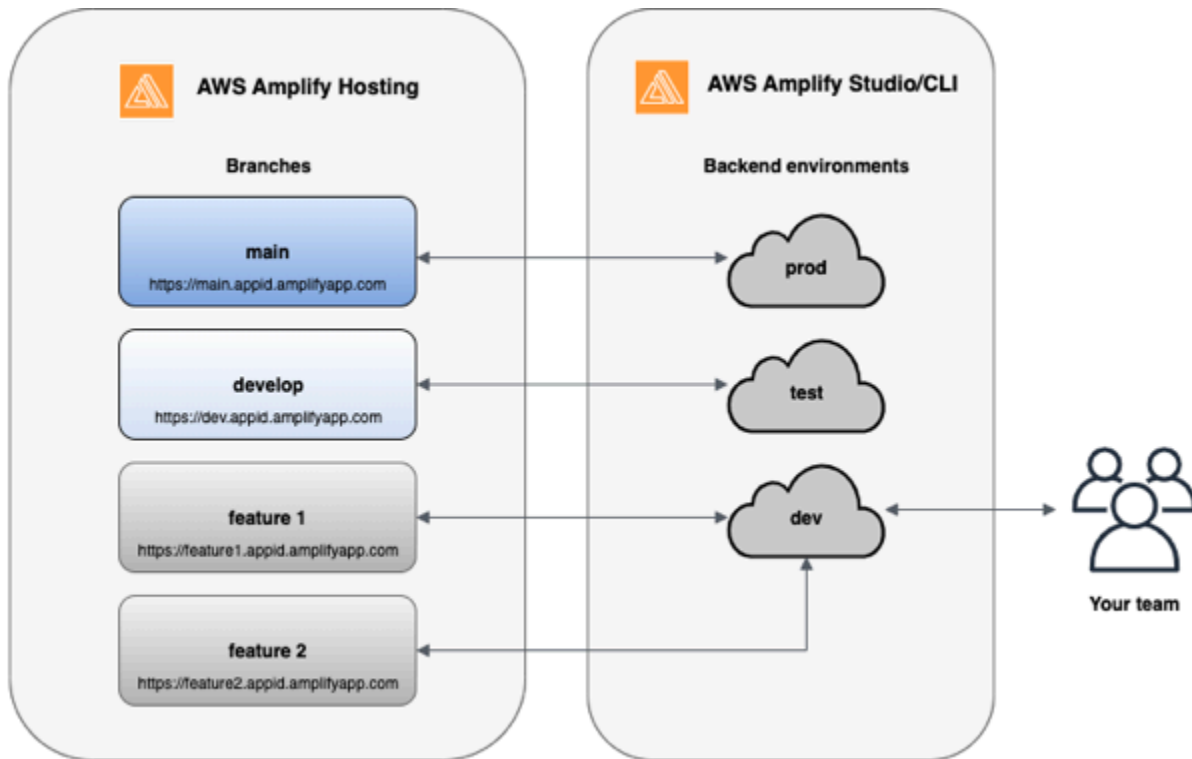
## Alur kerja tim dengan aplikasi Amplify Gen 1 fullstack

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

Amplify Hosting terus menerapkan sumber daya backend seperti GraphQL API dan fungsi 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.




## quick-notes

The app homepage lists all deployed frontend and backend environments.

Frontend environments | **Backend environments**

Each backend environment is a container for all of the cloud capabilities added to your app. An Amplify backend environment contains the list of categories enabled such as API, auth, and storage.

### prod



Categories added


- Authentication
- API

Deployment status

✔ Deployment completed 11/14/2019, 11:29:07 AM

▶ Edit backend

### test



Categories added


- Authentication
- API

Deployment status

✔ Deployment completed 11/14/2019, 11:29:07 AM

▶ Edit backend

### dev



Categories added

- Authentication
- API


Deployment status


✔ Deployment completed 11/14/2019, 11:29:07 AM


▶ Edit backend


6. Beralih ke tab Lingkungan frontend, lalu hubungkan penyedia repositori dan cabang main.


**From your existing code**  
Connect your source code from a Git repository or upload files to host a web app in minutes.

GitHub 

BitBucket 

GitLab 

AWS CodeCommit 

Deploy without Git provider 

[Continue](#)

7. Di layar pengaturan build, pilih lingkungan backend yang ada untuk mengatur deployment kontinu dengan cabang main. Pilih prod dari dropdown dan berikan peran layanan ke Amplify. Pilih Simpan dan deploy. Setelah build selesai, Anda dapat mengakses deployment cabang main di <https://main.appid.amplifyapp.com>.

## Configure build settings

### App build settings

**App name**  
Pick a name for your app.

Name cannot contain periods

**Existing Amplify backend detected**  
Connect your backend to continuously deploy changes to both your frontend and backend

Would you like Amplify Console to deploy changes to these resources with your frontend?

Yes - choose an existing environment or create a new one

Create new environment

Select dev

test

prod

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

### Add repository branch

**AWS CodeCommit**

Repository service provider

AWS CodeCommit

Branch

Select a branch from your repository.

develop

Backend environment

Select a backend environment for this branch.

test

Cancel Next

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 dapat diakses di <https://newinternet.appid.amplifyapp.com> dan digunakan bersama rekan tim. 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.

