Manual do usuário

AWS Amplify Hospedagem

AWS Amplify Hospedagem: Manual do usuário

Table of Contents

Bem-vindo à AWS Amplify hospedagem

O Amplify Hosting fornece um fluxo de trabalho baseado em git para hospedar aplicações full-stack da Web com tecnologia sem servidor com implantação contínua. O Amplify implanta seu aplicativo na rede AWS global de entrega de conteúdo (CDN). Este guia do usuário fornece as informações de que você precisa para começar a usar o Amplify Hosting.

Frameworks compatíveis

O Amplify Hosting oferece suporte a várias estruturas SSR comuns, estruturas de aplicações de página única (SPA) e geradores de sites estáticos, incluindo os à seguir.

Frameworks SSR

- Next.js
- Nuxt
- Astro com um adaptador comunitário
- SvelteKit com um adaptador comunitário
- Qualquer framework de SSR com um adaptador personalizado

Frameworks SPA

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

Geradores de sites estáticos

- Eleventy
- Gatsby
- Hugo
- Jekyll

- VuePress

Atributos do Amplify Hosting

Ramificações de atributos

Gerencie ambientes de preparação e produção para o frontend e backend conectando novas ramificações.

Domínios personalizados

Conecte seu aplicativo a um domínio personalizado.

Pré-visualizações de solicitação pull

Pré-visualize as alterações durante as análises de código.

End-to-end testando

Melhore a qualidade do seu aplicativo com end-to-end testes.

Ramificações protegidas por senha

Proteja o aplicativo web com senha para poder trabalhar em novos atributos sem torná-los acessíveis publicamente.

Redirecionamentos e regravações

Configure regravações e redirecionamentos para manter as classificações de SEO e direcionar o tráfego com base nos requisitos do seu aplicativo cliente.

Implantações atômicas

As implantações atômicas eliminam janelas de manutenção, garantindo que a aplicação da Web seja atualizada quando toda a implantação se encerrar. Isso elimina cenários em que não é possível fazer upload dos arquivos corretamente.

Conceitos básicos do Amplify Hosting

Para começar a usar o Amplify Hosting, consulte o tutorial Noções básicas da implantação de uma aplicação no Amplify Hosting. Depois de concluir o tutorial, você saberá como conectar um aplicativo web em um repositório Git (GitHub,, BitBucket GitLab, ou AWS CodeCommit) e implantá-lo no Amplify Hosting com implantação contínua.

Compilação de um backend

O AWS Amplify Gen 2 apresenta uma experiência de desenvolvedor TypeScript baseada em código para definir back-ends. Para saber como usar o Amplify Gen 2 para compilar e conectar um backend à sua aplicação, consulte Compilação e criação de backends nos documentos do Amplify.

Para entender melhor a abordagem do Amplify que prioriza o código do Gen 2, consulte o Workshop do Amplify Gen 2 no site do Workshop Studio da AWS . Neste tutorial abrangente, você cria uma aplicação com tecnologia sem servidor com React e Next.js e aprende a usar as bibliotecas de dados e autenticação do Amplify Gen 2 e a biblioteca Amplify UI para adicionar funcionalidade à aplicação.

Se você estiver buscando pela documentação para compilar backends para uma aplicação Gen 1 usando a CLI e o Amplify Studio, consulte Compilação e conexão de backends nos documentos do Amplify Gen 1.

Definição de preços do Amplify Hosting

AWS Amplify só cobra pelo que você usa. Para obter mais informações, consulte AWS Amplify Preço.

Noções básicas da implantação de uma aplicação no Amplify Hosting

Para ajudar você a entender como o Amplify Hosting funciona, os tutoriais a seguir orientam você na criação e implantação de aplicações criadas usando frameworks SSR comuns com suporte no Amplify.

Tutoriais

- Inscreva-se para um Conta da AWS
- Implemente um Next.js aplicativo no Amplify Hosting
- Implemente um Nuxt.js aplicativo no Amplify Hosting
- Implante um Astro.js aplicativo no Amplify Hosting
- Implemente um SvelteKit aplicativo no Amplify Hosting

Inscreva-se para um Conta da AWS

Para começar AWS, você precisa de um Conta da AWS. Para obter informações sobre como criar um Conta da AWS, consulte Introdução a um Conta da AWS no Guia de AWS Gerenciamento de contas referência.

Implemente um Next.js aplicativo no Amplify Hosting

Este tutorial explica como criar e implantar um Next.js aplicativo a partir de um repositório Git.

Antes de começar este tutorial, conclua os pré-requisitos a seguir.

Criar uma aplicação.

Crie um Next.js aplicativo básico para usar neste tutorial, usando as instruções create-next-app na documentação. Next.js

Crie um repositório Git

O Amplify suporta GitHub Bitbucket e. GitLab AWS CodeCommit Envie sua aplicação `create-next-app` para seu repositório Git.

Etapa 1: conectar um repositório Git

Nesta etapa, você conecta seu Next.js aplicativo em um repositório Git ao Amplify Hosting.

Para conectar uma aplicação a um repositório Git

1. Abra o console do Amplify.
2. Se você estiver implantando sua primeira aplicação na região atual, por padrão, você começará na página de serviço do AWS Amplify.

   Na parte superior da página, escolha Implantar uma aplicação.
3. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.

   Para GitHub repositórios, o Amplify usa GitHub o recurso Apps para autorizar o acesso ao Amplify. Para obter mais informações sobre como instalar e autorizar o GitHub aplicativo, consulteConfigurando o acesso do Amplify aos repositórios GitHub.

   > **Note**
   >
   > Depois de autorizar o console do Amplify com o Bitbucket GitLab, ou AWS CodeCommit, o Amplify busca um token de acesso do provedor do repositório, mas não armazena o token nos servidores. AWS O Amplify acessa seu repositório usando chaves de implantação instaladas somente em um repositório específico.
4. Na página Adicionar ramificação do repositório, faça o seguinte:
   a. Selecione o nome do repositório a ser conectado.
   b. Selecione o nome da ramificação do repositório a ser conectada.
   c. Escolha Próximo.

Etapa 2: confirmar as configurações de compilação

O Amplify detecta automaticamente a sequência de comandos de compilação a ser executada na ramificação que você está implantando. Nesta etapa, você revisa e confirma suas configurações de compilação.

Para confirmar as configurações de compilação de uma aplicação

1. Na página de Configurações da aplicação, localize a seção Configurações de compilação.

   Verifique se o Comando de compilação do frontend e o Diretório de saída da compilação estão corretos. Para este aplicativo de Next.js exemplo, o diretório de saída do Build está definido como `.next`.
2. O procedimento para adicionar um perfil de serviço varia dependendo se você deseja criar um novo perfil ou usar um existente.
   - Para criar um novo perfil:
     - Escolha Criar e usar um novo perfil de serviço.
   - Para usar um perfil existente:
     a. Escolha Usar um perfil existente.
     b. Na lista de perfis de serviço, selecione o perfil a ser usado.
3. Escolha Próximo.

Etapa 3: Implantar um aplicativo

Nesta etapa, você implanta seu aplicativo na rede AWS global de entrega de conteúdo (CDN).

Para salvar e implantar uma aplicação

1. Na página Revisão, confirme se os detalhes do repositório e as configurações da aplicação estão corretos.
2. Escolha Salvar e implantar. Sua compilação do frontend geralmente leva de 1 a 2 minutos, mas pode variar de acordo com o tamanho da aplicação.
3. Depois de concluir a implantação, sua aplicação poderá ser visualizada por meio do link para o domínio padrão `amplifyapp.com`.

> **Note**
>
> Para aumentar a segurança de seus aplicativos do Amplify, o domínio amplifyapp.com é registrado na Lista Pública de Sufixos (PSL). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para seus aplicativos do Amplify. Essa prática ajudará a defender seu

domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a Set-Cookiepágina na Rede de Desenvolvedores da Mozilla.

Etapa 4: (opcional) limpar recursos

Se você não precisa mais da aplicação que implantou no tutorial, é possível excluí-la. Esta etapa ajuda a garantir que você não será cobrado pelos recursos que não está utilizando.

Para excluir uma aplicação

1. No menu Configurações da aplicação, no painel de navegação, escolha Configurações gerais.
2. Na página Configurações gerais, escolha Excluir aplicação.
3. Na janela de confirmação, insira **`delete`**. Em seguida, selecione Excluir aplicação.

Adicionar recursos à sua aplicação

Agora que você tem uma aplicação implantada no Amplify, será possível explorar alguns dos recursos a seguir que estão disponíveis para sua aplicação hospedada.

Variáveis de ambiente

As aplicações geralmente precisam de informações de configuração no runtime. Essas configurações podem ser detalhes da conexão do banco de dados, chaves de API ou parâmetros. As variáveis de ambiente fornecem uma forma de expor essas configurações no momento da compilação. Para obter mais informações, consulte Environment variables.

Domínios personalizados

Neste tutorial, o Amplify hospeda sua aplicação para você no domínio padrão `amplifyapp.com` com uma URL como `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`. Quando você conecta seu aplicativo a um domínio personalizado, os usuários veem que seu aplicativo está hospedado em um URL personalizado, como `https://www.example.com`. Para obter mais informações, consulte Configuração de domínios personalizado.

Pré-visualizações de solicitação pull

As pré-visualizações de solicitação pull da Web oferecem às equipes uma maneira de visualizar as alterações das solicitações pull (PRs) antes de mesclar o código a uma ramificação de

produção ou integração. Para obter mais informações, consulte Pré-visualizações da Web para solicitações pull.

Gerenciar vários ambientes

Para saber como o Amplify funciona com ramificações de recursos e GitFlow fluxos de trabalho para oferecer suporte a várias implantações, consulte Implantações de ramificações de recursos e fluxos de trabalho de equipe.

Implemente um Nuxt.js aplicativo no Amplify Hosting

Use as instruções a seguir para implantar um Nuxt.js aplicativo no Amplify Hosting. O Nuxt implementou um adaptador predefinido usando o servidor Nitro. Isso permite que você implante um projeto Nuxt sem nenhuma configuração adicional.

Para implantar uma aplicação Nuxt no Amplify Hosting

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todas as aplicações, escolha Criar nova aplicação.
3. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
4. Na página Adicionar ramificação do repositório, faça o seguinte:
   a. Selecione o nome do repositório a ser conectado.
   b. Selecione o nome da ramificação do repositório a ser conectada.
   c. Escolha Próximo.
5. Se você quiser que o Amplify seja capaz de entregar registros de aplicativos para o Amazon CloudWatch Logs, você deve habilitar isso explicitamente no console. Abra a seção Configurações avançadas e escolha Ativar registros do aplicativo SSR na seção Implantação de Server-Side renderização (SSR).
6. Escolha Próximo.
7. Na página Revisar, escolha Salvar e implantar.

Implante um Astro.js aplicativo no Amplify Hosting

Use as instruções a seguir para implantar um Astro.js aplicativo no Amplify Hosting. É possível usar uma aplicação existente ou criar uma aplicação inicial usando um dos exemplos oficiais fornecidos pelo Astro. Para criar uma aplicação inicial, consulte Uso de um tema ou modelo inicial na Documentação do Astro.

Para implantar um site do Astro com SSR na Amplify Hosting, é necessário adicionar um adaptador à sua aplicação. Não mantemos um adaptador de propriedade da Amplify para o framework do Astro. Este tutorial usa o adaptador `astro-aws-amplify`, que foi criado por um membro da comunidade. Esse adaptador está disponível no github. com/alexnguyennz/astro-aws-amplify no site. GitHub AWS não mantém esse adaptador.

Para implantar uma aplicação Astro no Amplify Hosting

1. No seu computador local, navegue até a aplicação Astro a ser implantada.
2. Para instalar o adaptador, abra uma janela de terminal e execute o comando a seguir. Este exemplo usa o adaptador de comunidade disponível no github. com/alexnguyennz/astro-aws-amplify. Você pode *`astro-aws-amplify`* substituir pelo nome do adaptador que você está usando.

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

3. Na pasta do projeto da sua aplicação Astro, abra o arquivo `astro.config.mjs`. Atualize o arquivo para adicionar o adaptador. O arquivo deve ser semelhante ao seguinte.

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

   import sitemap from '@astrojs/sitemap';

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

4. Confirme a alteração e envie o projeto para seu repositório Git.

   Agora você está pronto para implantar sua aplicação do Astro no Amplify.

5. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
6. Na página Todas as aplicações, escolha Criar nova aplicação.
7. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
8. Na página Adicionar ramificação do repositório, faça o seguinte:

   a. Selecione o nome do repositório a ser conectado.

   b. Selecione o nome da ramificação do repositório a ser conectada.

   c. Escolha Próximo.

9. Na página de Configurações da aplicação, localize a seção Configurações de compilação. Em Diretório de saída da compilação, insira **`.amplify-hosting`**.
10. Você também deve atualizar os comandos de criação de frontend da aplicação na especificação de compilação. Para abrir a especificação de compilação, escolha Editar arquivo YML.
11. No arquivo `amplify.yml`, localize a seção de comandos de compilação de frontend. Digite **`mv node_modules ./.amplify-hosting/compute/default`**

    Seu arquivo de configurações de compilação do projeto devem se parecer com o seguinte.

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

12. Escolha Salvar.
13. Se você quiser que o Amplify seja capaz de entregar registros de aplicativos para o Amazon CloudWatch Logs, você deve habilitar isso explicitamente no console. Abra a seção Configurações avançadas e escolha Ativar registros do aplicativo SSR na seção Implantação de Server-Side renderização (SSR).
14. Escolha Próximo.
15. Na página Revisar, escolha Salvar e implantar.

Implemente um SvelteKit aplicativo no Amplify Hosting

Use as instruções a seguir para implantar um SvelteKit aplicativo no Amplify Hosting. É possível usar sua própria aplicação ou criar uma aplicação inicial. Para obter mais informações, consulte Criação de um projeto na SvelteKit documentação.

Para implantar um SvelteKit aplicativo com SSR no Amplify Hosting, você deve adicionar um adaptador ao seu projeto. Não mantemos um adaptador de propriedade da Amplify para a SvelteKit estrutura. Neste exemplo, estamos usando o `amplify-adapter` criado por um membro da comunidade. O adaptador está disponível no github. com/gzimbron/amplify-adapter no site. GitHub AWS não mantém esse adaptador.

Para implantar um SvelteKit aplicativo no Amplify Hosting

1. No seu computador local, navegue até o SvelteKit aplicativo a ser implantado.
2. Para instalar o adaptador, abra uma janela de terminal e execute o comando a seguir. Este exemplo usa o adaptador de comunidade disponível no github. com/gzimbron/amplify-adapter. Se você estiver usando um adaptador de comunidade diferente, *`amplify-adapter`* substitua pelo nome do seu adaptador.

```
npm install amplify-adapter
```

3. Na pasta do projeto do seu SvelteKit aplicativo, abra o `svelte.config.js` arquivo. Edite o arquivo para usar `amplify-adapter` ou *`'amplify-adapter'`* substituir pelo nome do seu adaptador. O arquivo deve ser semelhante ao seguinte.

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

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

        kit: {
                // adapter-auto only supports some environments, see https://
kit.svelte.dev/docs/adapter-auto for a list.
                // If your environment is not supported, or you settled on a
 specific environment, switch out the adapter.
                // See https://kit.svelte.dev/docs/adapters for more information
 about adapters.
                adapter: adapter()
        }
};

export default config;
```

4. Confirme a alteração e envie a aplicação para seu repositório Git.
5. Agora você está pronto para implantar seu SvelteKit aplicativo no Amplify.

   Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
6. Na página Todas as aplicações, escolha Criar nova aplicação.
7. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
8. Na página Adicionar ramificação do repositório, faça o seguinte:

   a. Selecione o nome do repositório a ser conectado.

   b. Selecione o nome da ramificação do repositório a ser conectada.

   c. Escolha Próximo.
9. Na página de Configurações da aplicação, localize a seção Configurações de compilação. Em Diretório de saída da compilação, insira **`build`**.
10. Você também deve atualizar os comandos de criação de frontend da aplicação na especificação de compilação. Para abrir a especificação de compilação, escolha Editar arquivo YML.
11. No arquivo `amplify.yml`, localize a seção de comandos de compilação de frontend. Insira **`- cd build/compute/default/`** e **`- npm i --production`**.

Seu arquivo de configurações de compilação do projeto devem se parecer com o seguinte.

```
version: 1
frontend:
    phases:
        preBuild:
            commands:
                - 'npm ci --cache .npm --prefer-offline'
        build:
            commands:
                - 'npm run build'
                - 'cd build/compute/default/'
                - 'npm i --production'

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

12. Escolha Salvar.
13. Se você quiser que o Amplify seja capaz de entregar registros de aplicativos para o Amazon CloudWatch Logs, você deve habilitar isso explicitamente no console. Abra a seção Configurações avançadas e escolha Ativar registros do aplicativo SSR na seção Implantação de Server-Side renderização (SSR).
14. Escolha Próximo.
15. Na página Revisar, escolha Salvar e implantar.

Implantação de aplicações renderizadas do lado do servidor com o Amplify Hosting

Você pode usar AWS Amplify para implantar e hospedar aplicativos web que usam renderização do lado do servidor (SSR). O Amplify Hosting detecta automaticamente os aplicativos criados usando a Next.js estrutura e você não precisa realizar nenhuma configuração manual no. Console de gerenciamento da AWS

O Amplify também oferece suporte a qualquer framework SSR baseado em Javascript com um adaptador de compilação de código aberto que transforme a saída de compilação de uma aplicação na estrutura de diretórios esperada pelo Amplify Hosting. Por exemplo, você pode implantar aplicativos criados com o Nuxt, o Astro e as SvelteKit estruturas instalando os adaptadores disponíveis.

Usuários avançados podem usar a especificação de implantação para criar um adaptador de compilação ou configurar um script de pós-compilação.

É possível implantar as seguintes estruturas no Amplify Hosting com configuração mínima.

Next.js

- O Amplify suporta Next.js 15 aplicativos sem a necessidade de um adaptador. Para começar, consulte o Amplifique o suporte para Next.js.

Nuxt.js

- O Amplify suporta implantações Nuxt.js de aplicativos com um adaptador predefinido. Para começar, consulte o Amplifique o suporte para Nuxt.js.

Astro.js

- O Amplify oferece suporte a implantações de Astro.js aplicativos com um adaptador comunitário. Para começar, consulte o Amplifique o suporte para Astro.js.

SvelteKit

- O Amplify oferece suporte a implantações de SvelteKit aplicativos com um adaptador comunitário. Para começar, consulte o Amplifique o suporte para SvelteKit.

Adaptadores de código aberto

- Usar um adaptador de código aberto: para obter instruções sobre como usar qualquer adaptador que não esteja na lista anterior, consulte Uso de adaptadores de código aberto para qualquer estrutura SSR.

- Desenvolver um adaptador de framework: os criadores de frameworks que desejem integrar os recursos fornecidos por um framework podem usar a especificação de implantação do Amplify Hosting para configurar a saída da compilação de acordo com a estrutura esperada pelo Amplify. Para obter mais informações, consulte Uso da especificação de implantação do Amplify Hosting para configurar a saída da compilação.
- Configurar um script pós-compilação: é possível usar a especificação de implantação do Amplify Hosting para manipular sua saída de compilação conforme necessário para cenários específicos. Para obter mais informações, consulte Uso da especificação de implantação do Amplify Hosting para configurar a saída da compilação. Para ver um exemplo, consulte Como implantar um servidor Express usando o manifesto de implantação.

Tópicos

Amplifique o suporte para Next.js

O Amplify oferece suporte à implantação e hospedagem de aplicativos web renderizados do lado do servidor (SSR) criados usando. Next.js Next.js é uma estrutura React para desenvolver SPAs com JavaScript. Você pode implantar aplicativos criados com Next.js versões de até Next.js 15, com recursos como otimização de imagem e middleware.

Os desenvolvedores podem usar Next.js para combinar geração estática de sites (SSG) e SSR em um único projeto. As páginas SSG são pré-renderizadas no momento da compilação e as páginas SSR são pré-renderizadas no momento da solicitação.

A pré-renderização pode melhorar o desempenho e a otimização de mecanismos de pesquisa. Como Next.js pré-renderiza todas as páginas no servidor, o conteúdo HTML de cada página está pronto quando chega ao navegador do cliente. Esse conteúdo também pode ser carregado mais

rápido. Tempos de carregamento mais rápidos melhoram a experiência do usuário final com um site e impactam positivamente a classificação de SEO do site. A pré-renderização também melhora a SEO, permitindo que os bots dos mecanismos de pesquisa encontrem e rastreiem facilmente o conteúdo HTML de um site.

Next.js fornece suporte analítico integrado para medir várias métricas de desempenho, como Time to first byte (TTFB) e First contentful paint (FCP). Para obter mais informações sobre Next.js, consulte Introdução ao Next.js site.

Next.js suporte de recursos

A computação do Amplify Hosting gerencia totalmente a renderização do lado do servidor (SSR) para aplicativos criados com as versões 12 a 15. Next.js

Se você implantou um Next.js aplicativo no Amplify antes do lançamento da computação do Amplify Hosting em novembro de 2022, seu aplicativo está usando o provedor SSR anterior do Amplify, o Classic (somente 11). Next.js A computação do Amplify Hosting não é compatível com aplicativos criados usando a Next.js versão 11 ou anterior. É altamente recomendável que você migre seus Next.js 11 aplicativos para o provedor de SSR gerenciado por computação do Amplify Hosting.

A lista a seguir descreve os atributos específicos que o provedor de SSR de computação do Amplify Hosting suporta.

Recursos compatíveis

- Server-side páginas renderizadas (SSR)
- Páginas estáticas
- Rotas de API
- Rotas dinâmicas
- Detecção de todas as rotas
- SSG (geração estática)
- Regeneração estática incremental (ISR)
- Roteamento de subcaminhos internacionalizado (i18n)
- Roteamento de domínio internacionalizado (i18n)
- Detecção automática de localidade internacionalizada (i18n)
- Middleware
- Variáveis de ambiente

- Otimização de imagem
- Next.js 13 diretório de aplicativos

Atributos não compatíveis

- Rotas de API do Edge (o middleware Edge não é suportado)
- On-DemandRegeneração estática incremental (ISR)
- Next.js streaming
- Execução de middleware em ativos estáticos e imagens otimizadas
- Executando código após uma resposta com `unstable_after` (recurso experimental lançado com Next.js 15)

Next.js imagens

O tamanho máximo de saída de uma imagem não pode exceder 4,3 MB. Você pode ter um arquivo de imagem maior armazenado em algum lugar e usar o componente Next.js Imagem para redimensioná-lo e otimizá-lo em um formato Webp ou AVIF e depois exibi-lo em um tamanho menor.

Observe que a Next.js documentação recomenda que você instale o módulo de processamento de imagem da Sharp para permitir que a otimização da imagem funcione corretamente na produção. Porém, isso não é necessário para implantações do Amplify. O Amplify implanta automaticamente o Sharp para você.

Implantando um aplicativo Next.js SSR no Amplify

Por padrão, o Amplify implanta novos aplicativos SSR usando o serviço de computação da Amplify Hosting com suporte para as versões 12 a 15. Next.js A computação do Amplify Hosting gerencia integralmente os recursos necessários para implantar uma aplicação de SSR. Os aplicativos SSR em sua conta do Amplify que você implantou antes de 17 de novembro de 2022 estão usando o provedor SSR Classic Next.js (somente 11).

É altamente recomendável que você migre aplicativos usando SSR clássico (somente Next.js 11) para o provedor de SSR computacional Amplify Hosting. O Amplify não realiza migrações automáticas para você. É necessário migrar manualmente seu aplicativo e, em seguida, iniciar uma nova compilação para concluir a atualização. Para instruções, consulte Migrando um aplicativo de Next.js 11 SSR para a computação do Amplify Hosting.

Use as instruções a seguir para implantar um novo aplicativo Next.js SSR.

Para implantar um aplicativo SSR no Amplify usando o provedor de SSR de computação do Amplify Hosting

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todas as aplicações, escolha Criar nova aplicação.
3. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
4. Na página Adicionar ramificação do repositório, faça o seguinte:
   a. Na lista Repositórios atualizados recentemente, selecione o nome do repositório a ser conectado.
   b. Na lista Ramificação, selecione o nome da ramificação do repositório a ser conectada.
   c. Escolha Próximo.
5. O aplicativo requer um perfil de serviço IAM que o Amplify assume ao chamar outros serviços em seu nome. É possível permitir que a computação do Amplify Hosting crie automaticamente um perfil de serviço ou especificar um perfil que criou.
   - Para permitir que o Amplify crie automaticamente um perfil e o anexe à sua aplicação:
     - Escolha Criar e usar um novo perfil de serviço.
   - Para anexar um perfil de serviço que você criou anteriormente:
     a. Selecione Usar um perfil de serviço existente.
     b. Selecione o perfil a ser usado na lista.
6. Escolha Próximo.
7. Na página Revisar, escolha Salvar e implantar.

Package.json configurações de arquivo

Quando você implanta um Next.js aplicativo, o Amplify inspeciona o script de criação do aplicativo no `package.json` arquivo para determinar o tipo de aplicativo.

Veja a seguir um exemplo do script de criação de um Next.js aplicativo. O script de compilação `"next build"` indica que o aplicativo é compatível com páginas SSG e SSR. Esse script de construção também é usado para Next.js 14 ou mais aplicativos somente SSG.

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

Veja a seguir um exemplo do script de criação de um aplicativo SSG de Next.js 13 anos ou anterior. O script de compilação `"next build && next export"` indica que o aplicativo é compatível somente com páginas SSG.

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

Amplifique as configurações de compilação para um aplicativo SSR Next.js

Depois de inspecionar o arquivo `package.json` da sua aplicação, o Amplify verifica as configurações da compilação da aplicação. É possível salvar as configurações da compilação no console do Amplify ou em um arquivo `amplify.yml` na raiz do seu repositório. Para obter mais informações, consulte Definição das configurações de compilação de uma aplicação do Amplify.

Se o Amplify detectar que você está implantando um aplicativo Next.js SSR e nenhum `amplify.yml` arquivo estiver presente, ele gerará uma especificação de construção para o aplicativo e definirá como. `baseDirectory .next` Se você estiver implantando um aplicativo em que um arquivo `amplify.yml` esteja presente, as configurações da compilação no arquivo substituirão todas as configurações da compilação no console. Portanto, é necessário definir manualmente o valor `baseDirectory` para `.next` no arquivo.

Veja a seguir um exemplo das configurações da compilação de um aplicativo em que `baseDirectory` está definido como `.next`. Isso indica que os artefatos de construção são de um Next.js aplicativo compatível com páginas SSG e SSR.

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

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

Amplifique as configurações de compilação para um aplicativo SSG de Next.js 13 anos ou anterior

Se o Amplify detectar que você está implantando um aplicativo SSG Next.js 13 ou anterior, ele gera uma especificação de compilação para o aplicativo e define como. `baseDirectory` `out` Se você estiver implantando um aplicativo em que um arquivo `amplify.yml` está presente, deverá definir manualmente o valor `baseDirectory` para `out` no arquivo. O `out` diretório é a pasta padrão Next.js criada para armazenar ativos estáticos exportados. Ao definir as configurações de especificação de compilação da sua aplicação, altere o nome da pasta `baseDirectory` para corresponder à configuração da sua aplicação.

Veja a seguir um exemplo das configurações de compilação de um aplicativo em que `baseDirectory` está definido `out` para indicar que os artefatos de compilação são de um aplicativo de Next.js 13 anos ou anterior que oferece suporte somente a páginas SSG.

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

```
    paths:
      - node_modules/**/*
```

Amplifique as configurações de compilação para um aplicativo Next.js SSG 14 ou posterior

Na Next.js versão 14, o `next export` comando foi descontinuado e substituído pelo `output: 'export'` no `next.config.js` arquivo para permitir exportações estáticas. Se você estiver implantando um aplicativo somente de Next.js 14 SSG no console, o Amplify gera uma especificação de construção para o aplicativo e define como. `baseDirectory .next` Se você estiver implantando um aplicativo em que um arquivo `amplify.yml` está presente, deverá definir manualmente o valor `baseDirectory` para `.next` no arquivo. Essa é a mesma `baseDirectory` configuração que o Amplify usa para Next.js `WEB_COMPUTE` aplicativos que suportam páginas SSG e SSR.