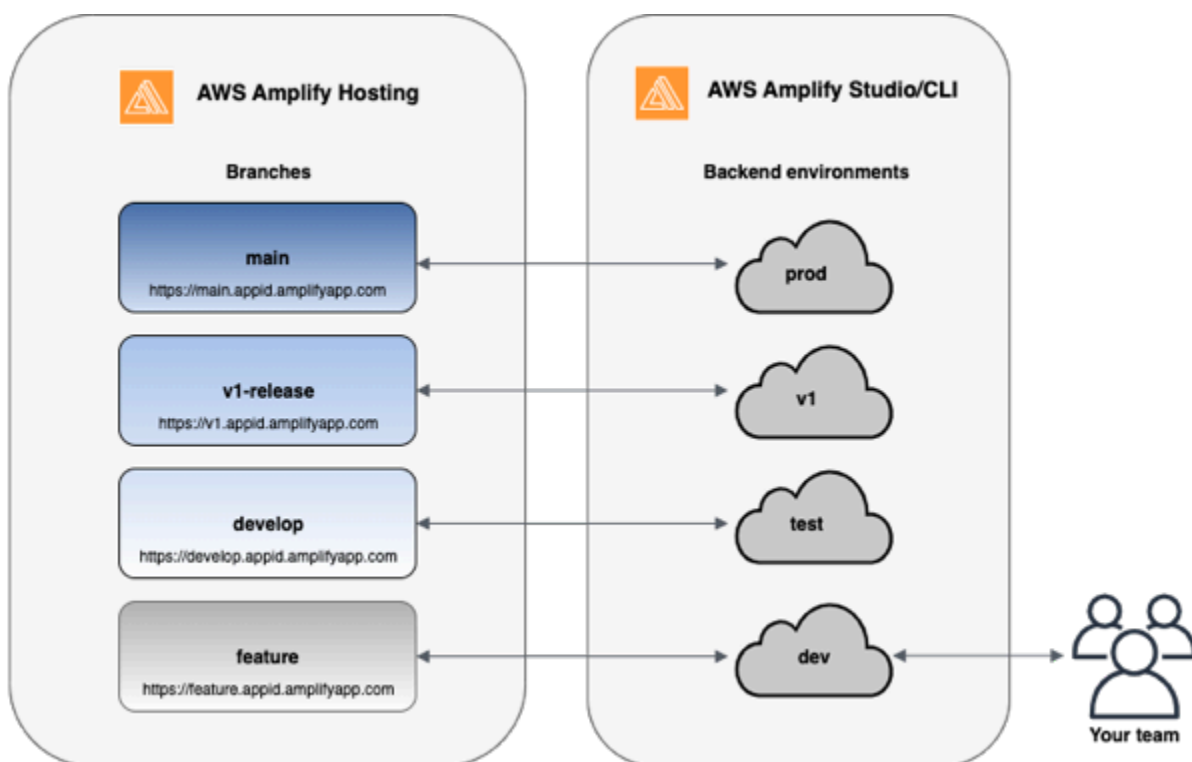
14. Hapus cabang fitur Anda dari Git, Amplify, dan hapus lingkungan backend dari cloud (Anda selalu dapat memutar yang baru berdasarkan dengan menjalankan 'amplify env checkout prod' dan menjalankan 'amplify env add').

```
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).

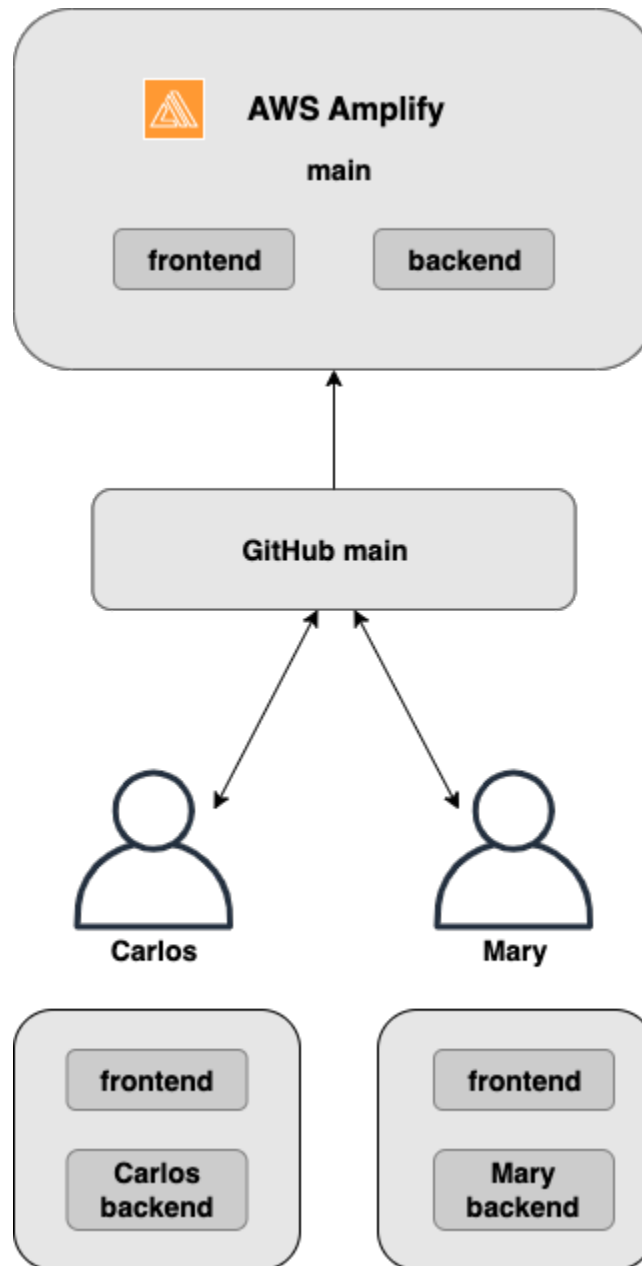
Diagram di bawah ini menunjukkan pengaturan yang disarankan dengan GitFlow. Anda dapat mengikuti proses yang sama seperti yang dijelaskan di bagian alur kerja cabang fitur di atas.



## 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 dapat mengakses deployment cabang main di <https://main.appid.amplifyapp.com> dengan lingkungan backend baru yang tertaut ke tersebut.
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. [Postingan blog ini](#) menjelaskan penggunaan fitur ini dengan berbagai alur kerja tim.

1. Pilih Pengaturan aplikasi > Pengaturan cabang > Edit.
2. Pilih Deteksi otomatis cabang untuk secara otomatis menghubungkan cabang ke Amplify yang cocok dengan kumpulan pola.
3. 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*`.
4. Siapkan perlindungan kata sandi otomatis untuk semua cabang yang dibuat secara otomatis dengan memilih kontrol akses deteksi otomatis Cabang.
  5. 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.
  6. 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 [Mengatur 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](#)

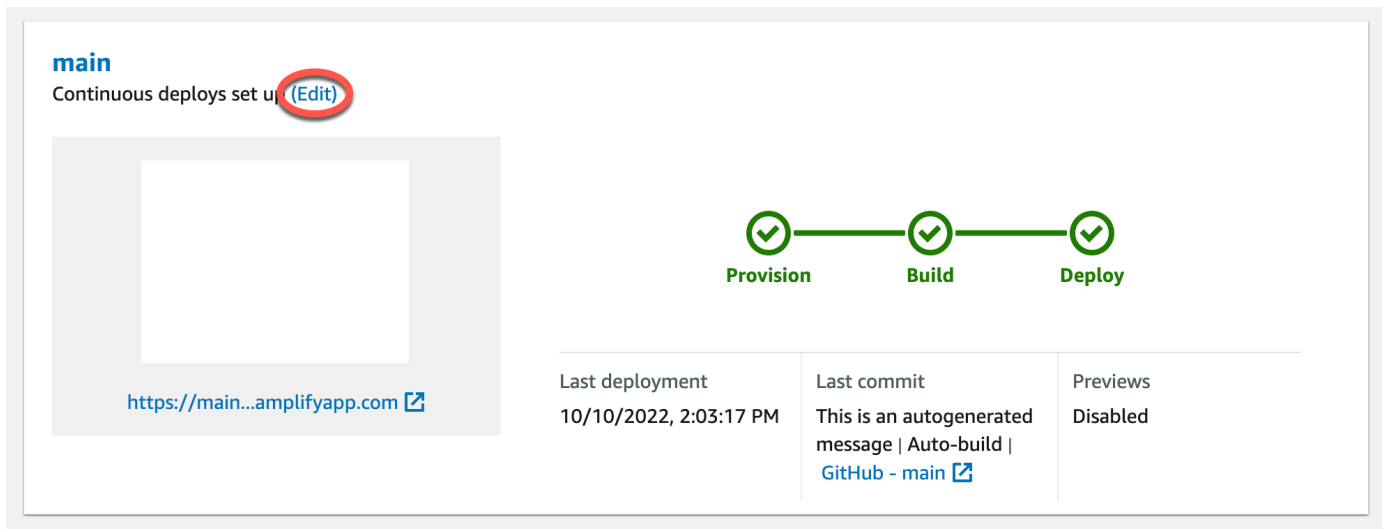
Amplify mendukung pembuatan waktu pembuatan otomatis dari file konfigurasi `aws-exports.js` Amplify untuk aplikasi Gen 1. Jika Anda menonaktifkan deployment CI/CD full stack, aplikasi Anda



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.

## Edit target backend

Select a backend environment to use with this branch

App name

Example-Amplify-App (this app) ▼

Environment

dev ▼

Enable full-stack continuous deployments (CI/CD)

Full-stack CI/CD allows you to continuously deploy frontend and backend changes on every code commit

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](#).

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 [Mengaktifkan atau menonaktifkan 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 setelah build di `amplify.yml` file aplikasi Anda. Untuk informasi selengkapnya tentang memperbarui `amplify.yml` file, lihat [Buat perintah dan pengaturan 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, CI/CD full-stack dimatikan. Menonaktifkan CI/CD full stack 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](#).
  - e. Pilih Selanjutnya.
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

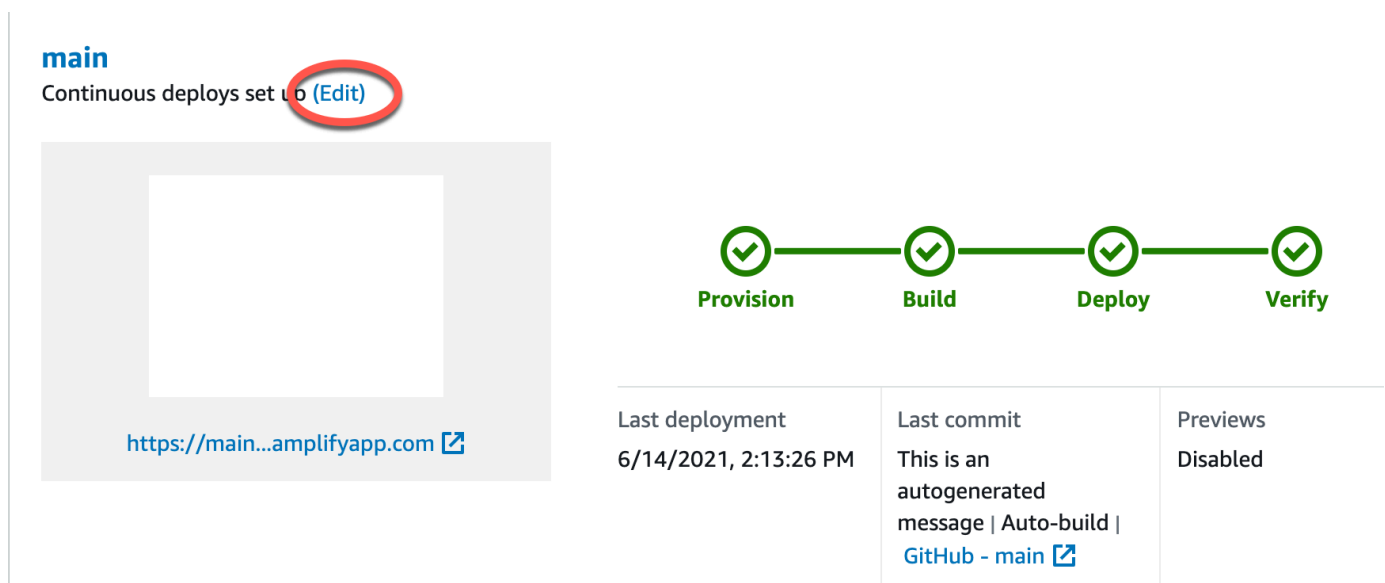
1. Masuk ke AWS Management Console dan buka konsol [Amplify](#).
2. Pilih aplikasi yang akan dihubungkan ke cabang baru.
3. Di panel navigasi, pilih Pengaturan Aplikasi, Umum.
4. Di bagian Cabang, pilih Hubungkan cabang.
5. Di halaman Tambahkan cabang repositori, untuk Cabang, pilih cabang dari repositori 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](#).
9. Secara default, CI/CD full-stack dimatikan. Menonaktifkan CI/CD full stack 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 Selanjutnya.
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.



**main**  
Continuous deploys set up (Edit)

<https://main...amplifyapp.com>

Provision Build Deploy Verify

Last deployment 6/14/2021, 2:13:26 PM	Last commit This is an autogenerated message   Auto-build   <a href="#">GitHub - main</a>	Previews Disabled
--	--	----------------------

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. CI/CD full stack diaktifkan secara default. Hapus centang opsi ini untuk menonaktifkan CI/CD full stack untuk backend ini. Menonaktifkan CI/CD full stack 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](#)
- [Buat backend untuk aplikasi Gen 1](#)

## 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: Menyebarakan 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, lihat [Menyiapkan akses Amplify ke GitHub repositori](#).

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 Selanjutnya.
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.

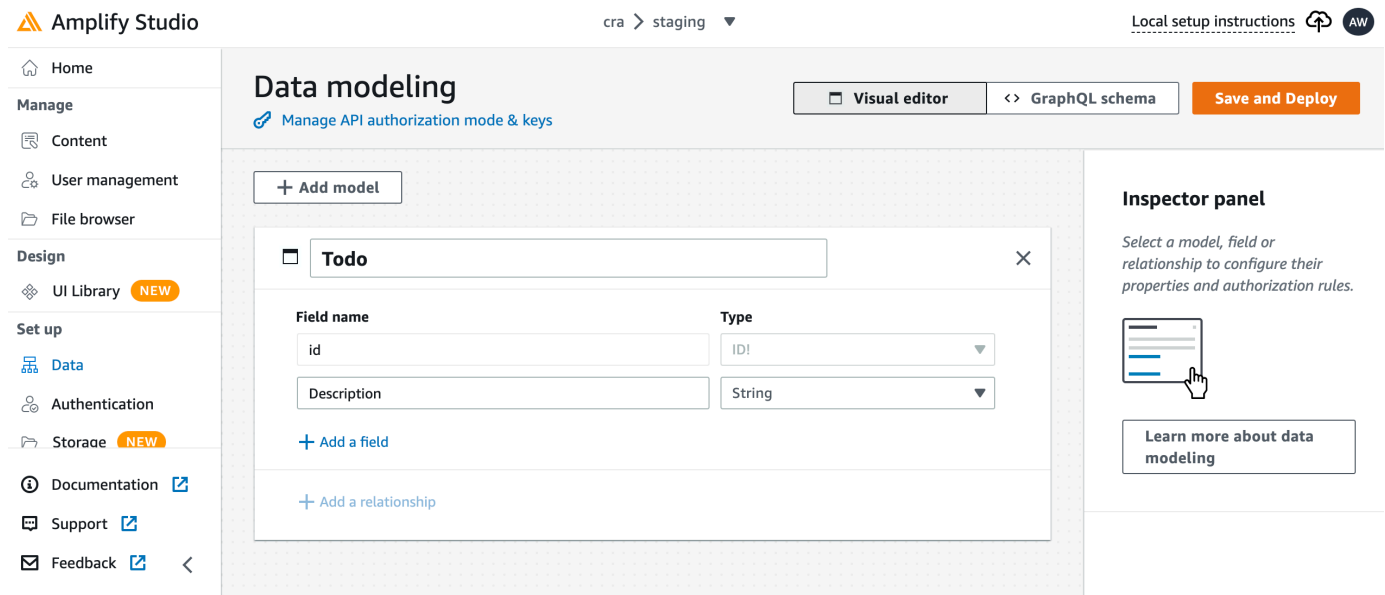
Buat model data

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.



- Untuk judul, masukkan **Todo**.
- Pilih Tambahkan bidang.
- Untuk nama Field, masukkan **Description**.

Screenshot berikut adalah contoh bagaimana model data Anda akan terlihat di desainer.



- Pilih Simpan dan Terapkan.
- 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

- Buka jendela terminal dan arahkan ke direktori root proyek lokal Anda.
- Jalankan perintah berikut di jendela terminal, ganti teks merah dengan ID aplikasi unik dan nama lingkungan backend untuk proyek Anda.

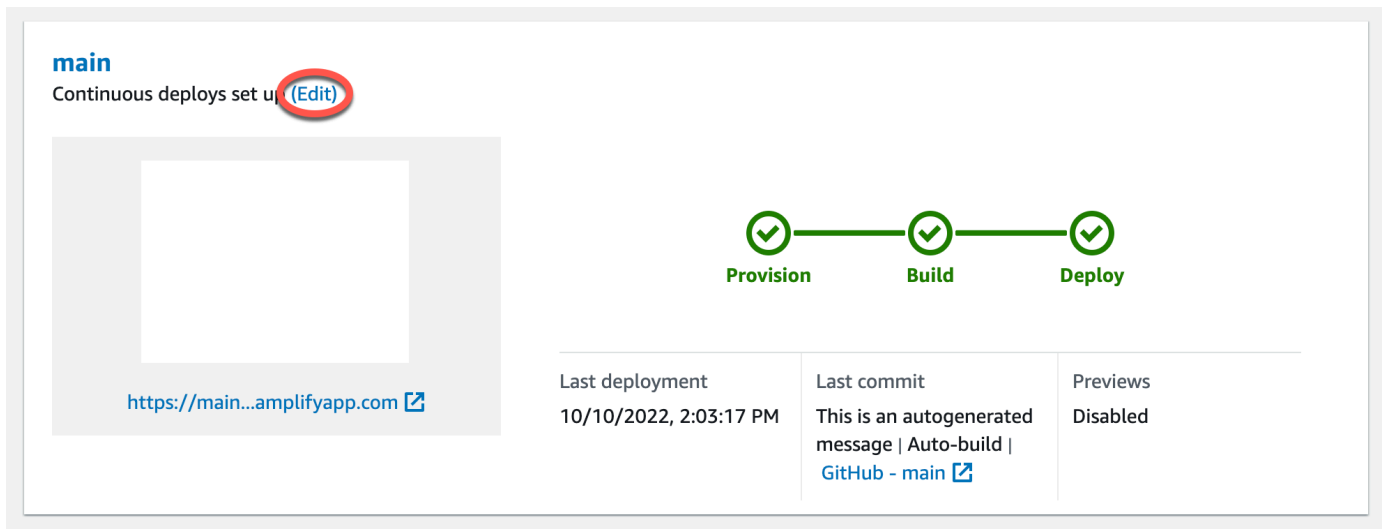
```
amplify pull --appId abcd1234 --envName staging
```

- 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

- Di beranda aplikasi, pilih tab Lingkungan hosting.
- Temukan cabang utama dan pilih Edit.



- 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.

CI/CD full stack diaktifkan secara default. Hapus centang opsi ini untuk menonaktifkan CI/CD full stack untuk backend ini. Menonaktifkan CI/CD full stack 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.

- 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, lihat [Menambahkan peran layanan](#).
- 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](#).

# Deployment manual

Penerapan manual memungkinkan Anda memublikasikan aplikasi web Anda dengan Amplify Hosting tanpa menghubungkan penyedia Git. Anda dapat menarik dan melepas folder 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.

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 [Terapkan 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. Di bagian Mulai deployment 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.

# Penyebaran manual Amazon S3 atau URL

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. Di bagian Mulai deployment 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 Amazon S3 Bucket, pilih nama bucket Amazon S3 dari daftar. Daftar kontrol akses (ACL) harus diaktifkan untuk bucket yang Anda pilih. Untuk informasi selengkapnya, lihat [Memecahkan masalah akses bucket Amazon S3](#).
    - b. Untuk File zip, pilih nama file zip yang akan di-deploy.
  - URL mana saja
    - Untuk URL sumber daya, masukkan URL ke file zip yang akan di-deploy.
8. Pilih Simpan dan deploy.

## Note

Ketika Anda membuat folder zip, pastikan Anda membuat folder zip untuk isi output build, 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

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

Jika Anda mendapatkan `AccessControlList` kesalahan saat menerapkan dari bucket Amazon S3, bucket dibuat dengan ACL 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.

## Terapkan ke tombol Amplify

Tombol Deploy to Amplify Hosting memungkinkan Anda untuk berbagi GitHub proyek secara publik atau di dalam tim Anda. Berikut gambar tombol ini:



## Tambahkan tombol Deploy to Amplify Hosting ke repositori atau blog

Tambahkan tombol ke file GitHub README.md Anda, posting blog, atau halaman markup lainnya yang membuat 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 GitHub repositori Anda. Anda dapat menyalin URL repositori Anda, seperti `https://github.com/username/repository`, atau Anda dapat memberikan deep link ke folder tertentu, seperti `https://github.com/username/repository/tree/branchname/folder` Amplify Hosting akan menyebarkan 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 untuk setiap 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">  
    
</a>
```