A seguir está um exemplo das configurações de compilação para um aplicativo somente Next.js 14 SSG com o `baseDirectory` definido como. `.next`

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

Migrando um aplicativo de Next.js 11 SSR para a computação do Amplify Hosting

Quando você implanta um novo Next.js aplicativo, por padrão, o Amplify usa a versão mais recente compatível do. Next.js Atualmente, o provedor SSR de computação Amplify Hosting oferece suporte à versão 15. Next.js

O console do Amplify detecta aplicativos em sua conta que foram implantados antes da versão de novembro de 2022 do serviço de computação Amplify Hosting com suporte total para as versões 12 a 15. Next.js O console exibe um banner de informações identificando aplicativos com ramificações que são implantadas usando o provedor SSR anterior da Amplify, o Classic (somente Next.js 11). É altamente recomendável que você migre seus aplicativos para o provedor de SSR de computação do Amplify Hosting.

Se você estiver atualizando seu aplicativo Next.js 11 hospedado para Next.js 12 ou posterior, poderá receber um `"target" property is no longer supported` erro quando uma implantação for acionada. Nesse caso, será necessário migrar para a computação do Amplify Hosting.

É necessário migrar manualmente o aplicativo e todas as suas ramificações de produção ao mesmo tempo. Um aplicativo não pode conter ramificações Classic (somente Next.js 11) e Next.js 12 ou posteriores.

Use as instruções a seguir para migrar um aplicativo para o provedor de SSR de computação do Amplify Hosting.

Para migrar um aplicativo para o provedor de SSR de computação do Amplify Hosting

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o Next.js aplicativo que você deseja migrar.

   > **Note**
   >
   > Antes de migrar um aplicativo no console do Amplify, você deve primeiro atualizar o arquivo package.json do aplicativo para usar a versão 12 ou posterior. Next.js

3. No painel de navegação, em Configurações do aplicativo, selecione Geral.
4. Na página inicial do aplicativo, o console exibirá um banner se o aplicativo tiver ramificações implantadas usando o provedor SSR Classic (somente Next.js 11). No banner, escolha Migrar.
5. Na janela de confirmação da migração, selecione as três instruções e escolha Migrar.
6. O Amplify criará e reimplantará seu aplicativo para concluir a migração.

Revertendo uma migração de SSR

Quando você implanta um Next.js aplicativo, o Amplify Hosting detecta as configurações em seu aplicativo e define o valor interno da plataforma para o aplicativo. Há três valores válidos de

plataforma. Um aplicativo SSG é definido com o valor `WEB` da plataforma. Um aplicativo SSR usando a Next.js versão 11 é definido com o valor `WEB_DYNAMIC` da plataforma. Um aplicativo SSR de Next.js 12 ou posterior é definido com o valor `WEB_COMPUTE` da plataforma.

Quando você migra um aplicativo usando as instruções na seção anterior, o Amplify altera o valor da plataforma do seu aplicativo de `WEB_DYNAMIC` para `WEB_COMPUTE`. Após a conclusão da migração para a computação do Amplify Hosting, você não pode reverter a migração no console. Para reverter a migração, é necessário usar o AWS Command Line Interface para alterar a plataforma do aplicativo de volta para o `WEB_DYNAMIC`. Abra uma janela do terminal e digite o comando a seguir, atualizando o ID do aplicativo e a região com suas informações exclusivas.

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

Adicionando a funcionalidade SSR a um aplicativo estático Next.js

Você pode adicionar a funcionalidade SSR a um Next.js aplicativo estático (SSG) existente implantado com o Amplify. Antes de iniciar o processo de conversão do aplicativo SSG em SSR, atualize o aplicativo para usar a Next.js versão 12 ou posterior e adicione a funcionalidade SSR. Em seguida, você precisará executar as etapas a seguir.

1. Use o AWS Command Line Interface para alterar o tipo de plataforma do aplicativo.
2. Adicione um perfil de serviço ao aplicativo.
3. Atualize o diretório de saída nas configurações da compilação do aplicativo.
4. Atualize o `package.json` arquivo do aplicativo para indicar que o aplicativo usa SSR.

Atualização da plataforma

Há três valores válidos de plataforma. Um aplicativo SSG está configurado para o tipo de plataforma `WEB`. Um aplicativo SSR usando a Next.js versão 11 está configurado para o tipo de `WEB_DYNAMIC` plataforma. Para aplicativos implantados em Next.js 12 ou posterior usando SSR gerenciado pela computação do Amplify Hosting, o tipo de plataforma está definido como. `WEB_COMPUTE`

Quando você implantou seu aplicativo como um aplicativo SSG, o Amplify definiu o tipo de plataforma como `WEB`. Use o AWS CLI para alterar a plataforma do seu aplicativo para`WEB_COMPUTE`. Abra uma janela de terminal e digite o comando a seguir, atualizando o texto em vermelho com seu ID de aplicativo e região exclusivos.

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

Adicionar um perfil de serviço

Uma função de serviço é a função AWS Identity and Access Management (IAM) que o Amplify assume ao ligar para outros serviços em seu nome. Siga estas etapas para adicionar um perfil de serviço a um aplicativo SSG que já está implantado com o Amplify.

Para criar um perfil de serviço

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Se você ainda não criou um perfil de serviço em sua conta do Amplify, consulte Adicionar um perfil de serviço para concluir esta etapa de pré-requisito.
3. Escolha o Next.js aplicativo estático ao qual você deseja adicionar uma função de serviço.
4. No painel de navegação, em Configurações do aplicativo, selecione Geral.
5. Na página Detalhes do aplicativo, selecione Editar
6. Em Perfil de serviço, escolha o nome de um perfil de serviço existente ou o nome do perfil de serviço que você criou na etapa 2.
7. Escolha Salvar.

Atualização das configurações de compilação

Antes de reimplantar seu aplicativo com a funcionalidade SSR, é necessário atualizar as configurações da compilação do aplicativo para definir o diretório de saída como `.next`. É possível editar as configurações da compilação no console do Amplify ou em um arquivo `amplify.yml` armazenado em seu repositório. Para obter mais informações, consulte Definição das configurações de compilação de uma aplicação do Amplify.

Veja a seguir um exemplo das configurações da compilação de um aplicativo em que `baseDirectory` está definido como `.next`.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
```

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

Atualização do arquivo package.json

Depois de adicionar um perfil de serviço e atualizar as configurações da compilação, atualize o arquivo `package.json` do aplicativo. Como no exemplo a seguir, defina o script de construção `"next build"` para indicar que o Next.js aplicativo é compatível com páginas SSG e SSR.

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

O Amplify detecta a alteração no arquivo `package.json` em seu repositório e reimplanta o aplicativo com a funcionalidade SSR.

Tornando as variáveis de ambiente acessíveis aos runtimes do lado do servidor

O Amplify Hosting suporta a adição de variáveis de ambiente às compilações do seu aplicativo, definindo-as na configuração do projeto no console do Amplify.

No entanto, um componente de Next.js servidor não tem acesso a essas variáveis de ambiente por padrão. Esse comportamento é intencional para proteger todos os segredos armazenados nas variáveis de ambiente que seu aplicativo usa durante a fase de compilação.

Para tornar variáveis de ambiente específicas acessíveis Next.js, você pode modificar o arquivo de especificação de compilação do Amplify para defini-las nos arquivos de ambiente que Next.js reconhece. Isso permite que o Amplify carregue essas variáveis de ambiente antes de compilar o aplicativo.

> **Important**
>
> É altamente recomendável que você não armazene credenciais, segredos ou informações confidenciais em suas variáveis de ambiente, pois qualquer usuário com acesso aos artefatos de implantação poderá lê-las.
> Para dar à sua função de computação SSR acesso aos AWS recursos, recomendamos o uso de funções do IAM.

O exemplo de especificação de compilação a seguir demonstra como adicionar variáveis de ambiente na seção de comandos de compilação.

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

Neste exemplo, a seção de comandos de compilação inclui dois comandos que gravam variáveis de ambiente no arquivo `.env.production` antes da execução da compilação do aplicativo. O Amplify Hosting permite que seu aplicativo acesse essas variáveis quando o aplicativo recebe tráfego.

A linha a seguir da seção de comandos de compilação no exemplo anterior demonstra como pegar uma variável específica do ambiente de compilação e adicioná-la ao arquivo `.env.production`.

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

Se as variáveis existirem em seu ambiente de compilação, o arquivo `.env.production` conterá as seguintes variáveis de ambiente.

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

A linha a seguir da seção de comandos de compilação no exemplo anterior demonstra como adicionar uma variável de ambiente com um prefixo específico ao arquivo `.env.production`. Neste exemplo, todas as variáveis com o prefixo `NEXT_PUBLIC_` são adicionadas.

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

Se existirem várias variáveis com o prefixo `NEXT_PUBLIC_` no ambiente de compilação, o `.env.production` arquivo terá uma aparência semelhante à seguinte.

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

Variáveis de ambiente do SSR para monorepos

Se você estiver implantando um aplicativo SSR em um monorepo e quiser tornar acessíveis variáveis de ambiente específicas Next.js, deverá prefixar o `.env.production` arquivo com a raiz do aplicativo. O exemplo de especificação de compilação a seguir para um Next.js aplicativo em um monorepo Nx demonstra como adicionar variáveis de ambiente na seção de comandos de compilação.

```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
          commands:
            - npm ci
        build:
          commands:
            - env | grep -e API_BASE_URL -e APP_ENV >> apps/app/.env.production
            - env | grep -e NEXT_PUBLIC_ >> apps/app/.env.production
            - npx nx build app
```