## Menyiapkan akses Amplify ke GitHub repositori

Amplify sekarang menggunakan fitur GitHub Apps untuk mengotorisasi akses read-only Amplify ke GitHub repositori. Dengan GitHub Aplikasi Amplify, izin lebih baik disetel, memungkinkan Anda untuk memberikan akses Amplify 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 di GitHub repo, secara default Amplify menggunakan GitHub Aplikasi untuk mengakses repo. Namun, aplikasi Amplify yang ada yang sebelumnya Anda sambungkan dari GitHub repo menggunakan OAuth untuk akses. CI/CD akan terus berfungsi untuk aplikasi ini, tetapi kami sangat menyarankan Anda memigrasinya untuk menggunakan GitHub Aplikasi Amplify baru.

Saat Anda menerapkan aplikasi baru atau memigrasi aplikasi yang ada menggunakan konsol Amplify, Anda secara otomatis diarahkan ke lokasi penginstalan untuk GitHub Aplikasi Amplify. Untuk mengakses halaman arahan penginstalan 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 Anda. Misalnya, untuk menginstal GitHub Aplikasi Amplify di wilayah AS Barat (Oregon), navigasikan ke `https://github.com/apps/aws-amplify-us-west-2`.

### Topik

- [Menginstal dan mengotorisasi GitHub Aplikasi Amplify untuk penerapan baru](#)
- [Memigrasi aplikasi yang ada ke OAuth Aplikasi Amplify GitHub](#)
- [Menyiapkan GitHub Aplikasi Amplify untuk AWS CloudFormation, CLI, dan penerapan SDK](#)
- [Menyiapkan pratinjau web dengan GitHub Aplikasi Amplify](#)

## Menginstal dan mengotorisasi GitHub Aplikasi 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 GitHub Aplikasi.

Untuk menginstal dan mengotorisasi GitHub Aplikasi Amplify

1. Masuk ke AWS Management Console, lalu buka [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 GitHub akun Anda. Pilih Izinkan.
5. Selanjutnya, Anda harus menginstal GitHub Aplikasi Amplify di GitHub akun Anda. Sebuah halaman terbuka di GitHub.com meminta izin untuk menginstal dan mengotorisasi AWS Amplify di GitHub akun Anda.
6. Pilih GitHub akun tempat Anda ingin menginstal GitHub Aplikasi Amplify.
7. Lakukan salah satu dari 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 akan diarahkan ke halaman Add repository branch 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 untuk dihubungkan.
12. Pilih Selanjutnya.
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 menggunakan OAuth untuk akses repo. Kami sangat menyarankan Anda memigrasikan aplikasi ini untuk menggunakan GitHub Aplikasi Amplify.

Gunakan petunjuk berikut untuk memigrasi aplikasi dan menghapus webhook OAuth yang terkait di GitHub akun Anda. Perhatikan bahwa prosedur migrasi bervariasi tergantung pada apakah GitHub aplikasi Amplify sudah diinstal. Setelah memigrasi aplikasi pertama dan menginstal dan mengotorisasi GitHub Aplikasi, Anda hanya perlu memperbarui izin repositori untuk migrasi aplikasi berikutnya.

## Memigrasi aplikasi dari OAuth ke GitHub Aplikasi

1. Masuk ke AWS Management Console, lalu buka [Konsol Amplify](#).
2. Pilih aplikasi yang ingin Anda migrasikan.
3. Di halaman informasi aplikasi, cari warna biru Migrasi ke pesan 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 GitHub akun Anda. Pilih Izinkan.
6. Pilih GitHub akun tempat Anda ingin menginstal GitHub Aplikasi Amplify.
7. Lakukan salah satu dari 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 akan 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 Anda yang ada, membuat yang baru, dan menyelesaikan migrasi.

## Menyiapkan GitHub Aplikasi Amplify untuk AWS CloudFormation, CLI, dan penerapan SDK

Aplikasi Amplify yang ada yang sebelumnya Anda sambungkan dari GitHub repositori menggunakan OAuth untuk akses repo. Ini dapat mencakup aplikasi yang Anda gunakan menggunakan Amplify Command Line Interface (CLI) AWS CloudFormation, atau SDK. Kami sangat menyarankan Anda memigrasikan aplikasi ini untuk menggunakan GitHub Aplikasi Amplify 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 SDK untuk menerapkan aplikasi Amplify baru yang menggunakan GitHub Aplikasi untuk akses repo. Proses ini mengharuskan Anda

menginstal GitHub Aplikasi Amplify terlebih dahulu di GitHub akun Anda. 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 instalasi untuk GitHub Aplikasi Amplify di AWS Wilayah tempat Anda akan menerapkan aplikasi Anda.

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 GitHub aplikasi Amplify.
3. Lakukan salah satu dari 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 Select scopes, 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 SDK.

Setelah GitHub aplikasi Amplify diinstal di GitHub akun Anda dan Anda telah membuat token akses pribadi, Anda dapat menerapkan aplikasi baru dengan Amplify CLI, AWS CloudFormation, atau

SDK. Gunakan `accessToken` bidang untuk menentukan token akses pribadi yang Anda buat di prosedur sebelumnya. Untuk informasi selengkapnya, lihat [CreateApp](#) di referensi Amplify API dan [AWS::Amplify::App](#) di Panduan AWS CloudFormation Pengguna.

Perintah CLI berikut menerapkan aplikasi Amplify baru yang menggunakan GitHub App untuk akses repositori. Ganti `myapp-using-githubapp`, `https://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 GitHub Aplikasi Amplify

Pratinjau web men-deploy setiap permintaan tarik pada GitHub repositori Anda ke URL pratinjau unik. Pratinjau sekarang menggunakan GitHub Aplikasi Amplify untuk akses ke GitHub repo Anda. Untuk petunjuk tentang menginstal dan mengotorisasi GitHub App untuk pratinjau web, lihat [Mengaktifkan pratinjau web](#).

## Pratinjau web untuk permintaan tarik

Dengan pratinjau web, tim pengembangan dan penjaminan kualitas (QA) dapat melihat pratinjau perubahan dari permintaan tarik (PR) 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 menyebarkan setiap permintaan tarik yang dibuat ke repositori Anda ke URL pratinjau unik yang sama sekali berbeda dari URL yang digunakan situs utama Anda. Untuk aplikasi dengan lingkungan backend yang disediakan menggunakan Amplify CLI atau Amplify Studio, setiap permintaan tarik (hanya repositori Git pribadi) membuat backend sementara yang dihapus saat PR ditutup.

Saat pratinjau web diaktifkan untuk aplikasi Anda, setiap PR dihitung dalam kuota Amplify 50 cabang per aplikasi. Untuk menghindari melebihi kuota ini, pastikan untuk menutup PR Anda. Untuk informasi lebih lanjut tentang kuota, lihat [Kuota layanan Amplify Hosting](#).

### Note

Saat ini, variabel `AWS_PULL_REQUEST_ID` lingkungan tidak tersedia saat digunakan AWS CodeCommit sebagai penyedia repositori Anda.

## Mengaktifkan pratinjau web

Untuk aplikasi yang disimpan dalam GitHub repo, pratinjau 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 menggunakan OAuth untuk akses, Anda harus terlebih dahulu memigrasikan aplikasi untuk menggunakan Aplikasi Amplify. GitHub Untuk petunjuk migrasi, lihat [Memigrasi aplikasi yang ada ke OAuth Aplikasi Amplify GitHub](#).

### Important

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.