```
    artifacts:
      baseDirectory: dist/apps/app/.next
      files:
        - '**/*'
    cache:
      paths:
        - node_modules/**/*
    buildPath: /
  appRoot: apps/app
```

As linhas a seguir da seção de comandos de compilação no exemplo anterior demonstram como pegar variáveis específicas do ambiente de compilação e adicioná-la ao arquivo `.env.production` para uma aplicação em um monorepo com a raiz `apps/app` da aplicação.

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

Implantando um Next.js aplicativo em um monorepo

O Amplify suporta aplicativos em monorepos genéricos, bem como aplicativos em monorepos criados usando npm workspace, pnpm workspace, Yarn workspace, Nx e Turborepo. Quando você implanta seu aplicativo, o Amplify detecta automaticamente a estrutura de compilação monorepo que você está usando. O Amplify aplica automaticamente as configurações da compilação para aplicativos em um espaço de trabalho npm, espaço de trabalho Yarn ou Nx. Os aplicativos Turborepo e pnpm requerem configuração adicional. Para obter mais informações, consulte Definição de configurações de compilação monorepo.

Para ver um exemplo detalhado do Nx, consulte a postagem do blog Compartilhar código entre Next.js aplicativos com o Nx no AWS Amplify Hosting.

Amplifique o suporte para Nuxt.js

Nuxt é uma estrutura para criar aplicativos web de pilha completa com. Vue.js

Adaptador

> Você pode implantar um Nuxt.js aplicativo no Amplify usando um adaptador predefinido com configuração zero. Para obter mais informações sobre o adaptador, consulte a Documentação do Nuxt.

Tutorial

Para saber como implantar um Nuxt.js aplicativo no Amplify, consulte. Implemente um Nuxt.js aplicativo no Amplify Hosting

Demonstração

Para uma demonstração em vídeo, consulte Nuxt Hosting With ZERO Configuration In Minutes (With AWS) on YouTube.

Amplifique o suporte para Astro.js

O Astro é um framework da Web para a criação de aplicações da Web baseadas em conteúdo.

Adaptador

Você pode implantar um Astro.js aplicativo no Amplify usando um adaptador de comunidade. Não mantemos um adaptador de propriedade da Amplify para o framework do Astro. No entanto, um adaptador está disponível no github. com/alexnguyennz/astro-aws-amplify no site. GitHub Esse adaptador foi criado por um membro da comunidade e não é mantido pela AWS.

Tutorial

Para saber como implantar uma aplicação Astro no Amplify, consulte Implante um Astro.js aplicativo no Amplify Hosting.

Demonstração

Para ver uma demonstração em vídeo, consulte Como implantar um site da Astro AWS no YouTube canal da Amazon Web Services.

Amplifique o suporte para SvelteKit

SvelteKit é uma estrutura para criar aplicativos web de pilha completa com o Svelte.

Adaptador

Você pode implantar um SvelteKit aplicativo no Amplify usando um adaptador de comunidade. Não mantemos um adaptador de propriedade da Amplify para a SvelteKit estrutura. No entanto, um adaptador está disponível no github. com/gzimbron/amplify-adapter no site. GitHub Esse adaptador foi criado por um membro da comunidade e não é mantido pela AWS.

Tutorial

Para saber como implantar um SvelteKit aplicativo no Amplify, consulte. Implemente um SvelteKit aplicativo no Amplify Hosting

Demonstração

Para ver uma demonstração em vídeo, consulte Como implantar um SvelteKit site (com API) AWS no YouTube canal da Amazon Web Services.

Implantação de uma aplicação SSR no Amplify

É possível aplicar essas instruções para implantar uma aplicação criada com qualquer framework com um pacote de implantação que esteja em conformidade com a saída da compilação esperada pelo Amplify. Se você estiver implantando um Next.js aplicativo, nenhum adaptador será necessário.

Se estiver implantando uma aplicação de SSR que use um adaptador de framework, primeiro será necessário instalar e configurar o adaptador. Para instruções, consulte Uso de adaptadores de código aberto para qualquer estrutura SSR.

Para implantar uma aplicação de SSR no Amplify Hosting

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todas as aplicações, escolha Criar nova aplicação.
3. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
4. Na página Adicionar ramificação do repositório, faça o seguinte:

   a. Selecione o nome do repositório a ser conectado.
   b. Selecione o nome da ramificação do repositório a ser conectada.
   c. Escolha Próximo.
5. Na página de configurações do aplicativo, o Amplify detecta Next.js automaticamente os aplicativos SSR.

   Se você estiver implantando um aplicativo SSR que usa um adaptador para outra estrutura, você deve habilitar explicitamente o Amazon Logs. CloudWatch Abra a seção Configurações avançadas e escolha Ativar registros do aplicativo SSR na seção Implantação de Server-Side renderização (SSR).
6. A aplicação precisará de um perfil de serviço do IAM que o Amplify assumirá para entregar os logs à sua Conta da AWS.

   O procedimento para adicionar um perfil de serviço varia dependendo se você deseja criar um novo perfil ou usar um existente.
   - Para criar um novo perfil:
     - Escolha Criar e usar um novo perfil de serviço.
   - Para usar um perfil existente:

     a. Escolha Usar um perfil existente.
     b. Na lista de perfis de serviço, selecione o perfil a ser usado.
7. Escolha Próximo.
8. Na página Revisar, escolha Salvar e implantar.

Recursos com suporte para SSR

Esta seção fornece informações sobre o suporte do Amplify aos recursos de SSR.

O Amplify fornece suporte de Node.js versão para corresponder à Node.js versão usada para criar seu aplicativo.

O Amplify fornece um recurso integrado de otimização de imagem compatível com todas as aplicações de SSR. Se você não quiser usar o recurso de otimização de imagem padrão, é possível implementar um carregador de otimização de imagem personalizado.

Tópicos

Node.js suporte de versão para Next.js aplicativos

Quando o Amplify cria e implanta um aplicativo de Next.js computação, ele usa a versão de Node.js tempo de execução que corresponde à versão principal usada para criar o aplicativo. Node.js

> **Note**
>
> O Amplify Hosting não suporta mais os tempos de execução de Node.js 14, Node.js 16 e Node.js 18. Os tempos de execução suportados incluem Node.js 20, Node.js 22 e Node.js 24.

É possível especificar a versão Node.js a ser usada no recurso de Substituição de pacote ao vivo no console do Amplify. Para obter mais informações sobre como configurar atualizações de pacote ao vivo, consulte Uso de versões específicas de pacotes e dependências na imagem de compilação. Você também pode especificar a versão de Node.js usando outros mecanismos, como comandos nvm. Se você não especificar uma versão, o Amplify vai usar por padrão a versão atual do contêiner de compilação do Amplify.

Otimização de imagem para aplicações de SSR

O Amplify Hosting fornece um recurso integrado de otimização de imagem compatível com todas as aplicações de SSR. Com a otimização de imagem do Amplify, é possível fornecer imagens de

alta qualidade no formato, dimensão e resolução corretos para o dispositivo que as acessa, tudo enquanto mantendo o menor tamanho de arquivo possível.

Atualmente, você pode usar o componente Next.js Image para otimizar imagens sob demanda ou implementar um carregador de imagens personalizado. Se você estiver usando Next.js 13 ou posterior, não precisará realizar nenhuma ação adicional para usar o recurso de otimização de imagem do Amplify. Se estiver implementando um carregador personalizado, consulte o tópico Uso de um carregador personalizado de imagem.

Como usar um carregador personalizado de imagens

Se você usar um carregador personalizado de imagem, o Amplify vai detectar o carregador no arquivo `next.config.js` da sua aplicação e não utilizará o recurso integrado de otimização de imagem. Para obter mais informações sobre os carregadores personalizados Next.js compatíveis, consulte a documentação das Next.js imagens.

Amazon CloudWatch Logs para aplicativos SSR

Amplify envia informações sobre seu tempo de execução de SSR para o Amazon CloudWatch Logs em seu. Conta da AWS Ao implantar um aplicativo SSR, o aplicativo requer um perfil de serviço IAM que o Amplify assume ao chamar outros serviços em seu nome. É possível permitir que a computação do Amplify Hosting crie automaticamente um perfil de serviço ou especificar um perfil que criou.

Se você optar por permitir que o Amplify crie uma função do IAM para você, a função já terá as permissões para criar CloudWatch registros. Se você criar sua própria função do IAM, precisará adicionar as seguintes permissões à sua política para permitir que o Amplify acesse o Amazon CloudWatch Logs.

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

Para obter mais informações sobre perfis de serviço, consulte Adição de um perfil de serviço com permissões para implantar recursos de backend.

Suporte para Amplify Next.js 11 SSR

Se você implantou um Next.js aplicativo no Amplify antes do lançamento da computação do Amplify Hosting em 17 de novembro de 2022, seu aplicativo está usando o provedor SSR anterior do Amplify, o Classic (somente 11). Next.js A documentação nesta seção se aplica somente aos aplicativos implantados usando o provedor SSR Classic (somente Next.js 11).

> **Note**
>
> É altamente recomendável que você migre seus Next.js 11 aplicativos para o provedor de SSR gerenciado por computação do Amplify Hosting. Para obter mais informações, consulte Migrando um aplicativo de Next.js 11 SSR para a computação do Amplify Hosting.

A lista a seguir descreve os recursos específicos que o provedor de SSR Amplify Classic (somente Next.js 11) suporta.

Recursos compatíveis

- Server-side páginas renderizadas (SSR)
- Páginas estáticas
- Rotas de API
- Rotas dinâmicas
- Detecção de todas as rotas
- SSG (geração estática)
- Regeneração estática incremental (ISR)
- Roteamento de subcaminhos internacionalizado (i18n)
- Variáveis de ambiente

Atributos não compatíveis

- Otimização de imagem
- On-DemandRegeneração estática incremental (ISR)
- Roteamento de domínio internacionalizado (i18n)
- Detecção automática de localidade internacionalizada (i18n)

- Middleware
- Middleware de borda
- Rotas de API do Edge

Preços de Next.js 11 aplicativos SSR

Ao implantar seu aplicativo Next.js 11 SSR, o Amplify cria recursos adicionais de back-end em sua conta, incluindo: AWS

- Um bucket do Amazon Simple Storage Service (Amazon S3) que armazena os recursos dos ativos estáticos do seu aplicativo. Para informações sobre as cobranças do Amazon S3, consulte Definição de preço do Amazon S3.
- Uma CloudFront distribuição da Amazon para servir o aplicativo. Para obter informações sobre CloudFront cobranças, consulte Amazon CloudFront Pricing.
- Quatro funções do Lambda @Edge para personalizar o conteúdo entregue. CloudFront

AWS Identity and Access Management permissões para Next.js 11 aplicativos SSR

O Amplify requer permissões AWS Identity and Access Management (IAM) para implantar um aplicativo SSR. Para aplicativos SSR, o Amplify implanta recursos como um bucket do Amazon S3, CloudFront uma distribuiçãoLambda@Edge, funções, uma fila do Amazon SQS (se estiver usando ISR) e funções do IAM. Sem as permissões mínimas exigidas, você receberá um erro `Access Denied` ao tentar implantar sua aplicação SSR. Para fornecer ao Amplify as permissões necessárias, é necessário especificar um perfil de serviço.

Para criar um perfil de serviço do IAM que o Amplify assume ao chamar outros serviços em seu nome, consulte Adição de um perfil de serviço com permissões para implantar recursos de backend. Essas instruções demonstram como criar um perfil que anexa a política `AdministratorAccess-Amplify` gerenciada.

A política `AdministratorAccess-Amplify` gerenciada fornece acesso a vários AWS serviços, incluindo ações do IAM, e deve ser considerada tão poderosa quanto a `AdministratorAccess` política. Essa política fornece mais permissões do que o necessário para implantar seu aplicativo SSR.

É recomendável seguir as práticas recomendadas de concessão de privilégio mínimo e redução das permissões concedidas ao perfil de serviço. Em vez de conceder permissões de acesso de

administrador ao seu perfil de serviço, é possível criar sua própria política do IAM gerenciada pelo cliente que concede somente as permissões necessárias para implantar seu aplicativo SSR. Consulte Criando políticas do IAM no Guia do usuário do IAM para obter instruções sobre como criar uma política gerenciada pelo cliente.

Se você criar sua própria política, consulte a lista a seguir das permissões mínimas necessárias para implantar um aplicativo SSR.

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

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

Solução de problemas de Next.js 11 implantações de SSR

Se você tiver problemas inesperados ao implantar um aplicativo SSR clássico (somente Next.js 11) com o Amplify, consulte os tópicos de solução de problemas a seguir.

Tópicos

- Minha aplicação armazena arquivos estáticos em uma pasta com um caminho reservado
- Minha inscrição atingiu um CloudFront limite
- As funções do Lambda@Edge são criadas na região Leste dos EUA (Norte da Virgínia)
- Meu Next.js aplicativo usa recursos não compatíveis
- As imagens no meu Next.js aplicativo não estão carregando
- Regiões não compatíveis

O diretório de saída da minha aplicação foi substituído

O diretório de saída de um Next.js aplicativo implantado com o Amplify deve ser definido como. `.next` Se o diretório de saída do seu aplicativo estiver sendo substituído, verifique o arquivo `next.config.js`. Para que o diretório de saída da compilação seja padronizado para `.next`, remova a seguinte linha do arquivo:

```
distDir: 'build'
```

Verifique se o diretório de saída está definido como `.next` nas suas configurações da compilação. Para obter informações sobre como visualizar as configurações da compilação do seu aplicativo, consulte Definição das configurações de compilação de uma aplicação do Amplify.

Veja a seguir um exemplo das configurações da compilação de um aplicativo em que `baseDirectory` está definido como `.next`.

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

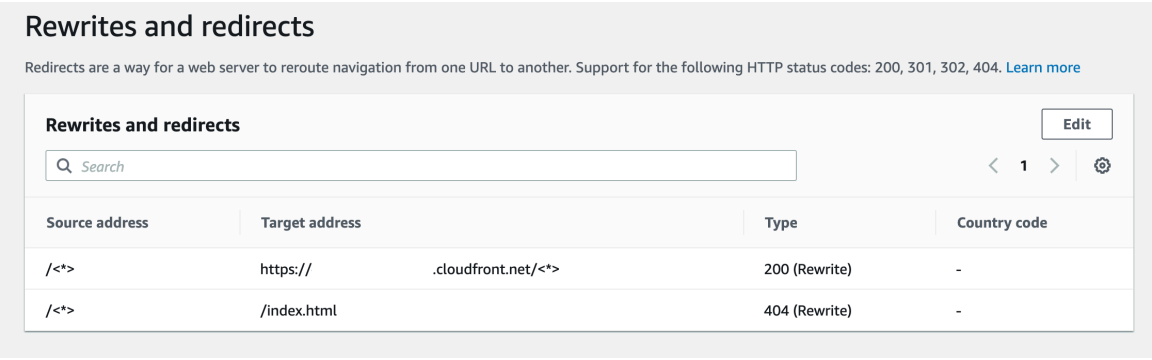
Recebo um erro 404 após implantar meu site SSR

Se você receber um erro 404 após implantar seu site, o problema pode ser causado pela substituição do diretório de saída. Para verificar seu arquivo `next.config.js` e verificar o diretório de saída de compilação correto na especificação de compilação do seu aplicativo, siga as etapas no tópico anterior, O diretório de saída da minha aplicação foi substituído.

Meu aplicativo não tem a regra de reescrita para distribuições CloudFront SSR

Quando você implanta um aplicativo SSR, o Amplify cria uma regra de reescrita para CloudFront suas distribuições de SSR. Se você não conseguir acessar seu aplicativo em um navegador da web, verifique se a regra de CloudFront reescrita existe para seu aplicativo no console do Amplify. Se estiver faltando, é possível adicioná-lo manualmente ou reimplantar seu aplicativo.

Para visualizar ou editar as regras de reescrita e redirecionamento de um aplicativo no console do Amplify, no painel de navegação, escolha Configurações do aplicativo e, em seguida, Regravações e redirecionamentos. A captura de tela a seguir mostra um exemplo das regras de reescrita que o Amplify cria para você quando você implanta um aplicativo SSR. Observe que, neste exemplo, existe uma regra de CloudFront reescrita.



Minha aplicação é muito grande para ser implantada

O Amplify limita o tamanho de uma implantação de SSR a 50 MB. Se você tentar implantar um aplicativo Next.js SSR no Amplify e receber `RequestEntityTooLargeException` um erro, seu aplicativo é muito grande para ser implantado. É possível tentar contornar esse problema adicionando o código de limpeza de cache ao seu arquivo `next.config.js`.

Veja a seguir um exemplo de código no arquivo `next.config.js` que executa a limpeza do cache.

```
module.exports = {
    webpack: (config, { buildId, dev, isServer, defaultLoaders, webpack }) => {
        config.optimization.splitChunks.cacheGroups = { }
```

```
        config.optimization.minimize = true;
        return config
      },
}
```

Minha compilação falha com um erro de falta de memória

Next.js permite armazenar artefatos de compilação em cache para melhorar o desempenho em compilações subsequentes. Além disso, o AWS CodeBuild contêiner do Amplify compacta e carrega esse cache no Amazon S3, em seu nome, para melhorar o desempenho da compilação subsequente. Isso pode fazer com que sua compilação falhe com um erro de falta de memória.

Execute as ações a seguir para evitar que seu aplicativo exceda o limite de memória durante a fase de compilação. Primeiro, remova `.next/cache/**/*` da seção cache.paths das suas configurações da compilação. Em seguida, remova a variável de ambiente `NODE_OPTIONS` do seu arquivo de configurações da compilação. Em vez disso, defina a variável de ambiente `NODE_OPTIONS` no console do Amplify para definir o limite máximo de memória do nó. Para mais informações sobre como configurar as variáveis de ambiente usando o console do Amplify, consulte Configurar variáveis de ambiente.

Depois de fazer essas alterações, tente sua versão novamente. Se for bem-sucedido, adicione `.next/cache/**/*` novamente à seção cache.paths do seu arquivo de configurações da compilação.

Para obter mais informações sobre a configuração do Next.js cache para melhorar o desempenho da compilação, consulte a AWS CodeBuild no Next.js site.

Minha aplicação tem ramificações SSR e SSG

Você não pode implantar um aplicativo que tenha ramificações SSR e SSG. Se você precisar implantar ramificações SSR e SSG, deverá implantar um aplicativo que use somente ramificações SSR e outro aplicativo que use somente ramificações SSG.

Minha aplicação armazena arquivos estáticos em uma pasta com um caminho reservado

Next.js pode servir arquivos estáticos de uma pasta chamada `public` que está armazenada no diretório raiz do projeto. Quando você implanta e hospeda um Next.js aplicativo com o Amplify, seu projeto não pode incluir pastas com o caminho. `public/static` O Amplify reserva o caminho `public/static` para uso ao distribuir o aplicativo. Se seu aplicativo incluir esse caminho, será necessário renomear a pasta `static` antes de implantar com o Amplify.

Minha inscrição atingiu um CloudFront limite

CloudFront as cotas de serviço limitam sua AWS conta a 25 distribuições com funções anexadas do Lambda @Edge. Se você exceder essa cota, poderá excluir quaisquer CloudFront distribuições não utilizadas da sua conta ou solicitar um aumento de cota. Para obter mais informações, consulte Solicitando um Aumento de Cota no Guia do Usuário do Service Quotas.

As funções do Lambda@Edge são criadas na região Leste dos EUA (Norte da Virgínia)

Quando você implanta um Next.js aplicativo, o Amplify cria funções do Lambda @Edge para personalizar o conteúdo entregue. CloudFront As funções do Lambda@Edge são criadas na região Leste dos EUA (Norte da Virgínia), não na região em que seu aplicativo é implantado. Essa é uma restrição do Lambda@Edge. Para obter mais informações sobre as funções do Lambda @Edge, consulte Restrições às funções de borda no Amazon CloudFront Developer Guide.

Meu Next.js aplicativo usa recursos não compatíveis

Os aplicativos implantados com o Amplify oferecem suporte às versões principais até Next.js a versão 11. Para obter uma lista detalhada dos Next.js recursos suportados e não suportados pelo Amplify, consulte. supported features

Quando você implanta um novo Next.js aplicativo, o Amplify usa a versão mais recente compatível do, Next.js por padrão. Se você tem um Next.js aplicativo existente que implantou no Amplify com uma versão mais antiga Next.js do, você pode migrar o aplicativo para o provedor de SSR de computação do Amplify Hosting. Para instruções, consulte Migrando um aplicativo de Next.js 11 SSR para a computação do Amplify Hosting.

As imagens no meu Next.js aplicativo não estão carregando

Quando você adiciona imagens ao seu Next.js aplicativo usando o `next/image` componente, o tamanho da imagem não pode exceder 1 MB. Quando você implanta o aplicativo no Amplify, imagens maiores que 1 MB retornarão um erro 503. Isso é causado por um limite do Lambda@Edge que restringe o tamanho de uma resposta gerada por uma função do Lambda, incluindo cabeçalhos e corpo, a 1 MB.

O limite de 1 MB se aplica a outros artefatos em seu aplicativo, como arquivos PDF e documentos.

Regiões não compatíveis

O Amplify não oferece suporte à implantação do aplicativo SSR Classic (somente Next.js 11) em todas as regiões em que o Amplify AWS está disponível. O SSR clássico (somente Next.js 11) não é

suportado nas seguintes regiões: Europa (Milão) eu-south-1, Oriente Médio (Bahrein) me-south-1 e Ásia-Pacífico (Hong Kong) ap-east-1.

Solução de problemas de implantações SSR

Se você tiver problemas inesperados ao implantar uma aplicação SSR com a computação do Amplify Hosting, consulte Solução de problemas de aplicações renderizadas do lado do servidor no capítulo de solução de problemas do Amplify.

Avançado: adaptadores de código aberto

Os criadores do framework podem usar a especificação de implantação baseada em sistema de arquivos para desenvolver adaptadores de compilação de código aberto personalizados para seus frameworks específicos. Esses adaptadores transformarão a saída da compilação de uma aplicação em um pacote de implantação em conformidade com a estrutura de diretórios esperada pelo Amplify Hosting. Esse pacote de implantação incluirá todos os arquivos e ativos necessários para hospedar uma aplicação, inclusive configuração de runtime, como regras de roteamento.

Se você não estiver usando um framework, poderá desenvolver sua própria solução para gerar a saída da compilação que o Amplify espera.

Tópicos

- Como usar a especificação de implantação do Amplify Hosting para configurar a saída da compilação
- Como implantar um servidor Express usando o manifesto de implantação
- Integração de otimização de imagem para criadores de frameworks
- Uso de adaptadores de código aberto para qualquer estrutura SSR

Como usar a especificação de implantação do Amplify Hosting para configurar a saída da compilação

A especificação de implantação do Amplify Hosting é uma especificação baseada em sistema de arquivos que define a estrutura de diretórios que facilita as implantações no Amplify Hosting. Um framework pode gerar essa estrutura esperada de diretórios como saída do seu comando de compilação, permitindo que o framework aproveite as vantagens dos serviços primitivos do

Amplify Hosting. A Amplify Hosting entende a estrutura do pacote de implantação e o implanta adequadamente.

Para ver uma demonstração em vídeo que explica como usar a especificação de implantação, consulte Como hospedar qualquer AWS Amplify site usando o YouTube canal Amazon Web Services.

Veja a seguir um exemplo da estrutura de pastas que o Amplify espera para o pacote de implantação. Em um alto nível, ele tem uma pasta chamada `static`, uma pasta chamada `compute` e um arquivo de manifesto de implantação chamado `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
```

Compatibilidade do Amplify com primitivo de SSR

A especificação de implantação do Amplify Hosting define um contrato que mapeia de perto os seguintes primitivos.

Ativos estáticos

Fornece aos frameworks a capacidade de hospedar arquivos estáticos.

Computação

Fornece estruturas com a capacidade de executar um servidor Node.js HTTP na porta 3000.

Otimização de imagem

Fornece aos framework um serviço para otimizar imagens em runtime.

Regras de roteamento

Fornece aos framework um mecanismo para mapear caminhos de solicitação de entrada para destinos específicos.

A ferramenta .amplificar- hosting/static directory

É necessário colocar no diretório `.amplify-hosting/static` todos os arquivos estáticos acessíveis ao público que deverão ser oferecidos diretamente do URL da aplicação. Os arquivos dentro desse diretório serão oferecidos por meio do primitivo de ativos estáticos.

Os arquivos estáticos podem ser acessados na raiz (/) do URL da aplicação sem nenhuma alteração em seu conteúdo, nome de arquivo ou extensão. Além disso, os subdiretórios são preservados na estrutura do URL e aparecem antes do nome do arquivo. Por exemplo, `.amplify-hosting/static/favicon.ico` será oferecido de `https://myAppId.amplify-hostingapp.com/`

`favicon.ico`, enquanto `.amplify-hosting/static/_nuxt/main.js` será oferecido de `https://myAppId.amplify-hostingapp.com/_nuxt/main.js`.

Se um framework for compatível com a capacidade de modificar o caminho base da aplicação, ele deverá prefixar o caminho base aos ativos estáticos dentro do diretório `.amplify-hosting/static`. Por exemplo, se o caminho base for `/folder1/folder2`, a saída de compilação para um ativo estático chamado `main.css` será `.amplify-hosting/static/folder1/folder2/main.css`.

A ferramenta .amplificar- hosting/compute directory

Um único recurso computacional é representado por um único subdiretório chamado `default` contido no diretório `.amplify-hosting/compute`. O caminho é `.amplify-hosting/compute/default`. Esse recurso computacional é mapeado para o primitivo computacional do Amplify Hosting.

O conteúdo do subdiretório `default` deve estar em conformidade com as regras a seguir.

- É necessário ter um arquivo na raiz do subdiretório `default` para atuar como ponto de entrada para o recurso computacional.
- O arquivo do ponto de entrada deve ser um Node.js módulo e deve iniciar um servidor HTTP que escuta na porta 3000.
- É possível colocar outros arquivos no subdiretório `default` e fazer referência a eles no código no arquivo do ponto de entrada.
- O conteúdo do subdiretório deve ser independente. O código no módulo de ponto de entrada não pode fazer referência a nenhum módulo fora do subdiretório. Observe que os frameworks podem agrupar seus servidores HTTP da maneira que quiserem. Se for possível iniciar o processo de computação com o comando `node server.js`, com `server.js is` indicando o nome do arquivo de entrada, diretamente do subdiretório, o Amplify vai considerar que a estrutura do diretório está em conformidade com a especificação de implantação.

O Amplify Hosting agrupa e implanta todos os arquivos dentro do subdiretório `default` em um recurso computacional provisionado. Cada recurso computacional recebe 512 MB de armazenamento temporário. Esse armazenamento não é compartilhado entre instâncias de execução, mas é compartilhado entre invocações subsequentes na mesma instância de execução. As instâncias de execução estão limitadas a um tempo máximo de 15 minutos de execução, e o único caminho gravável dentro da instância de execução é o diretório `/tmp`. O tamanho

descompactado de cada pacote de recursos computacionais não pode ultrapassar 220 MB. Por exemplo, o subdiretório `.amplify/compute/default` não pode ultrapassar 220 MB quando descompactado.

A ferramenta .amplify- -manifest.json hosting/deploy arquivo

Use o arquivo `deploy-manifest.json` para armazenar os detalhes da configuração e os metadados de uma implantação. Um arquivo `deploy-manifest.json` deve incluir, no mínimo, um atributo `version`, o atributo `routes` com uma rota abrangente especificada e o atributo `framework` com metadados de framework especificados.

A definição de objeto a seguir demonstra a configuração de um manifesto de implantação.

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

Os tópicos a seguir descrevem os detalhes e o uso de cada atributo no manifesto de implantação.

Como usar o atributo de versão

O atributo `version` define a versão da especificação de implantação que você está implementando. No momento, a única versão da especificação de implantação do Amplify Hosting é a versão 1. O exemplo de JSON a seguir demonstra como usar o atributo `version`.

```
"version": 1
```

Como usar o atributo routes

O atributo `routes` permite que as estruturas aproveitem o primitivo de regras de roteamento do Amplify Hosting. As regras de roteamento fornecem um mecanismo para rotear os caminhos de solicitação de entrada para um destino específico no pacote de implantação. As regras de roteamento determinam somente o destino de uma solicitação recebida e são aplicadas depois que a solicitação é transformada pelas regras de regravação e redirecionamento. Para obter mais informações sobre como o Amplify Hosting processar regravações e redirecionamentos, consulte Configuração de redirecionamentos e regravações para uma aplicação do Amplify.

As regras de roteamento não regravam nem transformam a solicitação. Se uma solicitação recebida corresponder ao padrão de caminho de uma rota, a solicitação será roteada no estado em que se encontra para o destino da rota.

As regras de roteamento especificadas na matriz `routes` devem obedecer às seguintes regras.

- É necessário haver uma rota abrangente especificada. Uma rota abrangente tem o padrão `/*` que corresponde a todas as solicitações recebidas.
- A matriz `routes` pode conter no máximo 25 itens.
- É necessário especificar uma rota `Static` ou uma rota `Compute`.
- Se você especificar uma rota `Static`, o diretório `.amplify-hosting/static` deverá existir.
- Se você especificar uma rota `Compute`, o diretório `.amplify-hosting/compute` deverá existir.
- Se você especificar uma rota `ImageOptimization`, também deverá especificar uma rota `Compute`. Isso é necessário porque a otimização de imagem ainda não é compatível com aplicações puramente estáticas.

A definição de objeto a seguir demonstra a configuração para o objeto `Route`.

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

A tabela a seguir descreve as propriedades do objeto `Route`.

Chave	Tipo	Obrigatório	Description
caminho	String	Sim	Define um padrão que corresponde aos caminhos da solicitação recebida (excluindo a string de consulta). O caminho pode ter até 255 caracteres.

Chave	Tipo	Obrigatório	Description
			Um caminho deve começar com a barra `/`. Um caminho pode conter qualquer um dos seguintes caracteres: [A-Z], [a-z], [0-9], [_-.*$/~"'@:+]. Somente os seguintes caracteres curinga são compatíveis para correspondência de padrão: • `*` (corresponde a 0 ou mais caracteres) • O padrão `/*` é chamado de padrão abrangente e corresponderá a todas as solicitações recebidas.

Chave	Tipo	Obrigatório	Description
target	Destino	Sim	Um objeto que define o destino para o qual rotear a solicitação correspondente. Se houver a especificação de uma rota `Compute`, deverá haver um `ComputeResource` correspondente. Se houver a especificação de uma rota `ImageOptimization` , também deverá haver um `imageSettings` especificado.

Chave	Tipo	Obrigatório	Description
fallback	Destino	Não	Um objeto que define o destino para o fallback se o destino original retornar um erro 404. O tipo `target` e o tipo `fallback` não podem ser iguais para uma rota específic a. Por exemplo, não é permitido fazer o fallback de `Static` para `Static`. Os fallbacks só são compatíveis com solicitações GET que não tenham um corpo. Se houver um corpo na solicitação, ele será descartado durante o fallback.

A definição de objeto a seguir demonstra a configuração para o objeto `Target`.

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

A tabela a seguir descreve as propriedades do objeto `Target`.

Chave	Tipo	Obrigatório	Description
kind	Targetkind	Sim	Um `enum` que define o tipo de destino. Os valores válidos são `Static`, `Compute` e `ImageOpti mization` .
src	String	Sim para `Compute` Não para outros primitivos	Uma string que especifica o nome do subdiretório no pacote de implantaç ão que contém o código executável do primitivo. Válido e necessário somente para o primitivo Compute. O valor deve apontar para um dos recursos computacionais presentes no pacote de implantação. No momento, o único valor compatível para esse campo é `default`.
cacheControl	String	Não	Uma string que especifica o valor do Cache-Control cabeçalho a ser aplicado à resposta. Válido somente para o estático e os

Chave	Tipo	Obrigatório	Description
			ImageOptimization primitivos. O valor especificado é substituído por cabeçalhos personalizados. Para obter mais informações sobre os cabeçalhos personalizados do Amplify Hosting, consulte Configuração de cabeçalhos personalizados para uma aplicação do Amplify. **Note** Esse Cache-Control cabeçalho é aplicado somente às respostas bem-sucedidas com um código de status definido como 200 (OK).

A definição de objeto a seguir demonstra o uso da enumeração `TargetKind`.

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

A lista a seguir especifica os valores válidos para a enumeração `TargetKind`.

Estático

Roteia as solicitações para o primitivo de ativos estáticos.

Computação

Roteia as solicitações para o primitivo de computação.

ImageOptimization

Roteia as solicitações para o primitivo de otimização de imagem.

O exemplo de JSON a seguir demonstra como usar o atributo `routes` com várias regras de roteamento especificadas.

```
"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"
      }
    }
  ]
```

Para obter mais informações sobre como especificar regras de roteamento em seu manifesto de implantação, consulte Práticas recomendadas para a configuração de regras de roteamento.