## Cara mengaktifkan pratinjau web untuk permintaan tarik

1. Pilih Hosting, lalu Pratinjau.

### Note

Pratinjau terlihat di menu Pengaturan aplikasi hanya ketika aplikasi disiapkan untuk penerapan berkelanjutan 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 hal berikut ini:
    - 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 Anda, 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. Lalu, 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.

Lain kali Anda mengirimkan permintaan tarik untuk cabang, Amplify membangun dan menerapkan 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 Anda 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 khusus 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 [Mengatur subdomain otomatis untuk domain kustom Amazon Route 53](#).

# Tambahkan pengujian end-to-end Cypress ke aplikasi Amplify

Anda dapat menjalankan pengujian end-to-end (E2E) dalam fase pengujian aplikasi Amplify untuk menangkap regresi sebelum mendorong kode ke produksi. Fase pengujian dapat dikonfigurasi dalam spesifikasi build YAML. Saat ini, Anda hanya dapat menjalankan kerangka pengujian Cypress selama pembuatan.

## Tutorial: Mengatur end-to-end tes dengan Cypress

Cypress adalah kerangka pengujian JavaScript berbasis yang memungkinkan Anda menjalankan tes E2E di browser. 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.

## Tambahkan pengujian ke aplikasi Amplify yang ada

Anda dapat menambahkan pengujian Cypress ke aplikasi yang sudah ada dengan memperbarui setelan build aplikasi di konsol Amplify. Spesifikasi build YAMAL berisi kumpulan perintah build dan setelan terkait yang Amplify gunakan untuk menjalankan build Anda. Gunakan test langkah untuk menjalankan perintah pengujian apa pun pada waktu pembuatan. Untuk pengujian E2E, Amplify Hosting menawarkan integrasi yang lebih dalam dengan Cypress yang memungkinkan Anda membuat laporan UI untuk pengujian Anda.

Daftar berikut menjelaskan pengaturan pengujian dan bagaimana mereka digunakan.

### Pretest

Instal dependensi yang diperlukan untuk menjalankan tes Cypress. Amplify Hosting menggunakan [mochawesome](#) untuk menghasilkan laporan guna melihat hasil pengujian Anda dan [menunggu untuk menyiapkan server localhost](#) selama pembuatan.

### pengujian

Jalankan perintah cypress untuk melakukan tes menggunakan mochawesome.



## PostTest

Laporan mochawesome dihasilkan dari output JSON. 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
```

### Artefak>BaseDirectory

Direktori dari mana tes dijalankan.

artefak> configFilePath

Data laporan pengujian yang dihasilkan.

artefak>file

Artefak yang dihasilkan (tangkapan layar dan video) 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
  configFile: '**/mochawesome.json'
  files:
    - '**/*.png'
    - '**/*.mp4'

```

## Menonaktifkan pengujian

Setelah konfigurasi pengujian ditambahkan ke setelan `amplify.yml` build Anda, test langkah berjalan untuk setiap build, di setiap cabang. Jika Anda ingin menonaktifkan pengujian secara global agar tidak berjalan, atau hanya menjalankan pengujian untuk cabang tertentu, Anda dapat menggunakan variabel `USER_DISABLE_TESTS` lingkungan tanpa mengubah setelan 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.

### Environment Variables Manage variables

Environment variables are key/value pairs that contain any constant values your app needs at build time. For instance, database connection details or third party API keys. [Learn more](#) ↗

Branch ↕	Variable ↕	Value ↕
All branches	USER_DISABLE_TESTS	True


Rows per page

↕
|<
<
1
>
>|





Untuk menonaktifkan pengujian untuk cabang tertentu, tambahkan variabel `USER_DISABLE_TESTS` lingkungan dengan nilai `false` untuk semua cabang, lalu tambahkan penggantian untuk setiap cabang yang ingin Anda nonaktifkan dengan nilai `true`. Pada tangkapan layar berikut, tes dinonaktifkan di cabang utama, dan diaktifkan untuk setiap cabang lainnya.

## Environment Variables

[Manage variables](#)

Environment variables are key/value pairs that contain any constant values your app needs at build time. For instance, database connection details or third party API keys. [Learn more](#) 

Branch ▾	Variable ▾	Value ▾
All branches	USER_DISABLE_TESTS	False
main	USER_DISABLE_TESTS	True

Rows per page 15 ▾   1  

Menonaktifkan pengujian dengan variabel ini akan menyebabkan langkah pengujian dilewati sepenuhnya selama build. Untuk mengaktifkan kembali pengujian, atur nilai ini ke `false`, atau hapus variabel lingkungan.

# Menggunakan pengalihan

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.

## Jenis pengalihan

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 perbandingan A/B dari 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.

# Membuat dan mengedit pengalihan

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. Prosedur untuk menambahkan pengalihan bervariasi tergantung pada apakah Anda ingin menambahkan aturan satu per satu atau melakukan pengeditan massal:
  - Untuk membuat pengalihan individual, pilih Tambahkan penulisan ulang.
    - a. Untuk alamat Sumber, masukkan alamat asli yang diminta pengguna.
    - b. Untuk alamat Target, masukkan alamat tujuan yang merender konten ke pengguna.
    - c. Untuk Jenis, pilih jenis pengalihan dari daftar.
    - d. (Opsional) Untuk kode Negara, masukkan kondisi kode negara dua huruf.
  - Untuk mengedit pengalihan massal, pilih Buka editor teks.
    - Tambahkan atau perbarui pengalihan secara manual di Menulis ulang dan mengarahkan editor JSON.
6. Pilih Simpan.

## Urutan pengalihan

Pengalihan dijalankan 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
```

## 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.

## Pengalihan dan penulisan ulang sederhana

Bagian ini mencakup kode contoh untuk skenario pengalihan umum.

### Note

Pencocokan domain alamat asli tidak peka huruf besar/kecil.

Anda dapat menggunakan kode contoh berikut untuk mengalihkan secara permanen suatu halaman ke alamat baru.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
<code>/original.html</code>	<code>/destination.html</code>	permanent redirect (301)	

```
JSON [{"source": "/original.html", "status": "301", "target": "/destination.html", "condition": null}]
```

Anda dapat menggunakan kode contoh berikut untuk mengalihkan path di suatu folder ke path yang sama di folder berbeda.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
/docs/<*>	/documents/<*>	permanent redirect (301)	

```
JSON [{"source": "/docs/<*>", "status": "301", "target": "/documents/<*>", "condition": null}]
```

Anda dapat menggunakan kode contoh berikut untuk mengalihkan 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)	

```
JSON [{"source": "/<*>", "status": "200", "target": "/index.html", "condition": null}]
```

Anda dapat menggunakan kode contoh berikut untuk menggunakan penulisan ulang guna mengubah subdomain yang ditampilkan kepada pengguna.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
https://mydomain.com	https://www.mydomain.com	rewrite (200)	

```
JSON [{"source": "https://mydomain.com", "status": "200", "target": "https://www.mydomain.com", "condition": null}]
```

Anda dapat menggunakan kode 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)	

```
JSON [{"source": "https://mydomain.com", "status": "302", "target": "https://www.mydomain.com/documents/", "condition": null}]
```

Anda dapat menggunakan kode contoh berikut untuk mengalihkan path di folder yang tidak dapat ditemukan ke halaman 404 kustom.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
/<*>	/404.html	not found (404)	

```
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 peramban tanpa memicu 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	/index.html	200	

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
<code>json webp)\$)([^\n.]+\$/&gt;</code>			

```
JSON [{"source": "</^[^\n.]+$/>\.(?!css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|webp)$)([^\n.]+$/>", "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.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
<code>/images/&lt;*&gt;</code>	<code>https://images.otherdomain.com/&lt;*&gt;</code>	<code>rewrite (200)</code>	

```
JSON [{"source": "/images/<*>", "status": "200", "target": "https://images.otherdomain.com/<*>", "condition": null}]
```

## Garis miring di belakang dan URL bersih

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 membuat URL bersih dengan menambahkan garis miring saat diperlukan. Tabel berikut menampilkan berbagai skenario:

Input pengguna di peramban	URL di bilah alamat	Dokumen ditampilkan
<code>/about</code>	<code>/about</code>	<code>/about.html</code>
<code>/about</code> (when <code>about.html</code> returns <code>404</code> )	<code>/about/</code>	<code>/about/index.html</code>

Input pengguna di peramban	URL di bilah alamat	Dokumen ditampilkan
/about/	/about/	/about/index.html

## Placeholder

Anda dapat menggunakan kode contoh berikut untuk mengalihkan path di struktur folder tertentu 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)	

```
JSON [{"source": "/docs/<year>/<month>/<date>/<itemid>", "status": "301", "target": "/documents/<year>/<month>/<date>/<itemid>", "condition": null}]
```

## String kueri dan parameter path

Anda dapat menggunakan kode contoh berikut untuk mengalihkan path ke folder dengan nama yang sesuai 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)	

```
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 kode contoh berikut untuk mengalihkan semua path yang tidak dapat ditemukan di tingkat tertentu struktur folder ke `index.html` di folder tertentu.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
<code>/documents/ &lt;folder&gt;/ &lt;child-folder&gt;/ &lt;grand-child- folder&gt;</code>	<code>/documents/ index.html</code>	not found (404)	

```
JSON [{"source": "/documents/<x>/<y>/<z>", "status": "404", "target": "/documents/index.html", "condition": null}]
```

## Pengalihan berbasis wilayah

Anda dapat menggunakan kode contoh berikut untuk mengalihkan permintaan berdasarkan wilayah.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
<code>/documents</code>	<code>/documents/us/</code>	temporary redirect (302)	<code>&lt;US&gt;</code>

```
JSON [{"source": "/documents", "status": "302", "target": "/documents/us/", "condition": "<US>"}]
```

## 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
<code>/docs/&lt;*&gt;</code>	<code>/documents/&lt;*&gt;</code>	permanent redirect (301)	

Dua contoh berikut menunjukkan pengalihan yang tidak valid dengan ekspresi wildcard.

Alamat asli	Alamat Tujuan	Jenis Pengalihan	Kode Negara
<code>/docs/&lt;*&gt;/content</code>	<code>/documents/&lt;*&gt;/content</code>	permanent redirect (301)	
<code>/docs/&lt;*&gt;/content/&lt;*&gt;</code>	<code>/documents/&lt;*&gt;/content/&lt;*&gt;</code>	permanent redirect (301)	

# Membatasi akses ke cabang

Jika Anda mengerjakan fitur yang belum dirilis, Anda dapat melindungi cabang fitur dengan kata sandi 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).

Untuk mengatur kata sandi di cabang fitur

1. Masuk ke AWS Management Console dan buka 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 Kelola kontrol akses, lakukan salah satu hal berikut.
  - Untuk menetapkan nama pengguna dan kata sandi yang berlaku untuk semua cabang yang terhubung
    - Aktifkan Kelola akses untuk semua cabang. Misalnya, jika Anda memiliki cabang utama, pengembang, dan fitur yang terhubung, Anda dapat menerapkan nama pengguna dan kata sandi yang sama untuk semua cabang.
  - Untuk menerapkan nama pengguna dan kata sandi ke cabang individu
    - 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.

## Variabel-variabel lingkungan

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 tidak memungkinkan Anda membuat variabel lingkungan dengan awalan. AWS Awalan ini dicadangkan untuk Amplify penggunaan internal saja.

### Important

Jangan gunakan variabel lingkungan untuk menyimpan rahasia. Simpan rahasia dalam rahasia lingkungan yang dibuat menggunakan AWS Systems Manager Parameter Store. Untuk informasi selengkapnya, lihat [Rahasia lingkungan](#).

## Amplify variabel lingkungan

Variabel lingkungan berikut dapat diakses secara default dalam konsol Amplify.

Nama variabel	Deskripsi	Nilai contoh
<code>_BUILD_TIMEOUT</code>	Durasi waktu tunggu build dalam hitungan menit	30
<code>_LIVE_UPDATES</code>	Alat akan ditingkatkan ke versi terbaru.	<code>[{"name": "Amplify CLI", "pkg": "@aws-amplify/cli", "type": "npm", "version": "latest"}]</code>
<code>USER_DISABLE_TESTS</code>	Langkah pengujian dilewati selama pembuatan. Anda dapat menonaktifkan pengujian untuk semua cabang atau cabang tertentu di aplikasi.	true

Nama variabel	Deskripsi	Nilai contoh
	Variabel lingkungan ini digunakan untuk aplikasi yang melakukan pengujian selama fase build. Untuk informasi selengkapnya tentang pengaturan variabel ini, lihat <a href="#">Menonaktifkan pengujian</a> .	
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
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



Nama variabel	Deskripsi	Nilai contoh
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
AMPLIFY_FACEBOOK_CLIENT_SECRET	Rahasia klien Facebook	example123456
AMPLIFY_GOOGLE_CLIENT_ID	ID klien Google	123456
AMPLIFY_GOOGLE_CLIENT_SECRET	Rahasia klien Google	example123456

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_DIFF_DEPLOY	Mengaktifkan atau menonaktifkan deployment frontend berbasis diff. Untuk informasi selengkapnya, lihat <a href="#">Mengaktifkan atau menonaktifkan build dan deploy frontend berbasis diff.</a>	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 <a href="#">Mengaktifkan atau menonaktifkan build backend berbasis diff untuk aplikasi Gen 1</a>	true
AMPLIFY_BACKEND_PULL_ONLY	Amplify mengelola variabel lingkungan ini. Ini hanya berlaku untuk aplikasi Gen 1. Untuk informasi selengkapnya, lihat <a href="#">Mengedit frontend yang ada agar mengarah ke backend berbeda</a>	true

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_BACKEND_APP_ID	Amplify mengelola variabel lingkungan ini. Ini hanya berlaku untuk aplikasi Gen 1. Untuk informasi selengkapnya, lihat <a href="#">Mengedit frontend yang ada agar mengarah ke backend berbeda</a>	abcd1234
AMPLIFY_SKIP_BACKEND_BUILD	Jika Anda tidak memiliki bagian backend dalam spesifikasi build dan ingin menonaktifkan build backend, setel variabel lingkungan ini ke <code>true</code> . Ini hanya berlaku untuk aplikasi Gen 1.	<code>true</code>
AMPLIFY_ENABLE_DEBUG_OUTPUT	Atur variabel ini <code>true</code> untuk mencetak jejak tumpukan di log. Ini berguna untuk men-debug kesalahan build backend.	<code>true</code>
AMPLIFY_MONOREPO_APP_ROOT	Path yang digunakan untuk menentukan root aplikasi dari aplikasi monorepo, bergantung pada root repositori.	<code>apps/react-app</code>
AMPLIFY_USERPOOL_ID	ID untuk kumpulan pengguna Amazon Cognito yang diimpor untuk autentikasi	<code>us-west-2_example</code>

Nama variabel	Deskripsi	Nilai contoh
AMPLIFY_WEBCLIENT_ID	<p>ID untuk klien aplikasi yang akan digunakan oleh aplikasi web</p> <p>Klien aplikasi harus dikonfigurasi dengan akses ke kumpulan pengguna Amazon Cognito yang ditentukan oleh variabel lingkungan AMPLIFY_USERPOOL_ID.</p>	123456
AMPLIFY_NATIVECLIENT_ID	<p>ID untuk klien aplikasi yang akan digunakan oleh aplikasi asli</p> <p>Klien aplikasi harus dikonfigurasi dengan akses ke kumpulan pengguna Amazon Cognito yang ditentukan oleh variabel lingkungan AMPLIFY_USERPOOL_ID.</p>	123456
AMPLIFY_IDENTITYPOOL_ID	ID untuk kumpulan identitas Amazon Cognito	example-identitypool-id
AMPLIFY_PERMISSIONS_BOUNDARY_ARN	<p>ARN untuk kebijakan IAM untuk digunakan sebagai batas izin yang berlaku untuk semua peran IAM yang dibuat oleh Amplify. Untuk informasi selengkapnya, lihat <a href="#">Batas Izin IAM untuk peran yang dihasilkan Amplify</a>.</p>	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 token OAuth, bukan kunci AWS akses dan kunci rahasia.

## Tetapkan variabel lingkungan

Gunakan petunjuk berikut untuk menyetel 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.

## Akses variabel lingkungan pada waktu pembuatan

Untuk mengakses variabel lingkungan selama build, edit pengaturan build sehingga menyertakan variabel lingkungan dalam perintah build.

Setiap perintah dalam konfigurasi build Anda berjalan di dalam shell Bash. Untuk informasi lebih lanjut seputar penggunaan variabel lingkungan di Bash, lihat [Ekspansi Shell](#) di Manual GNU Bash.

Untuk mengedit pengaturan build guna menyertakan variabel lingkungan

1. Masuk ke AWS Management Console dan buka konsol [Amplify](#).
2. Di konsol Amplify, pilih Hosting, lalu pilih Build settings.
3. Di bagian Spesifikasi build aplikasi, pilih Edit.
4. Tambahkan variabel lingkungan ke perintah build. Anda sekarang dapat mengakses variabel lingkungan selama build berikutnya. Contoh ini mengubah perilaku npm (BUILD\_ENV) dan menambahkan token API (TWITCH\_CLIENT\_ID) untuk layanan eksternal ke file lingkungan untuk digunakan nanti.

```
build:
  commands:
    - npm run build:$BUILD_ENV
    - echo "TWITCH_CLIENT_ID=$TWITCH_CLIENT_ID" >> backend/.env
```

5. Pilih Simpan.

## Membuat variabel lingkungan dapat diakses oleh runtime sisi server

Komponen server Next.js tidak memiliki akses ke variabel lingkungan aplikasi Anda 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 harus memodifikasi file spesifikasi build Amplify untuk mengatur variabel lingkungan dalam file lingkungan yang dikenali Next.js. Hal ini memungkinkan Amplify untuk memuat variabel lingkungan sebelum membangun aplikasi. Untuk informasi selengkapnya tentang memodifikasi spesifikasi build Anda, lihat contoh cara [menambahkan variabel lingkungan di bagian perintah build](#).

## 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 Build settings, cari bagian Select a backend environment to use with this branch. 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.

Select a backend environment to use with this branch

App name  
docs (this app) ▼

Environment  
Create new environment ▼

If you don't provide a value in this field, your branch name will be used by default.

backend

Enable full-stack continuous deployments (CI/CD)  
Full-stack CI/CD allows you to continuously deploy frontend and backend changes on every code commit

Select an existing service role or create a new one so Amplify Hosting may access your resources.

amplifyconsole-backend-role ▼

Create a new service role. In the window that opens, accept the pre-selected defaults on each screen to create a new service role.

[Create new role](#)

- 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 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.

## 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 konfigurasi 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. Untuk contoh penggunaan variabel lingkungan waktu build Amplify untuk tujuan ini, lihat [Akses variabel lingkungan pada waktu pembuatan](#).

## Rahasia lingkungan

Rahasia lingkungan mirip dengan variabel lingkungan, tetapi mereka adalah pasangan nilai kunci Parameter Store AWS Systems Manager (SSM) yang dapat dienkrpsi. Beberapa nilai harus dienkrpsi, seperti kunci pribadi Masuk dengan Apple untuk Amplify.



## Tetapkan rahasia lingkungan

Gunakan petunjuk berikut untuk menyetel rahasia lingkungan untuk aplikasi Amplify 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.

## Akses rahasia lingkungan

Mengakses rahasia lingkungan selama build mirip dengan [mengakses variabel lingkungan](#), kecuali bahwa rahasia lingkungan disimpan `process.env.secrets` sebagai string JSON.

## Amplify 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 -----

## Header kustom

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 AWS Management Console atau dengan mengunduh dan memperbarui `amplify.yml` file dan menyimpannya di direktori root proyek. Header kustom yang ditentukan dengan cara ini harus dimigrasi keluar buildspec dan file `amplify.yml`. Untuk instruksi, lihat [Migrasi header kustom](#).

## Format YAML 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 [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. 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.

## Migrasi header kustom

Sebelumnya, header HTTP kustom ditentukan untuk aplikasi baik dengan mengedit `buildspec` 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 dari `buildspec` 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.

## 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, lihat [Format YAML 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.

## Contoh header 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'"
```

## Header Cache-Control Kustom

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, lihat [Menggunakan header untuk mengendalikan durasi cache](#).

# Webhook masuk

Siapkan webhook masuk di konsol Amplify untuk memulai build tanpa memasukkan kode ke repositori Git Anda. Anda dapat menggunakan pemicu 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 AWS Management Console dan buka 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 memicu build.
  - Jalankan perintah curl di jendela terminal untuk memicu build baru.

# Pemantauan

AWS Amplify memancarkan metrik melalui Amazon CloudWatch dan menyediakan log akses dengan informasi terperinci tentang permintaan yang dibuat ke aplikasi Anda. Gunakan topik di bagian ini untuk mempelajari cara menggunakan metrik dan log ini untuk memantau aplikasi Anda.

Topik

- [Pemantauan CloudWatch dengan](#)
- [Log akses](#)

## Pemantauan CloudWatch dengan

AWS Amplify terintegrasi dengan Amazon CloudWatch, memungkinkan Anda memantau metrik untuk aplikasi Amplify Anda dalam waktu dekat. Anda dapat membuat alarm yang mengirim notifikasi jika metrik melebihi ambang batas yang ditetapkan. Untuk informasi selengkapnya tentang cara kerja CloudWatch layanan, lihat [Panduan CloudWatch Pengguna Amazon](#).

## Metrik

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. Dalam tabel berikut, statistik paling relevan dicantumkan di deskripsi untuk setiap metrik.

Metrik	Deskripsi
Permintaan	<p>Jumlah total permintaan pemirsa yang diterima aplikasi Anda.</p> <p>Statistik paling relevan adalah Sum. Gunakan statistik Sum untuk mendapatkan jumlah total permintaan.</p>

Metrik	Deskripsi
BytesDownloaded	<p>Jumlah total data yang ditransfer dari aplikasi Anda (diunduh) dalam byte oleh pemirsa untuk permintaan GET, HEAD, dan OPTIONS.</p> <p>Statistik paling relevan adalah Sum.</p>
BytesUploaded	<p>Jumlah total data yang ditransfer ke aplikasi Anda (diunggah) dalam byte menggunakan permintaan POST dan PUT.</p> <p>Statistik paling relevan adalah Sum.</p>
4XXKesalahan	<p>Jumlah permintaan yang menampilkan pesan kesalahan dalam kisaran kode status HTTP 400-499.</p> <p>Statistik paling relevan adalah Sum. Gunakan statistik Sum untuk mendapatkan total kejadian kesalahan ini.</p>
5XXKesalahan	<p>Jumlah permintaan yang menampilkan pesan kesalahan dalam kisaran kode status HTTP 500-599.</p> <p>Statistik paling relevan adalah Sum. Gunakan statistik Sum untuk mendapatkan total kejadian kesalahan ini.</p>

Metrik	Deskripsi
Latensi	<p>Waktu ke byte pertama dalam detik. Ini adalah total waktu antara saat Amplify Hosting menerima permintaan dan ketika mengembalikan respons ke jaringan. Waktu tidak termasuk latensi jaringan yang dialami untuk mengirimkan respons ke perangkat pemirsa.</p> <p>Statistik paling relevan adalah Average, Maximum, Minimum, p10, p50, p90, p95, dan p100.</p> <p>Gunakan statistik Average untuk mengevaluasi latensi yang diharapkan.</p>

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.

Anda dapat mengakses CloudWatch metrik AWS Management Console di <https://console.aws.amazon.com/cloudwatch/>. Atau, Anda dapat mengakses metrik di konsol Amplify menggunakan prosedur berikut.

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 Pengaturan Aplikasi, kemudian Pemantauan.
4. Di halaman Pemantauan, pilih Metrik.

## 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 API. 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 Pengaturan Aplikasi, kemudian Pemantauan.
4. Di halaman Pemantauan, pilih Alarm.
5. Pilih Buat alarm.
6. Di jendela Buat alarm, konfigurasi 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.
  - d. Untuk Atur notifikasi, lakukan salah satu langkah berikut:
    - 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 memicu alarm.
  - iv. Tentukan durasi periode evaluasi.
- e. Pilih Buat alarm.

 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.

## CloudWatch Log Amazon untuk aplikasi SSR

Amplify mengirimkan informasi tentang runtime Next.js ke Amazon CloudWatch Logs di file Anda. Akun AWSSaat 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](#). Untuk informasi selengkapnya tentang penerapan aplikasi yang dirender di sisi server, lihat. [Terapkan aplikasi yang dirender sisi server dengan Amplify Hosting](#)

## Log akses

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.

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 akuntansi 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.

Gunakan prosedur berikut untuk mengambil log akses untuk aplikasi.

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 Pengaturan Aplikasi, kemudian Pemantauan.
4. Di halaman Pemantauan, pilih Log akses.
5. Pilih Edit rentang waktu.
6. Di jendela Edit rentang waktu, untuk Tanggal mulai, tentukan hari pertama dalam interval dua minggu log diambil. Untuk Waktu mulai, pilih waktu pada hari pertama pengambilan log dimulai.



7. 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;
```