Como usar o atributo computerResources

O atributo `computeResources` permite que as estruturas forneçam metadados sobre os recursos computacionais provisionados. Cada recurso computacional deve ter uma rota correspondente associada.

A definição de objeto a seguir demonstra o uso do objeto `ComputeResource`.

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

```
};

type ComputeRuntime = 'nodejs20.x' | 'nodejs22.x' | 'nodejs24.x';
```

A tabela a seguir descreve as propriedades do objeto `ComputeResource`.

Chave	Tipo	Obrigatório	Description
name	String	Sim	Especifica o nome do recurso de computação. O nome deve corresponder ao nome do subdiretório dentro de `.amplify-hosting/compute directory` . O único valor válido para a versão 1 da especificação de implantação é `default`.
runtime	ComputeRuntime	Sim	Define o runtime do recurso computacional provisionado. Os valores válidos são `nodejs20.x` , `nodejs22.x` e `nodejs24.x` .
entrypoint	String	Sim	Especifica o nome do arquivo inicial com base no qual o código será executado para o recurso computaci

Chave	Tipo	Obrigatório	Description
			onal especificado. O arquivo deve existir dentro do subdiretó rio que representa um recurso computaci onal.

Se você tiver uma estrutura de diretórios semelhante ao seguinte exemplo.

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

O JSON para o atributo `computeResource` será semelhante ao seguinte exemplo.

```
"computeResources": [
    {
      "name": "default",
      "runtime": "nodejs20.x",
      "entrypoint": "index.js",
    }
  ]
```

Como usar o atributo imageSettings

O atributo `imageSettings` permite que os frameworks personalizem o comportamento do primitivo de otimização de imagem, que fornece otimização sob demanda de imagens em runtime.

A definição de objeto a seguir demonstra o uso do objeto `ImageSettings`.

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

```
  dangerouslyAllowSVG: boolean;
};

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

A tabela a seguir descreve as propriedades do objeto `ImageSettings`.

Chave	Tipo	Obrigatório	Description
sizes	Number[]	Sim	Uma matriz de larguras de imagem compatíveis.
domínios	String[]	Sim	Uma matriz de domínios externos permitidos que podem usar a otimização de imagem. Deixe a matriz vazia para permitir que somente o domínio de implantação use a otimização de imagem.
remotePatterns	RemotePattern[]	Sim	Uma matriz de padrões externos permitidos que podem usar a otimização de imagem. Semelhant e aos domínios, mas fornece mais controle com expressões regulares (regex).
formats	ImageFormat[]	Sim	Uma variedade de formatos de imagem de saída permitidos.

Chave	Tipo	Obrigatório	Description
minimumCacheTTL	Número	Sim	A duração do cache em segundos para as imagens otimizadas.
dangerouslyAllowSVG	Booleano	Sim	Permite URLs de imagem de entrada em SVG. Esse recurso está desabilitado por padrão para fins de segurança.

A definição de objeto a seguir demonstra o uso do objeto `RemotePattern`.

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

A tabela a seguir descreve as propriedades do objeto `RemotePattern`.

Chave	Tipo	Obrigatório	Description
protocolo	String	Não	O protocolo do padrão remoto permitido. O único valor válido é `https`.
hostname	String	Sim	O nome de host do padrão remoto permitido. É possível especificar um caractere literal ou curinga. Um “*” único

Chave	Tipo	Obrigatório	Description
			corresponde a um único subdomínio. Um “**” duplo correspon de a qualquer número de subdomínios. O Amplify não permite curingas gerais que especifiquem apenas “**”.
porta	String	Não	A porta do padrão remoto permitido.
pathname	String	Não	O nome de caminho do padrão remoto permitido.

O exemplo a seguir demonstra o atributo `imageSettings`.

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

```
    "minumumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  }
```

Como usar o atributo framework

Use o atributo `framework` para especificar os metadados do framework.

A definição de objeto a seguir demonstra a configuração para o objeto `FrameworkMetadata`.

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

A tabela a seguir descreve as propriedades do objeto `FrameworkMetadata`.

Chave	Tipo	Obrigatório	Description
name	String	Sim	O nome do framework .
version	String	Sim	A versão do framework. Ele deve ser uma string válida de versionamento semântico (semver).

Práticas recomendadas para a configuração de regras de roteamento

As regras de roteamento fornecem um mecanismo para rotear os caminhos de solicitação de entrada para destinos específicos no pacote de implantação. Em um pacote de implantação, os criadores do framework podem emitir arquivos para a saída da compilação que são implantados em qualquer um dos seguintes destinos:

- Primito de ativos estáticos: os arquivos estão contidos no diretório `.amplify-hosting/static`.

- Primitivo de computação: os arquivos estão contidos no diretório `.amplify-hosting/compute/default`.

Os criadores do framework também fornecem uma matriz de regras de roteamento no arquivo de manifesto de implantação. Cada regra na matriz é comparada com a solicitação recebida em ordem de passagem sequencial, até que haja uma correspondência. Quando houver uma regra correspondente, a solicitação será roteada para o destino especificado na regra correspondente. Como opção, é possível especificar um destino de fallback para cada regra. Se o destino original retornar um erro 404, a solicitação será roteada para o destino de fallback.

A especificação de implantação exige que a última regra na ordem de passagem seja uma regra abrangente. Uma regra abrangente é especificada com o caminho `/*`. Se a solicitação recebida não corresponder a nenhuma das rotas anteriores na matriz de regras de roteamento, a solicitação será roteada para o destino da regra abrangente.

Para frameworks de SSR como Nuxt.js, o destino da regra abrangente deve ser a computação primitiva. Isso ocorre porque os aplicações de SSR têm páginas renderizadas no lado do servidor com rotas que não são previsíveis no momento da compilação. Por exemplo, se uma aplicação Nuxt.js tiver uma página em `/blog/[slug]` na qual `[slug]` esteja um parâmetro de rota dinâmica. O destino regra abrangente é a única maneira de rotear solicitações para essas páginas.

Por outro lado, é possível usar padrões de caminho específicos para direcionar rotas que sejam conhecidas no momento da compilação. Por exemplo, Nuxt.js fornece ativos estáticos do caminho `/_nuxt`. Isso significa que o caminho `/_nuxt/*` pode ser direcionado por uma regra específica de roteamento que roteia solicitações para o primitivo de ativos estáticos.

Roteamento de pasta pública

A maioria dos frameworks de SSR oferece a capacidade de fornecer ativos estáticos mutáveis diretamente de uma pasta `public`. Em geral, arquivos como `favicon.ico` e `robots.txt` são mantidos dentro da pasta `public` e são fornecidos diretamente do URL raiz da aplicação. Por exemplo, o arquivo `favicon.ico` é fornecido diretamente de `https://example.com/favicon.ico`. Observe que não há um padrão de caminho previsível para esses arquivos. Eles são quase que totalmente ditados pelo nome do arquivo. A única maneira de direcionar arquivos dentro da pasta `public` é usar a rota abrangente. No entanto, o destino geral da rota precisa ser o primitivo de computação.

Recomendamos uma das seguintes abordagens para gerenciar sua pasta `public`.

1. Use um padrão de caminho para direcionar caminhos de solicitação que contenham extensões de arquivo. Por exemplo, é possível usar `/*.*` para direcionar todos os caminhos de solicitação que contenham uma extensão de arquivo.

   Observe que essa abordagem pode não ser confiável. Por exemplo, se houver arquivos sem extensões de arquivo dentro da pasta `public`, eles não serão direcionados por essa regra. Outro problema a ser observado com essa abordagem é que a aplicação pode ter páginas com pontos em seus nomes. Por exemplo, uma página em `/blog/2021/01/01/hello.world` será direcionada pela regra `/*.*` . Isso não é ideal, pois a página não é um ativo estático. No entanto, é possível adicionar um destino alternativo a essa regra para garantir que a solicitação retorne para o primitivo computacional quando houver um erro 404 do primitivo estático.

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

2. Identifique os arquivos na pasta `public` no momento da compilação e emita uma regra de roteamento para cada arquivo. Essa abordagem não é escalável, pois há um limite de 25 regras imposto pela especificação de implantação.

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

3. Recomende que os usuários do framework armazenem todos os ativos estáticos mutáveis dentro de uma subpasta dentro da pasta `public`.

   No exemplo a seguir, o usuário pode armazenar todos os ativos estáticos mutáveis dentro da pasta `public/assets`. Em seguida, é possível usar uma regra de roteamento com o padrão de caminho `/assets/*` para direcionar todos os ativos estáticos mutáveis dentro da pasta `public/assets`.

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

4. Especifique um fallback estático para a rota abrangente. Essa abordagem tem desvantagens que são descritas com mais detalhes na próxima seção Catch-all roteamento alternativo.

Catch-all roteamento alternativo

Para frameworks de SSR como Nuxt.js, no qual uma rota abrangente é especificada para o destino primitivo de computação, os criadores do framework podem considerar a especificação de um fallback estático para a rota abrangente a fim de solucionar o problema de roteamento da pasta `public`. No entanto, esse tipo de regra de roteamento interrompe as páginas 404 renderizadas no lado do servidor. Por exemplo, se o usuário final visitar uma página que não exista, a aplicação vai renderizar uma página 404 com um código de status 404. No entanto, se a rota abrangente tiver um fallback estático, a página 404 não será renderizada. Em vez disso, a solicitação retornará para o primitivo estático e ainda terminará com um código de status 404, mas a página 404 não será renderizada.

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

Roteamento de caminho base

Espera-se que frameworks com a capacidade de modificar o caminho base da aplicação possam prefixar o caminho base aos ativos estáticos dentro do diretório `.amplify-hosting/static`. Por exemplo, se o caminho base for `/folder1/folder2`, a saída de compilação para um ativo estático chamado main.css será `.amplify-hosting/static/folder1/folder2/main.css`.

Isso significa que também é necessário atualizar as regras de roteamento para refletir o caminho base. Por exemplo, se o caminho base for `/folder1/folder2`, a regra de roteamento para os ativos estáticos na pasta `public` terá a seguinte aparência.

```
{
    "path": "/folder1/folder2/*.*",
    "target": {
        "kind": "Static"
    }
}
```

Da mesma forma, também é necessário prefixar o caminho base nas rotas do lado do servidor. Por exemplo, se o caminho base for `/folder1/folder2`, a regra de roteamento para a rota `/api` terá a seguinte aparência.

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

No entanto, o caminho base não deverá ser prefixado à rota abrangente. Por exemplo, se o caminho base for `/folder1/folder2`, a rota abrangente permanecerá da seguinte maneira.

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

Nuxt.js exemplos de rotas

Veja a seguir um exemplo de arquivo `deploy-manifest.json` para uma aplicação Nuxt que demonstra como especificar regras de roteamento.

```
{
  "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": "nodejs22.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

Veja a seguir um exemplo de arquivo `deploy-manifest.json` para Nuxt que demonstra como especificar regras de roteamento que incluem caminhos base.

```
{
  "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": "nodejs22.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
```

```
}
```

Para obter mais informações sobre o uso do atributo `routes`, consulte Como usar o atributo routes.

Como implantar um servidor Express usando o manifesto de implantação

Este exemplo explica como implantar um servidor Express básico usando a especificação de implantação do Amplify Hosting. É possível aproveitar o manifesto de implantação fornecido para especificar roteamento, recursos computacionais e outras configurações.

Configurar um servidor Express localmente antes da implantação no Amplify Hosting

1. Crie um novo diretório para seu projeto e instale o Express e o 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. Adicione um arquivo `tsconfig.json` à raiz do seu projeto com o conteúdo a seguir.

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

3. Crie um diretório chamado `src` na raiz do projeto.

4. Crie um arquivo `index.ts` no diretório `src`. Esse será o ponto de entrada para a aplicação que inicia um servidor Express. Configure o servidor para escutar na porta 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. Adicione os seguintes scripts ao seu arquivo `package.json`.

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

6. Crie um diretório chamado `public` na raiz do seu projeto. Em seguida, crie um arquivo chamado `hello-world.txt` com o conteúdo a seguir.

```
Hello world!
```

7. Adicione um arquivo `.gitignore` à raiz do projeto com o conteúdo a seguir.

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

Configurar o manifesto de implantação do Amplify

1. Crie um arquivo chamado `deploy-manifest.json` no diretório raiz do projeto.
2. Copie e cole o manifesto a seguir no seu arquivo `deploy-manifest.json`.

```
{
  "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": "nodejs22.x",
      "entrypoint": "index.js"
    }
  ]
}
```

O manifesto descreve como o Amplify Hosting deve processar a implantação da sua aplicação. As configurações principais são as seguintes.

- version: indica a versão da especificação de implantação que você está usando.
- framework: ajuste para especificar a configuração do seu servidor Express.
- imageSettings: a menos que você esteja lidando com otimização de imagem, essa seção é opcional para um servidor Express.
- routes: são essenciais para direcionar o tráfego para as partes certas da sua aplicação. A rota `"kind": "Compute"` direciona o tráfego para a lógica do seu servidor.
- computeResources: use essa seção para especificar o runtime e o ponto de entrada do seu servidor Express.

Em seguida, configure um script de pós-compilação que mova os artefatos da aplicação compilada para o pacote de implantação `.amplify-hosting`. A estrutura de diretórios estará alinhada com a especificação de implantação do Amplify Hosting.

Configurar o script de pós-compilação

1. Crie um diretório chamado `bin` na raiz do projeto.
2. Crie um arquivo chamado `postbuild.sh` no diretório `bin`. Adicione o conteúdo a seguir ao arquivo `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. Adicione um script `postbuild` ao seu arquivo `package.json`. O arquivo deve ser semelhante ao seguinte.

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

```
  "serve": "node dist/index.js",
  "postbuild": "chmod +x bin/postbuild.sh && ./bin/postbuild.sh"
}
```

4. Execute o comando a seguir para compilar sua aplicação.

```
npm run build
```

5. (Opcional) Ajuste suas rotas para o Express. É possível modificar as rotas em seu manifesto de implantação para que se ajustem ao seu servidor Express. Por exemplo, se você não tiver nenhum ativo estático no diretório `public`, talvez precise apenas da rota abrangente `"path": "/*"` direcionada para Compute. Isso dependerá da configuração do seu servidor.