# Pemberitahuan email untuk build

Anda dapat mengatur notifikasi email untuk AWS Amplify aplikasi untuk memberi tahu pemangku kepentingan atau anggota tim saat 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 AWS Management Console dan buka 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 hal berikut ini:
  - 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.

# Gambar build kustom dan pembaruan paket langsung

## Topik

- [Gambar build kustom](#)
- [Pembaruan paket langsung](#)

## Gambar build kustom

Anda dapat menggunakan gambar build kustom untuk menyediakan lingkungan build khusus untuk aplikasi Amplify. Jika Anda memiliki dependensi tertentu yang membutuhkan waktu lama untuk diinstal selama build menggunakan container default Amplify, Anda dapat membuat image Docker Anda sendiri dan mereferensikannya selama pembuatan. Gambar dapat di-host di Amazon Elastic Container Registry Public.

### Note

Pengaturan build terlihat di menu Hosting 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](#).

## Ketentuan gambar build kustom

Agar image build kustom berfungsi sebagai image build Amplify, image tersebut harus memenuhi persyaratan berikut:

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 keluaran apa pun karena build-runner kami tidak dapat menghasilkan output apa pun.
3. Git: Git harus terinstal pada gambar untuk membuat klon repositori Git Anda. Jika ketergantungan ini hilang, langkah repositori Kloning akan gagal.

4. OpenSSH: Untuk mengkloning repositori Anda dengan aman, kami memerlukan OpenSSH untuk mengatur kunci SSH sementara selama pembuatan. 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 untuk menginstalnya.

1. NVM (Node Version Manager): Kami menyarankan Anda menginstal pengelola versi ini jika Anda perlu menangani versi yang berbeda dari Node. 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 khusus Anda.
3. Tar: Amplify dapat menggunakan Tar utilitas untuk membuka kompres file yang diunduh selama proses pembuatan. Kami menyarankan Anda menginstalnya di gambar khusus Anda.

## Mengonfigurasi gambar build kustom

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 AWS Management Console dan buka 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-exemplerepo`, Anda harus memasukkan **`public.ecr.aws/xxxxxxxx/ecr-exemplerepo`**.
8. Pilih Simpan.

## Pembaruan paket langsung

Pembaruan paket langsung memungkinkan Anda menentukan versi paket dan dependensi yang akan digunakan dalam image 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`.

## Mengonfigurasi pembaruan paket langsung

Untuk mengonfigurasi pembaruan paket langsung

1. Masuk ke AWS Management Console dan buka 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.

# Menambahkan peran layanan

Amplify memerlukan izin untuk menyebarkan 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 diasumsikan Amplify saat memanggil layanan lain atas nama Anda. Dalam panduan 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

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 Use case, pilih Amplify, lalu pilih, Next.
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 penerapan aplikasi baru
    - 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
    - a. Di panel navigasi, pilih Pengaturan aplikasi, lalu Pengaturan umum.
    - b. Pada halaman Pengaturan umum, pilih Edit.
    - c. Pada halaman Edit pengaturan umum, pilih peran yang baru saja Anda buat dari daftar peran Layanan.

d. Pilih Simpan.

7. Konsol Amplify sekarang memiliki izin untuk menerapkan resource backend untuk aplikasi Anda.

## Pencegahan Deputi Bingung

Masalah deputi yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. 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.



# Mengelola 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.

## Menggunakan header untuk mengendalikan durasi cache

Cache-ControlHeader 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 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).

Anda dapat menentukan header kustom untuk aplikasi di bagian Custom header pada konsol Amplify. Untuk contoh YAML format, lihat [Header Cache-Control Kustom](#).

## Menyetel header cache-control untuk meningkatkan performa aplikasi

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.

# Mencatat log panggilan API Amplify dengan AWS CloudTrail

AWS Amplify terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil pengguna, peran, atau AWS layanan di Amplify. CloudTrail merekam semua panggilan API untuk Amplify sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari konsol Amplify dan panggilan kode ke operasi API Amplify. Jika membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan CloudTrail kejadian ke bucket Amazon S3, termasuk kejadian untuk Amplify. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam CloudTrail konsol di Riwayat kejadian. Dengan menggunakan informasi yang CloudTrail dikumpulkan, Anda dapat menentukan permintaan yang dibuat ke Amplify, alamat IP asal permintaan tersebut dibuat, pembuat permintaan, waktu permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjut CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

## Informasi Amplify di CloudTrail

CloudTrail diaktifkan di AWS akun Anda secara default. Aktivitas yang terjadi di Amplify akan dicatat dalam CloudTrail kejadian bersama kejadian AWS layanan lainnya di Riwayat kejadian. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat CloudTrail peristiwa dengan riwayat peristiwa](#) di Panduan AWS CloudTrail Pengguna.

Untuk catatan peristiwa yang sedang berlangsung di akun AWS Anda, termasuk peristiwa untuk Amplify, buat jejak. Jejak CloudTrail memungkinkan mengirim file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data kejadian yang dikumpulkan di CloudTrail log. Untuk informasi selengkapnya, pelajari topik berikut di Panduan Pengguna AWS CloudTrail:

- [Membuat jejak untuk AWS akun Anda](#)
- [CloudTrail Layanan yang didukung dan integrasi](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima berkas CloudTrail log dari beberapa wilayah](#) dan [Menerima berkas CloudTrail log dari beberapa akun](#)

Semua operasi Amplify dicatat oleh CloudTrail dan didokumentasikan di [AWS AmplifyReferensi API UI AdminAWS Amplify](#), dan [Referensi API UI Admin Amplify](#). 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 hal berikut:

- Permintaan dibuat dengan kredensial pengguna root atau AWS Identity and Access Management (IAM).
- Permintaan dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat [elemenCloudTrail userIdentity](#) di PanduanAWS CloudTrail Pengguna.