A estrutura final de diretórios deve ter a seguinte aparência.

```
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
```

Implantar seu servidor

1. Faça push do seu código para o repositório Git e então implante sua aplicação no Amplify Hosting.
2. Atualize suas configurações de compilação a fim de apontar `baseDirectory` para `.amplify-hosting` da seguinte maneira. Durante a compilação, o Amplify detectará o arquivo de manifesto no diretório `.amplify-hosting` e implantará seu servidor Express conforme configurado.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - nvm use 20
        - npm install
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .amplify-hosting
    files:
      - '**/*'
```

3. Para verificar se sua implantação foi bem-sucedida e se o servidor está funcionando corretamente, visite sua aplicação no URL padrão fornecido pelo Amplify Hosting.

Integração de otimização de imagem para criadores de frameworks

Os autores do framework podem integrar o recurso de otimização de imagem do Amplify usando a especificação de implantação do Amplify Hosting. Para habilitar a otimização de imagem, seu manifesto de implantação deverá conter uma regra de roteamento direcionada ao serviço de otimização de imagem. O exemplo a seguir demonstra como configurar a regra de roteamento.

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

{
  "routes": [
    {
      "path": "/images/*",
```

```
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=31536000, immutable"
      }
    }
  ]
}
```

Para obter mais informações sobre como definir as configurações de otimização de imagem usando a especificação de implantação, consulte Como usar a especificação de implantação do Amplify Hosting para configurar a saída da compilação.

Compreensão da API de otimização de imagem

É possível invocar a otimização de imagem em runtime por meio do URL de domínio da aplicação Amplify, no caminho definido pela regra de roteamento.

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

A otimização de imagem impõe as seguintes regras para as imagens.

- O Amplify não pode otimizar os formatos GIF, APNG e SVG nem convertê-los para outro formato.
- As imagens SVG não são exibidas a menos que a configuração `dangerouslyAllowSVG` esteja habilitada.
- A largura ou a altura das imagens de origem não podem exceder 11 MB ou 9.000 pixels.
- O limite de tamanho de uma imagem otimizada é de 4 MB.
- HTTP ou HTTPS são os únicos protocolos com suporte para o fornecimento de imagens com URLs remotos.

Cabeçalhos HTTP

O cabeçalho de solicitação HTTP Accept é usado para especificar os formatos de imagem, expressos como tipos MIME, permitidos pelo cliente (em geral, um navegador da Web). O serviço de otimização de imagem tentará converter a imagem para o formato especificado. O valor especificado para esse cabeçalho terá uma prioridade superior ao parâmetro de consulta de formato. Por exemplo, um valor válido para o cabeçalho Accept é `image/png, image/webp, */*` .
A configuração de formatos especificada no manifesto de implantação do Amplify restringirá os

formatos aos que estiverem na lista. Mesmo que o cabeçalho Accept solicite um formato específico, ele será ignorado se o formato não estiver na lista de permissões.

Parâmetros de solicitação de URI

A tabela a seguir descreve os parâmetros de solicitação de URI para otimização de imagem.

Parâmetro de consulta	Tipo	Obrigatório	Description	Exemplo
url	String	Sim	Um caminho relativo ou URL absoluto para a imagem de origem. Para um URL remoto, há suporte para o protocolo HTTPS. O valor deve estar codificado no URL.	`?url=https%3A%2F%2Fwww.example.com%2Fbuffalo.png`
width	Número	Sim	A largura da imagem otimizada em pixels.	`?width=800`
height	Número	Não	A altura da imagem otimizada em pixels. Se não for especificada, a imagem passará por ajuste de escala automático para	`?height=600`

Parâmetro de consulta	Tipo	Obrigatório	Description	Exemplo
			corresponder à largura.	
fit	Valores de enumeração: `cover`, `contain`, `fill`, `inside`, `outside`	Não	Como a imagem é redimensionada para se ajustar à largura e à altura especificadas.	`?width=800&height=600&fit=cover`
position	Valores de enumeração: `center`, `top`, `right`, `bottom`, `left`	Não	Uma posição a ser usada quando o ajuste for `cover` ou `contain`.	`?fit=contain&position=centre`
trim	Número	Não	Apara pixels de todas as bordas que contenham valores semelhantes à cor de fundo especificada do pixel superior esquerdo.	`?trim=50`

Parâmetro de consulta	Tipo	Obrigatório	Description	Exemplo
estender	Objeto	Não	Adiciona pixels às bordas da imagem usando a cor derivada dos pixels da borda mais próxima. O formato é `{top}_{right}_{bottom}_{left}` , com cada valor indicando o número de pixels a serem adicionados.	`?extend=10_0_5_0`
extract	Objeto	Não	Corta a imagem no retângulo especificado delimitado pela parte superior, esquerda, largura e altura. O formato é {left}_{top}_{width}_{right}, com cada valor indicando o número de pixels a serem recortados.	`?extract=10_0_5_0`

Parâmetro de consulta	Tipo	Obrigatório	Description	Exemplo
formato	String	Não	O formato de saída desejado para a imagem otimizada.	`?format=webp`
quality	Número	Não	A qualidade da imagem, de 1 a 100. Usado somente ao converter o formato da imagem.	`?quality=50`
rotate	Número	Não	Gira a imagem de acordo com o ângulo especificado em número de graus.	`?rotate=45`
flip	Booleano	Não	Espelha a imagem verticalmente (de cima para baixo) no eixo x. Isso sempre ocorre antes da rotação, se houver.	`?flip`

Parâmetro de consulta	Tipo	Obrigatório	Description	Exemplo
flop	Booleano	Não	Espelha a imagem horizontalmente (da esquerda para a direita) no eixo y. Isso sempre ocorre antes da rotação, se houver.	`?flop`
sharpen	Número	Não	Aprimoramento da nitidez na definição das bordas na imagem. Os valores válidos estão entre 0,000001 e 10.	`?sharpen=1`
mediano	Número	Não	Aplica um filtro mediano. Isso remove o ruído ou suaviza as bordas de uma imagem.	`?sharpen=3`
blur	Número	Não	Aplica um desfoque gaussiano com o sigma especificado. Os valores válidos vão de 0,3 a 1.000.	`?blur=20`

Parâmetro de consulta	Tipo	Obrigatório	Description	Exemplo
gamma	Número	Não	Aplica uma correção de gama para melhorar o brilho percebido de uma imagem redimensionada. O valor precisar estar entre 1,0 e 3,0.	`?gamma=1`
negate	Booleano	Não	Inverte as cores da imagem.	`?negate`
normalize	Booleano	Não	Melhora o contraste da imagem ampliando sua iluminação para englobar uma faixa dinâmica completa.	`?normalize`

Parâmetro de consulta	Tipo	Obrigatório	Description	Exemplo
threshold	Número	Não	Substitui qualquer pixel na imagem por um pixel preto, se sua intensidade for menor que o limite especificado. Ou por um pixel branco, se for maior que o limite. Os valores válidos estão entre 0 e 255.	`?threshold=155`
tint	String	Não	Tinge a imagem usando o RGB fornecido enquanto preserva a iluminação da imagem.	`?tint=#7743CE`
grayscale	Booleano	Não	Transforma a imagem em tons de cinza (preto e branco).	`?grayscale`

Código de status de resposta.

A tabela a seguir descreve os código de status de resposta para otimização de imagem.

Success - HTTP status code 200

A solicitação foi atendida com sucesso.

BadRequest - Código de status HTTP 400

- Um parâmetro de consulta de entrada foi especificado incorretamente.
- O URL remoto não está listado como permitido na configuração `remotePatterns`.
- O URL remoto não é resolvido para uma imagem.
- A largura ou altura solicitadas não estão listadas como permitidas na configuração `sizes`.
- A imagem solicitada é SVG, mas a configuração `dangerouslyAllowSvg` está desabilitada.

Not Found - HTTP status code 404

A imagem de origem não foi encontrada.

Content too large - HTTP status code 413

A imagem de origem ou a imagem otimizada ultrapassa o tamanho máximo permitido em bytes.

Noções básicas do armazenamento em cache otimizado de imagens

O Amplify Hosting armazena em cache imagens otimizadas em nossa CDN para que solicitações subsequentes à mesma imagem, com os mesmos parâmetros de consulta, sejam atendidas diretamente do cache. O tempo de vida útil (TTL) do cache é controlado pelo cabeçalho `Cache-Control`. A lista a seguir descreve suas opções para especificar o cabeçalho `Cache-Control`.

- Usando a chave `Cache-Control` dentro da regra de roteamento direcionada à otimização de imagem.
- Usando cabeçalhos personalizados definidos na aplicação Amplify.
- Para imagens remotas, o cabeçalho `Cache-Control` retornado pela imagem remota será respeitado.

O `minimumCacheTTL` especificado nas configurações de otimização de imagem define o limite inferior da diretiva Cache-Control max-age. Por exemplo, se o URL de uma imagem remota responder com um `Cache-Control s-max-age=10`, mas o valor `minimumCacheTTL` for 60, o sistema usará 60.

Uso de adaptadores de código aberto para qualquer estrutura SSR

É possível usar qualquer adaptador de compilação de framework de SSR que tenha sido criado para integração com o Amplify Hosting. Cada framework que oferece um adaptador determina como o

adaptador é configurado e conectado ao seu processo de criação. Normalmente, você instalará o adaptador como uma dependência de desenvolvimento do npm.

Após criar uma aplicação com um framework, use a documentação do framework para aprender como instalar o adaptador do Amplify Hosting e configurá-lo no arquivo de configuração da sua aplicação.

Em seguida, crie um arquivo `amplify.yml` no diretório raiz do seu projeto. No arquivo `amplify.yml`, defina `baseDirectory` para o diretório de saída de compilação da sua aplicação. O framework executará o adaptador durante o processo de compilação para transformar a saída no pacote de implantação do Amplify Hosting.

É possível usar qualquer nome para o diretório de saída da compilação, mas o nome do arquivo `.amplify-hosting` tem significado. Primeiro, o Amplify procura por um diretório definido como `baseDirectory`. Se ele existir, o Amplify vai procurar pela saída da compilação lá. Se o diretório não existir, o Amplify vai procurar a saída da compilação em `.amplify-hosting`, mesmo que isso não tenha sido definido pelo cliente.

Veja a seguir um exemplo das configurações de compilação para uma aplicação. O `baseDirectory` está definido como `.amplify-hosting` para indicar que a saída da compilação está na pasta `.amplify-hosting`. A aplicação será implantada com êxito desde que o conteúdo da pasta `.amplify-hosting` corresponda à especificação de implantação do Amplify Hosting.

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

Após a configuração da sua aplicação para usar um adaptador de framework, será possível implantá-la no Amplify Hosting. Para obter instruções detalhadas, consulte Implantação de uma aplicação SSR no Amplify

Implantar um site estático para o Amplify a partir de um bucket do Amazon S3

É possível usar a integração entre o Amplify Hosting e o Amazon S3 para hospedar conteúdo estático de sites armazenados no S3 com apenas alguns cliques. A implantação no Amplify Hosting oferece os seguintes benefícios e recursos.

- Implantação automática na rede de entrega de AWS conteúdo (CDN) disponível globalmente, desenvolvida por CloudFront
- Suporte a HTTPS
- Conecte facilmente seu site a um domínio personalizado usando o console do Amplify
- Traga seus próprios certificados SSL personalizados
- Monitore seu site com registros e CloudWatch métricas de acesso integrados
- Configure proteção por senha para o seu site
- Crie regras de redirecionamento e reescrita no console do Amplify

Você pode iniciar o processo de implantação no console do Amplify AWS CLI, no ou no. AWS SDKs Você só pode implantar no Amplify a partir de um bucket de uso geral do Amazon S3 localizado em sua própria conta. O Amplify não oferece suporte ao acesso a buckets do S3 entre contas.

Quando você implanta seu aplicativo de um bucket de uso geral do Amazon S3 na Amplify Hosting, as AWS cobranças são baseadas no modelo de preços do Amplify. Para obter mais informações, consulte AWS Amplify Preço.

> **Important**
>
> O Amplify Hosting não está disponível em todos os Regiões da AWS lugares onde o Amazon S3 está disponível. Para implantar um site estático no Amplify Hosting, o bucket de uso geral do Amazon S3 contendo seu site deve estar localizado em uma região em que o Amplify esteja disponível. Para obter uma lista das regiões onde o Amplify está disponível, consulte Endpoints do Amplify no Referência geral da Amazon Web Services.

Consulte os tópicos a seguir para saber como implantar e atualizar um site estático do Amazon S3 para o Amplify Hosting.

Tópicos

Implantação de um site estático a partir do S3 usando o console do Amplify

Use as instruções a seguir para implantar um novo site estático a partir de um bucket de uso geral do Amazon S3 usando o console do Amplify.

Para implantar um site estático a partir de um bucket de uso geral do Amazon S3 usando o console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/
2. Na página Todas as aplicações, escolha Criar nova aplicação.
3. Na página Começar a desenvolver com o Amplify, escolha Implantar sem Git.
4. Escolha Próximo.
5. Na página Iniciar uma implantação manual, faça o seguinte.
    a. Em Nome da aplicação, insira o nome da sua aplicação.
    b. Em Nome da ramificação, insira o nome da ramificação a ser implantada.
6. Em Método, escolha Amazon S3.
7. Em Localização no S3 dos objetos a serem hospedados, escolha Procurar. Selecione o bucket de uso geral do Amazon S3 a ser usado e, em seguida, selecione Escolher prefixo.
8. Escolha Salvar e implantar.

Criação de uma política de bucket para implantar um site estático S3 usando o AWS SDKs

Você pode usar o AWS SDKs para implantar um site estático do Amazon S3 na Amplify Hosting. Se você implantar seu site usando um SDK, deverá criar sua própria política de bucket que conceda permissão ao Amplify Hosting para recuperar os objetos em seu bucket do S3.

Para saber mais sobre como criar políticas de bucket, consulte Políticas de bucket para o Amazon S3, no Guia do usuário do Amazon Simple Storage Service.

O exemplo de política de bucket a seguir concede permissões ao Amplify Hosting para listar buckets e recuperar objetos de bucket para o ID do aplicativo Conta da AWS Amplify e a ramificação especificados.