## Memahami entri berkas log Amplify

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang Anda tentukan. CloudTrail File log berisi satu atau beberapa entri log. Sebuah peristiwa mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail Berkas log bukan merupakan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan [ListApps](#) operasi Referensi APIAWS 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 UIAWS 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"  
}
```

# 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 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](#) . 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](#)
- [Perlindungan Data di Amplify](#)
- [Validasi Kepatuhan untuk AWS Amplify](#)
- [Keamanan Infrastruktur di AWS Amplify](#)
- [Pencatatan dan pemantauan peristiwa keamanan di Amplify](#)
- [Pencegahan confused deputy lintas layanan](#)
- [Praktik terbaik keamanan untuk Amplify](#)

## 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](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Cara Amplify bekerja dengan IAM](#)
- [Contoh kebijakan berbasis identitas untuk Amplify](#)
- [AWS kebijakan terkelola untuk AWS Amplify](#)
- [Pemecahan masalah identitas dan akses Amplify](#)

## 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 kredensial 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 kredensial 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 kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di 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 [Menggunakan autentikasi multi-faktor \(MFA\) dalam AWS](#) 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 kredensial 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 memiliki grup yang bernama IAMAdmins dan memberikan izin ke 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 [Kapan harus membuat pengguna IAM \(bukan peran\)](#) 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. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). 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 [Menggunakan peran IAM](#) 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 mengotentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat 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 [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda memanggil 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 [Membuat 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 instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan dalam instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam 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 [Membuat kebijakan IAM](#) 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 [Memilih 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 (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) dalam Panduan Developer Amazon Simple Storage Service.

## 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 (SCP) — SCP 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 di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) 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 Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .



- 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
<a href="#">Kebijakan berbasis identitas</a>	Ya
<a href="#">Kebijakan berbasis sumber daya</a>	Tidak
<a href="#">Tindakan kebijakan</a>	Ya
<a href="#">Sumber daya kebijakan</a>	Ya
<a href="#">Kunci kondisi kebijakan</a>	Ya
<a href="#">ACL</a>	Tidak
<a href="#">ABAC (tanda dalam kebijakan)</a>	Parsial
<a href="#">Kredensial sementara</a>	Ya
<a href="#">Sesi akses teruskan (FAS)</a>	Ya



Fitur IAM	Dukungan Amplify
<a href="#">Peran layanan</a>	Ya
<a href="#">Peran terkait layanan</a>	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 [Membuat kebijakan IAM](#) 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 prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Perbedaan IAM role dan kebijakan berbasis sumber daya](#) di 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 support izin tingkat sumber daya, seperti operasi daftar, gunakan wildcard (\*) untuk menunjukkan bahwa pernyataan berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk daftar jenis sumber daya Amplify dan ARN masing-masing, lihat [Jenis sumber daya yang ditentukan AWS Amplify](#) di 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	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 `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 tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM: variabel dan tag](#) 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 (ACL) di Amplify

Mendukung ACL

Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

## Kontrol akses berbasis atribut (ABAC) dengan Amplify

Mendukung ABAC (tanda dalam kebijakan)

Parsial

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 tag milik prinsipal cocok dengan tag 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 tag, berikan informasi tentang tag 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 [Apa itu 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 penggunaan kredensial sementara

Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensial 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 kredensial 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 [Peralihan peran \(konsol\)](#) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensial sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensial 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 [Membuat 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 akan dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amplify, termasuk format ARN 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 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 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 [Mengonfigurasi akses API yang dilindungi MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan dalam 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 `ReadOnlyAWS 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 [AWS kebijakan yang dikelola](#) 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.
- `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.

Untuk melihat izin kebijakan ini, lihat [AmplifyBackendDeployFullAccess](#) di 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
<a href="#">AmplifyBackendDeployFullAccess</a> — Pembaruan ke kebijakan yang sudah ada	Tambahkan akses baca ke <code>arn:aws:ssm:*:*:parameter/cdk-bootstrap/*</code> sumber daya untuk memungkinkan Amplify mendeteksi versi bootstrap CDK di akun pelanggan.	31 Mei 2024
<a href="#">AmplifyBackendDeployFullAccess</a> – Pembaruan ke kebijakan yang ada	Tambahkan pernyataan <code>AmplifyDiscoverRDSVpcConfig</code> kebijakan baru dengan izin hanya-baca Amazon RDS dan Amazon EC2 yang dicakup oleh sumber daya dan kondisi akun. Izin ini mendukung perintah Amplify Gen <code>npx amplify generate schema-from-database 2</code> yang memungkinkan pelanggan menghasilkan skema data TypeScript dari database SQL yang ada.  Tambahkan <code>rds:DescribeDBProxies</code> , <code>rds:DescribeDBInstances</code> <code>rds:DescribeDBClusters</code> , <code>rds:DescribeDBSubnetGroups</code> ,	April 17, 2024

Perubahan	Deskripsi	Tanggal
	<p>dan <code>ec2:DescribeSubnets</code> izin. <code>npx amplify generate schema-from-database</code> Perintah ini memerlukan izin ini untuk memeriksa apakah host DB tertentu di-host di Amazon RDS dan secara otomatis menghasilkan konfigurasi VPC Amazon yang diperlukan untuk menyediakan sumber daya lain yang diperlukan untuk menyiapkan AWS AppSync API yang didukung oleh database SQL.</p>	
<p><a href="#">AmplifyBackendDeployFullAccess</a> – Pembaruan ke kebijakan yang ada</p>	<p>Tambahkan tindakan <code>cloudformation:DeleteStack</code> kebijakan untuk mendukung penghapusan tumpukan saat <code>DeleteBranch</code> API dipanggil.</p> <p>Tambahkan tindakan <code>lambda:GetFunction</code> kebijakan untuk mendukung fungsi hotswapping.</p> <p>Tambahkan tindakan <code>lambda:UpdateFunctionConfiguration</code> kebijakan untuk mendukung pembaruan fungsi Lambda.</p>	<p>April 5, 2024</p>

Perubahan	Deskripsi	Tanggal
<a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada	Tambahkan <code>cloudformation:TagResource</code> dan <code>cloudformation:UntagResource</code> izin untuk mendukung panggilan ke AWS CloudFormation API.	April 4, 2024
<a href="#">AmplifyBackendDeployFullAccess</a> – Pembaruan ke kebijakan yang ada	<p>Tambahkan tindakan <code>lambda:InvokeFunction</code> 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.</p> <p>Tambahkan tindakan <code>lambda:UpdateFunctionCode</code> kebijakan untuk mendukung fungsi hotswapping.</p>	Januari 02, 2024
<a href="#">AmplifyBackendDeployFullAccess</a> – Pembaruan ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk mendukung <code>UpdateApiKey</code> operasi. Ini diperlukan untuk mengaktifkan penerapan aplikasi yang berhasil setelah keluar dan memulai ulang kotak pasir tanpa menghapus sumber daya.	17 November 2023

Perubahan	Deskripsi	Tanggal
<a href="#">AmplifyBackendDeployFullAccess</a> – Pembaruan ke kebijakan yang ada	Tambahkan <code>amplify:GetBackendEnvironment</code> izin untuk mendukung penerapan aplikasi Amplify.	6 November 2023
<a href="#">AmplifyBackendDeployFullAccess</a> – Kebijakan baru	Amplify menambahkan kebijakan baru dengan izin minimum yang diperlukan untuk menerapkan sumber daya backend Amplify.	Oktober 8, 2023
<a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada	Tambahkan <code>ecr:DescribeRepositories</code> izin yang diperlukan oleh Amplify Command Line Interface (CLI).	1 Juni 2023



Perubahan	Deskripsi	Tanggal
<p><a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada</p>	<p>Tambahkan tindakan kebijakan untuk mendukung penghapusan tag dari AWS AppSync sumber daya.</p> <p>Tambahkan tindakan kebijakan untuk mendukung sumber daya Amazon Polly.</p> <p>Tambahkan tindakan kebijakan untuk mendukung pembaruan konfigurasi OpenSearch domain.</p> <p>Tambahkan tindakan kebijakan untuk mendukung penghapusan tag dari AWS Identity and Access Management peran.</p> <p>Tambahkan tindakan kebijakan untuk mendukung penghapusan tag dari sumber daya Amazon DynamoDB.</p> <p>Tambahkan <code>cloudfront:GetCloudFrontOriginAccessIdentityConfig</code> izin <code>cloudfront:GetCloudFrontOriginAccessIdentity</code> dan ke blok <code>CLISDKCalls</code> pernyataan untuk mendukung Amplify publish dan hosting alur kerja.</p>	<p>Februari 24, 2023</p>

Perubahan	Deskripsi	Tanggal
	<p>Tambahkan <code>s3:PutBucketPublicAccessBlock</code> izin ke blok <code>CLIManageviaCFNPolicy</code> pernyataan untuk memungkinkan mendukung praktik terbaik keamanan Amazon S3 AWS CLI untuk mengaktifkan fitur Akses Publik Blok Amazon S3 di bucket internal.</p> <p>Tambahkan <code>cloudformation:DescribeStacks</code> izin ke blok <code>CLISDKCalls</code> pernyataan untuk mendukung pengambilan AWS CloudFormation tumpukan pelanggan saat mencoba ulang di prosesor backend Amplify untuk menghindari duplikasi eksekusi jika tumpukan diperbarui.</p> <p>Tambahkan <code>cloudformation:ListStacks</code> izin ke blok <code>CLICloudformationPolicy</code> pernyataan. Izin ini diperlukan untuk sepenuhnya mendukung CloudFormation <code>DescribeStacks</code> tindakan tersebut.</p>	

Perubahan	Deskripsi	Tanggal
<a href="#">AdministratorAccess-Amplify</a> - 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
<a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk memblokir akses publik ke bucket Amplify deployment Amazon S3.	27 April 2022
<a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada	<p>Tambahkan tindakan untuk memungkinkan pelanggan menghapus aplikasi yang dirender sisi server (SSR) mereka. Ini juga memungkinkan CloudFront distribusi yang sesuai untuk dihapus dengan sukses.</p> <p>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 <a href="#">UpdateEventSourceMapping</a> tindakan.</p>	April 17, 2022

Perubahan	Deskripsi	Tanggal
<a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada	Tambahkan tindakan kebijakan untuk mengaktifkan tindakan Amplify UI Builder di semua resource.	Desember 2, 2021
<a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada	<p>Tambahkan tindakan kebijakan untuk mendukung fitur autentikasi Amazon Cognito yang menggunakan penyedia identitas sosial.</p> <p>Tambahkan tindakan kebijakan untuk mendukung lapisan Lambda.</p> <p>Tambahkan tindakan kebijakan untuk mendukung kategori Amplify Storage.</p>	November 8, 2021

Perubahan	Deskripsi	Tanggal
<p><a href="#">AdministratorAccess-Amplify</a> - Perbarui ke kebijakan yang ada</p>	<p>Tambahkan tindakan Amazon Lex untuk mendukung kategori Amplify Interactions.</p> <p>Tambahkan tindakan Amazon Rekognition untuk mendukung kategori Amplify Predictions.</p> <p>Tambahkan tindakan Amazon Cognito untuk mendukung konfigurasi MFA di kumpulan pengguna Amazon Cognito.</p> <p>Tambahkan CloudFormation tindakan untuk mendukung AWS CloudFormation StackSets.</p> <p>Tambahkan tindakan Amazon Location Service untuk mendukung kategori Amplify Geo.</p> <p>Tambahkan tindakan Lambda untuk mendukung lapisan Lambda di Amplify.</p> <p>Tambahkan tindakan CloudWatch Log untuk mendukung CloudWatch Acara.</p> <p>Tambahkan tindakan Amazon S3 untuk mendukung kategori Amplify Storage.</p>	<p>27 September 2021</p>

Perubahan	Deskripsi	Tanggal
	Tambahkan tindakan kebijakan untuk mendukung aplikasi yang dirender sisi server (SSR).	

Perubahan	Deskripsi	Tanggal
<p><a href="#">AdministratorAccess-Amplify</a></p> <p>- Perbarui ke kebijakan yang ada</p>	<p>Konsolidasikan semua tindakan Amplify menjadi satu <code>amplify:*</code> tindakan.</p> <p>Tambahkan tindakan Amazon S3 untuk mendukung mengenkripsi bucket Amazon S3 pelanggan.</p> <p>Tambahkan tindakan batas izin IAM untuk mendukung Amplify aplikasi yang memiliki batas izin diaktifkan.</p> <p>Tambahkan tindakan Amazon SNS untuk mendukung melihat nomor telepon originasi, serta melihat, membuat, memverifikasi, dan menghapus nomor telepon tujuan.</p> <p>Amplify Studio: Tambahkan Amazon Cognito AWS Lambda, IAM, AWS CloudFormation dan tindakan kebijakan untuk mengaktifkan pengelolaan backend di konsol Amplify dan Amplify Studio.</p> <p>Tambahkan pernyataan kebijakan AWS Systems Manager (SSM) untuk mengelola rahasia lingkungan Amplify.</p>	<p>28 Juli 2021</p>

Perubahan	Deskripsi	Tanggal
	Tambahkan AWS CloudFormation <code>ListResources</code> 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 (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, lihat hal 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 penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

## Perlindungan Data di Amplify

AWS Amplify sesuai dengan [model tanggung jawab AWS bersama model tanggung jawab](#), 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 kredensial 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 sumber daya. AWS
- 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 SDK. 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 SSD yang dienkripsi untuk titik keberadaan lokasi tepi (POP), dan volume EBS terenkripsi untuk Regional Edge Cache (REC). Kode fungsi dan konfigurasi dalam CloudFront Fungsi selalu disimpan dalam format terenkripsi pada SSD terenkripsi pada POP lokasi tepi, dan 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 Private Certificate Authority. AWS Certificate Manager Untuk informasi selengkapnya, lihat [Proses penandatanganan Signature Version 4](#) dan [Tentang ACM PCA](#).

## 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:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.

- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

 Note

Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi 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 Manager](#)Ini 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 (Amazon EC2) dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya tentang penggunaan CloudWatch metrik dan alarm dengan Amplify, lihat. [Pemantauan](#)
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari instans Amazon EC2, 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 CloudTrail menangkap 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 memberikan aliran data real-time dari aplikasi Anda sendiri, aplikasi S oftware-as-a -Service (SaaS), dan AWS 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 lebih berhak 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 pengguna utama 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:service::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	<a href="#">Ya</a>	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	<a href="#">Ya</a>	Jumlah maksimum pekerjaan bersamaan yang dapat Anda buat di akun ini di Wilayah saat ini.
Domain per aplikasi	Setiap Wilayah yang didukung: 5	<a href="#">Ya</a>	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	<a href="#">Ya</a>	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	<a href="#">Ya</a>	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

# Pemecahan Masalah Amplify Hosting

Jika Anda mengalami kesalahan atau masalah penerapan saat bekerja dengan Amplify Hosting, lihat topik di bagian ini.

Topik

- [Memecahkan masalah Amplify umum](#)
- [Memecahkan masalah image build Amazon Linux 2023](#)
- [Memecahkan masalah domain kustom](#)
- [Memecahkan masalah aplikasi yang dirender sisi server](#)

## Memecahkan masalah Amplify umum

Informasi berikut dapat membantu Anda memecahkan masalah umum dengan Amplify Hosting.

Topik

- [Kode status HTTP 429 \(Terlalu banyak permintaan\)](#)

## 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 lebih lanjut tentang kuota, lihat [Kuota 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 header kontrol cache 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.

## Memecahkan masalah image build Amazon Linux 2023

Informasi berikut dapat membantu Anda memecahkan masalah dengan image build Amazon Linux 2023 (AL2023).

Topik

- [Bagaimana cara menjalankan fungsi Amplify dengan runtime Python](#)
- [Bagaimana cara menjalankan perintah yang memerlukan hak superuser atau root](#)

### Bagaimana cara 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 AL2023 memiliki symlink untuk versi Python yang lebih lama yang sudah diinstal sebelumnya. Oleh karena itu, Anda tidak perlu lagi memperbarui perintah build dalam spesifikasi build aplikasi menggunakan instruksi yang tersedia di FAQ [Amplify Hosting GitHub](#) .

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
```

## Bagaimana cara menjalankan perintah yang memerlukan 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 `install -y gcc`, gunakan `sudo yum install -y gcc`.

Gambar build Amazon Linux 2 menggunakan pengguna root, tetapi gambar AL2023 Amplify menjalankan kode Anda dengan pengguna khusus. 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 domain kustom

Jika Anda mengalami masalah saat menyambungkan domain kustom ke aplikasi Amplify, lihat bantuan [Memecahkan masalah domain kustom](#).

## Memecahkan masalah aplikasi yang dirender sisi server

Jika Anda mengalami masalah saat menerapkan aplikasi SSR ke Amplify, lihat bantuan. [Memecahkan masalah penerapan SSR](#)



# AWS Amplify Referensi hosting

Gunakan topik di bagian ini untuk menemukan materi referensi terperinci seputar AWS Amplify.

Topik

- [AWS CloudFormation Dukungan](#)
- [Dukungan AWS Command Line Interface](#)
- [Dukungan penandaan sumber daya](#)
- [Memperkuat Hosting API](#)

## AWS CloudFormation Dukungan

Gunakan AWS CloudFormation template untuk menyediakan sumber daya Amplify, memungkinkan penerapan aplikasi web berulang dan andal. AWS CloudFormation menyediakan bahasa umum bagi Anda untuk menggambarkan 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 [Memperkuat CloudFormation dokumentasi](#). Untuk Amplify Studio, lihat [Memperkuat UI Builder CloudFormation dokumentasi](#).

## Dukungan AWS Command Line Interface

Gunakan AWS Command Line Interface untuk membuat aplikasi Amplify secara terprogram dari baris perintah. Untuk informasi, lihat [AWS CLIdokumentasi](#).

## 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](#).

## Memperkuat Hosting API

Referensi ini memberikan deskripsi tindakan dan tipe data untuk Amplify Hosting API. Untuk informasi lebih lanjut, lihat [Memperkuat referensi API dokumentasi](#).

# Riwayat dokumen untuk AWS Amplify

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir AWS Amplify.

- Pembaruan dokumentasi terbaru: 31 Mei 2024

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	31 Mei 2024
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	April 17, 2024
Bab memulai yang diperbarui	Diperbarui <a href="#">Memulai dengan Amplify Hosting</a> chapter untuk menggunakan contoh aplikasi Next.js dalam tutorial.	April 12, 2024
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	April 5, 2024
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan	April 4, 2024

Perubahan	Deskripsi	Tanggal
	perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	
Bab Pemecahan Masalah Baru	Menambahkan <a href="#">Pemecahan Masalah Amplify Hosting</a> chapter untuk menjelaskan cara memperbaiki masalah yang Anda temui dengan aplikasi yang digunakan untuk Amplify Hosting.	April 2, 2024
Dukungan baru untuk sertifikat SSL/TLS kustom	Menambahkan <a href="#">Menggunakan sertifikat SSL/TLS</a> topik ke bagian untuk <a href="#">Menyiapkan domain khusus</a> menjelaskan dukungan Amplify untuk sertifikat SSL/TLS kustom saat menghubungkan aplikasi ke domain kustom.	Februari 20, 2024
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	Januari 2, 2024
Dukungan baru untuk kerangka kerja SSR	Menambahkan <a href="#">Amplify dukungan untuk kerangka kerja SSR</a> topik untuk menjelaskan dukungan Amplify untuk kerangka kerja SSR berbasis JavaScript apa pun dengan adaptor sumber terbuka.	November 19, 2023

Perubahan	Deskripsi	Tanggal
Dukungan baru untuk peluncuran fitur pengoptimalan gambar	Menambahkan <a href="#">Pengoptimalan gambar untuk aplikasi SSR</a> topik untuk menjelaskan dukungan bawaan untuk pengoptimalan gambar untuk aplikasi yang dirender sisi server.	November 19, 2023
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	17 November 2023
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	6 November 2023
Topik subdomain wildcard baru	Menambahkan <a href="#">Subdomain wildcard</a> topik untuk menjelaskan dukungan untuk subdomain wildcard pada domain kustom.	6 November 2023
Kebijakan terkelola baru	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan kebijakan AmplifyBackendDeployFullAccess AWS terkelola baru untuk Amplify.	Oktober 8, 2023

Perubahan	Deskripsi	Tanggal
Dukungan baru untuk peluncuran fitur kerangka kerja monorepo	Memperbarui <a href="#">Pengaturan build monorepo</a> topik untuk menjelaskan dukungan penerapan aplikasi di monorepos yang dibuat menggunakan ruang kerja npm, ruang kerja pnpm, ruang kerja Yarn, Nx, dan Turborepo .	19 Juni 2023
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	1 Juni 2023
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	Februari 24, 2023
Bab rendering sisi server yang diperbarui	Memperbarui <a href="#">Terapkan aplikasi yang dirender sisi server dengan Amplify Hosting</a> chapter untuk menjelaskan perubahan terbaru pada dukungan Amplify untuk Next.js versi 12 dan 13.	17 November 2022

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	30 Agustus 2022
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">Membangun backend untuk aplikasi</a> topik untuk menjelaskan cara menerapkan backend menggunakan Amplify Studio.	23 Agustus 2022
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	27 April 2022
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	April 17, 2022
Peluncuran fitur GitHub Aplikasi Baru	Menambahkan <a href="#">Menyiapkan akses Amplify ke GitHub repositori</a> topik untuk menjelaskan GitHub Aplikasi baru untuk mengotorisasi akses Amplify ke GitHub repositori Anda.	5 April 2022

Perubahan	Deskripsi	Tanggal
Peluncuran fitur Amplify Studio baru	Memperbarui <a href="#">Selamat datang di AWS Amplify Hosting</a> topik untuk menjelaskan pembaruan pada Amplify Studio yang menyediakan desainer visual untuk membuat komponen UI yang dapat Anda sambungkan ke data backend Anda.	Desember 2, 2021
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify guna mendukung Amplify Studio.	Desember 2, 2021
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	November 8, 2021
Topik kebijakan terkelola yang diperbarui	Memperbarui <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan perubahan terbaru pada kebijakan AWS terkelola untuk Amplify.	27 September 2021

Perubahan	Deskripsi	Tanggal
Topik kebijakan terkelola baru	Menambahkan <a href="#">AWS kebijakan terkelola untuk AWS Amplify</a> topik untuk menjelaskan kebijakan AWS terkelola untuk Amplify dan perubahan terbaru pada kebijakan tersebut.	28 Juli 2021
Pembaruan Bab rendering sisi server	Pembaruan bab <a href="#">Terapkan aplikasi yang dirender sisi server dengan Amplify Hosting</a> untuk menjelaskan dukungan baru untuk Next.js versi 10.x.x dan Next.js versi 11.	22 Juli 2021
Pembaruan Bab konfigurasi pengaturan build	Penambahan topik <a href="#">Pengaturan build monorepo</a> untuk menjelaskan cara mengonfigurasi pengaturan build dan variabel lingkungan <code>AMPLIFY_MONOREPO_APP_ROOT</code> baru ketika mendeploy aplikasi monorepo dengan Amplify.	20 Juli 2021



Perubahan	Deskripsi	Tanggal
Pembaruan Bab deployment cabang fitur	<p>Penambahan topik <a href="#">Pembuatan waktu pembuatan otomatis konfigurasi Amplify (hanya aplikasi Gen 1)</a> untuk menjelaskan cara menghasilkan secara otomatis file <code>aws-exports.js</code> pada waktu build. Penambahan topik <a href="#">Build backend bersyarat (hanya aplikasi Gen 1)</a> untuk menjelaskan cara mengaktifkan build backend bersyarat. Penambahan topik <a href="#">Gunakan backend Amplify di seluruh aplikasi (hanya aplikasi Gen 1)</a> 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.</p>	30 Juni 2021
Pembaruan Bab keamanan	<p>Penambahan topik <a href="#">Perlindungan Data di Amplify</a> untuk menjelaskan cara menerapkan model tanggung jawab bersama dan cara Amplify menggunakan enkripsi untuk melindungi data at rest dan in transit.</p>	3 Juni 2021

Perubahan	Deskripsi	Tanggal
Dukungan baru untuk peluncuran fitur SSR	Penambahan bab <a href="#">Terapkan aplikasi yang dirender sisi server dengan Amplify Hosting</a> 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 <a href="#">Keamanan di Amplify</a> 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 <a href="#">Gambar build kustom dan pembaruan paket langsung</a> untuk menjelaskan cara mengonfigurasi gambar build kustom yang di-hosting di Amazon Elastic Container Registry Public.	12 Maret 2021
Pembaruan topik pemantauan	Memperbarui topik <a href="#">Pemantauan</a> untuk menjelaskan cara mengakses data CloudWatch metrik Amazon dan menyetel alarm.	2 Februari 2021

Perubahan	Deskripsi	Tanggal
Topik CloudTrail logging baru	Menambahkan <a href="#">panggilan Logging Amplify API menggunakan AWS CloudTrail</a> topik untuk menjelaskan cara AWS CloudTrail menangkap dan mencatat semua tindakan API untuk Referensi API AWS Amplify Konsol dan Referensi API UI AWS Amplify Admin.	2 Februari 2021
Peluncuran fitur UI Admin baru	Pembaruan topik <a href="#">Selamat datang di AWS Amplify Hosting</a> 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 <a href="#">Mengelola kinerja aplikasi</a> topik untuk menjelaskan cara mengaktifkan mode kinerja guna mengoptimalkan kinerja hosting yang lebih cepat.	4 November 2020
Pembaruan topik header kustom	Pembaruan topik <a href="#">Header kustom</a> 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 <a href="#">Mengatur subdomain otomatis untuk domain kustom Route 53</a> untuk menjelaskan cara menggunakan deployment cabang fitur berbasis pola untuk aplikasi yang terhubung ke domain kustom Amazon Route 53. Penambahan topik <a href="#">Akses pratinjau web dengan subdomain</a> untuk menjelaskan cara mengatur pratinjau web dari permintaan tarik agar dapat diakses dengan subdomain.	20 Juni 2020
Topik notifikasi baru	Penambahan topik <a href="#">Notifikasi</a> 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	Memperbarui <a href="#">Menyiapkan domain khusus</a> 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 melepaskan	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.