Para usar este exemplo:

- *amzn-s3-demo-website-bucket/prefix*Substitua pelo nome do bucket e do prefixo do seu site.
- *111122223333*Substitua pelo seu Conta da AWS ID.
- *region-id*Substitua pelo em Região da AWS que seu aplicativo Amplify está localizado, como. **us-east-1**
- *app_id*Substitua pelo ID do aplicativo Amplify. Essas informações estão disponíveis no console do Amplify.
- *branch_name*Substitua pelo nome da sua filial.

> **Note**
>
> Em sua política de bucket, `aws:SourceArn` deve ser um ARN de ramificação codificado em URL (codificação percentual).

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

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

```
}
```

Atualização de um site estático implantado no Amplify a partir de um bucket do S3

Se você atualizar qualquer um dos objetos de um site estático em um bucket do S3 de uso geral hospedado no Amplify, deverá reimplantar a aplicação no Amplify Hosting para que as alterações entrem em vigor. O Amplify Hosting não detecta automaticamente as alterações no bucket do S3. Recomendamos que você use a AWS Command Line Interface (CLI) para atualizar seu site.

Sincronizar atualizações com o S3

Depois de fazer alterações nos arquivos de projeto do seu site, use o seguinte comando s3 sync a seguir para sincronizar as alterações feitas no diretório de origem local com o bucket de uso geral do Amazon S3 de destino. Para usar esse exemplo, *<source>* substitua pelo nome do seu diretório local e *<target>* pelo nome do seu bucket do Amazon S3.

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

Reimplante o site no Amplify Hosting

Use o comando amplify start-deployment a seguir para reimplantar sua aplicação atualizada em um bucket do Amazon S3 para Amplify Hosting. Para usar esse exemplo, *<app_id>* substitua pelo id do seu aplicativo Amplify, *<branch_name>* pelo nome da sua ramificação e *s3://amzn-s3-demo-website-bucket/prefix* pelo seu S3 bucket e prefixo.

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

Atualização de uma implantação do S3 para usar um bucket e um prefixo em vez de um arquivo .zip

Se você já tem um site estático existente implantado no Amplify Hosting a partir de um arquivo .zip em um bucket de uso geral do Amazon S3, é possível atualizar a implantação da aplicação para usar o nome e o prefixo do bucket que contêm os objetos a serem hospedados. Esse tipo de

implantação elimina a necessidade de fazer a transferência de um arquivo separado para seu bucket que contenha o conteúdo compactado da saída da compilação.

Para migrar um site estático de um arquivo .zip para o conteúdo do bucket

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/
2. Na página Todas as aplicações, escolha o nome da aplicação implantada manualmente do qual você deseja migrar usando um arquivo .zip para usar os arquivos da aplicação diretamente.
3. Na página Visão geral da aplicação, escolha Implantar atualizações.
4. Na página Implantar atualizações, em Método, escolha Amazon S3.
5. Em Localização no S3 dos objetos a serem hospedados, escolha Procurar. Selecione o bucket a ser usado e, em seguida, selecione Escolher prefixo.
6. Escolha Salvar e implantar.

Implantação de uma aplicação no Amplify sem um repositório Git

As implantações manuais permitem que você publique sua aplicação da Web com o Amplify Hosting sem conectar um provedor do Git. É possível arrastar e soltar uma pasta compactada da sua área de trabalho e hospedar seu site em segundos. Como alternativa, é possível referenciar ativos em um bucket do Amazon S3 ou especificar uma URL pública para o local em que seus arquivos estão armazenados.

> **Note**
>
> As implantações manuais têm um limite máximo de tamanho de arquivo .zip de 5 GB devido às restrições de operação de cópia do Amazon S3. Se algum de seus artefatos de construção exceder esse tamanho, considere dividi-lo em arquivos menores ou usar um método de implantação alternativo.

Para o Amazon S3, você também pode configurar AWS Lambda gatilhos para atualizar seu site sempre que novos ativos forem carregados. Consulte a postagem do blog Implantar arquivos armazenados no Amazon S3, no Dropbox ou no seu desktop no console AWS Amplify para obter mais detalhes sobre como configurar esse cenário.

O Amplify Hosting não oferece suporte a implantações manuais para aplicativos renderizados do lado do servidor (SSR). Para obter mais informações, consulte Implantação de aplicações renderizadas do lado do servidor com o Amplify Hosting.

Implantações manuais de arrastar e soltar

Para implantar manualmente um aplicativo usando arrastar e soltar

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. No canto superior direito, escolha Criar nova aplicação.
3. Na página Começar a desenvolver com o Amplify, escolha Implantar sem Git. Em seguida, escolha Próximo.
4. Na página Iniciar uma implantação manual, em Nome da aplicação, insira o nome da sua aplicação.

5. Em Nome da ramificação, insira um nome significativo, como **`development`** ou **`production`**.
6. Em Método, escolha Arrastar e soltar.
7. Arraste e solte uma pasta da sua área de trabalho na zona de soltura ou use Escolher pasta .zip para selecionar os arquivos do seu computador. O arquivo que você arrastar e soltar ou selecionar deve ser uma pasta zip que contenha o conteúdo da saída da sua compilação.
8. Escolha Salvar e implantar.

Implantação manual do Amazon S3 ou URL

> **Note**
>
> Se você estiver implantando um site estático a partir do S3, o procedimento a seguir exige que você faça o upload de uma pasta zip com o conteúdo da saída da compilação para o bucket do S3. Recomendamos que você implante um site estático diretamente do S3 usando o nome e o prefixo do bucket. Para obter mais informações sobre esse processo simplificado, consulte Implantar um site estático para o Amplify a partir de um bucket do Amazon S3.

Para implantar manualmente um aplicativo do Amazon S3 ou de uma URL pública

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. No canto superior direito, escolha Criar nova aplicação.
3. Na página Começar a desenvolver com o Amplify, escolha Implantar sem Git. Em seguida, escolha Próximo.
4. Na página Iniciar uma implantação manual, em Nome da aplicação, insira o nome da sua aplicação.
5. Em Nome da ramificação, insira um nome significativo, como **`development`** ou **`production`**.
6. Em Método, escolha Amazon S3 ou Qualquer URL.
7. O procedimento para carregar seus arquivos depende do método de upload.
   - Amazon S3
     a. Em S3 location of objects to host, escolha Procurar S3. Em seguida, selecione o nome do bucket do Amazon S3 na lista. As listas de controle de acesso (ACLs) devem estar habilitadas para o bucket selecionado. Para obter mais informações, consulte Solução de problemas de acesso ao bucket do Amazon S3 para implantações manuais.

b. Selecione o nome do arquivo .zip a ser implantado.

c. Escolha Escolher prefixo.

- Qualquer URL

  - Em URL do recurso, insira a URL do arquivo .zip a ser implantado.

8. Escolha Salvar e implantar.

> **Note**
>
> Ao criar a pasta zip, certifique-se de compactar o conteúdo da saída da compilação e não a pasta de nível superior. Por exemplo, se a saída da compilação gerar uma pasta chamada “build” ou “public”, primeiro navegue até essa pasta, selecione todo o conteúdo e compacte-o a partir daí. Se você não fizer isso, verá um erro de “Acesso negado” porque o diretório raiz do site não será inicializado corretamente.

Solução de problemas de acesso ao bucket do Amazon S3 para implantações manuais

Ao criar um bucket do Amazon S3, você usa a configuração de propriedade de objetos do Amazon S3 para controlar se as listas de controle de acesso ACLs () estão ativadas ou desativadas para o bucket. Para implantar manualmente um aplicativo no Amplify a partir de um bucket do Amazon S3 ACLs , ele deve estar habilitado no bucket.

Se você receber um `AccessControlList` erro ao implantar a partir de um bucket do Amazon S3, o bucket foi criado com ACLs desativado e você deve habilitá-lo no console do Amazon S3. Para obter instruções, consulte Definir propriedade de objetos em um bucket existente no Guia do usuário do Amazon Simple Storage Service.

Gerenciamento da configuração de compilação de uma aplicação do Amplify

É possível personalizar as definições e as configurações de compilação para suas implantações do Amplify. Quando você implanta uma aplicação, o Amplify detecta automaticamente o framework de frontend e as configurações de compilação associadas. É possível personalizar as configurações de compilação na especificação de compilação da aplicação (buildspec) para adicionar variáveis de ambiente, executar comandos de compilação e especificar dependências de compilação.

A imagem de compilação padrão do Amplify vem com vários pacotes e dependências pré-instaladas, mas é possível usar o recurso de atualizações de pacote ao vivo para especificar uma versão específica ou garantir que a versão mais recente esteja sempre instalada. Se você tiver dependências específicas que levem muito tempo para instalar durante uma compilação usando o contêiner padrão do Amplify, será possível criar sua própria imagem de compilação personalziada. Você também pode personalizar o tamanho da instância de compilação para fornecer à implantação da sua aplicação os recursos de CPU, memória e espaço em disco necessários.

As compilações são iniciadas automaticamente com cada confirmação no seu repositório Git e com cada nova implantação. É possível configurar o recurso de webhooks de entrada para iniciar uma compilação sem se comprometer com seu repositório Git.

O recurso de notificações de compilação permite que você compartilhe informações com os membros da equipe sobre êxitos e fracassos de compilação.

Tópicos

Definição das configurações de compilação de uma aplicação do Amplify

Quando você implanta uma aplicação, o Amplify detecta automaticamente o framework de frontend e as configurações de compilação associadas, inspecionando o arquivo `package.json` em seu repositório Git. Você tem as seguintes opções para armazenar as configurações de compilação do seu aplicativo:

- Salvar as configurações de compilação no console do Amplify: o console do Amplify detecta automaticamente configurações de compilação e as salva para que possam ser acessadas por meio do console do Amplify. O Amplify aplica essas configurações a todas as suas ramificações, a menos que um arquivo `amplify.yml` seja encontrado em seu repositório.
- Salvar as configurações de compilação no repositório – Faça download do arquivo `amplify.yml` e adicione-o à raiz do repositório.

> **Note**
>
> As configurações de compilação ficam visíveis no menu Hospedagem do console do Amplify somente quando uma aplicação for configurada para implantação contínua e conectada a um repositório git. Para obter instruções sobre esse tipo de implantação, consulte Noções básicas.

Referência de especificação de compilação

A especificação de compilação (buildspec) de uma aplicação do Almplify é coleção de comandos de compilação e configurações do YAML que o Amplify usa para executar sua compilação. A lista a seguir descreve essas configurações e como elas são usadas.

version

O número da versão YAML do Amplify.

appRoot

O caminho dentro do repositório em que esse aplicativo reside em. Ignorado, a menos que vários aplicativos sejam definidos.

env

Adicione variáveis de ambiente a essa seção. Também é possível adicionar variáveis de ambiente usando o console.

backend

Execute comandos da Amplify CLI para provisionar um backend, atualizar funções do Lambda ou esquemas do GraphQL como parte da implantação contínua.

frontend

Executa comandos de compilação de frontend.

teste

Execute comandos durante uma fase de teste. Saiba como adicionar testes ao seu aplicativo.

fases da compilação

O frontend, o backend e o teste têm três fases que representam os comandos executados durante cada sequência da compilação.

- preBuild – O script preBuild é executado antes que a compilação em si seja iniciada, mas depois que o Amplify instala as dependências.
- build (criação) – Seus comandos de criação.
- postBuild – O script pós-compilação é executado depois que a compilação tiver sido concluída e o Amplify copiou todos os artefatos necessários para o diretório de saída.

buildpath

O caminho a ser usado para executar a compilação. O Amplify usa esse caminho para localizar seus artefatos de compilação. Se você não especificar um caminho, o Amplify usa a raiz do aplicativo monorepo, por exemplo `apps/app`.

artifacts>base-directory

O diretório no qual os artefatos de compilação existem.

artifacts>files

Especifique os arquivos dos artefatos que você deseja implantar. Digite `**/*` para incluir todos os arquivos.

cache

Especifica dependências de tempo de compilação, como a pasta node_modules. Durante a primeira compilação, os caminhos fornecidos aqui são armazenados em cache. Nas compilações subsequentes, o Amplify restaura o cache nos mesmos caminhos antes de executar seus comandos.

O Amplify considera que todos os caminhos de cache fornecidos são relativos à raiz do seu projeto. Contudo, o Amplify não permite a navegação fora da raiz do projeto. Por exemplo, se você especificar um caminho absoluto, a compilação terá êxito sem nenhum erro, mas o caminho não será armazenado em cache.

Sintaxe de referência do YAML de especificação de compilação

O exemplo de especificação de compilação a seguir demonstra a sintaxe básica do YAML.

```
version: 1
env:
  variables:
    key: value
backend:
  phases:
    preBuild:
      commands:
        - *enter command*
    build:
      commands:
        - *enter command*
    postBuild:
        commands:
        - *enter command*
frontend:
  buildpath:
  phases:
    preBuild:
      commands:
        - cd react-app
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
```

```
    files:
        - location
        - location
    discard-paths: yes
    baseDirectory: location
  cache:
    paths:
        - path # A cache path relative to the project root
        - path # Traversing outside of the project root is not allowed
test:
  phases:
    preTest:
      commands:
        - *enter command*
    test:
      commands:
        - *enter command*
    postTest:
      commands:
        - *enter command*
  artifacts:
    files:
        - location
        - location
    configFilePath: *location*
    baseDirectory: *location*
```

Edição da especificação de compilação

É possível personalizar as configurações de compilação de uma aplicação editando a especificação de compilação (buildspec) no console do Amplify. As configurações de compilação são aplicadas a todas as ramificações da aplicação, exceto às ramificações que tenham um arquivo `amplify.yml` salvo no repositório Git.

Para editar as configurações de compilação no console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha a aplicação para a qual deseja editar as configurações de compilação.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Configurações de compilação.

4. Na página Configurações da compilação, na seção Especificação de compilação da aplicação, escolha Editar.
5. Na janela Editar especificação de compilação, insira suas atualizações.
6. Escolha Salvar.

É possível usar os exemplos descritos nos tópicos a seguir para atualizar suas configurações de compilação para cenários específicos.

Tópicos

Definição de configurações de compilação específicas de ramificações com scripts

É possível usar o script shell bash para definir configurações de compilação específicas de ramificação. Por exemplo, o script a seguir usa a variável de ambiente do sistema $ AWS_BRANCH para executar um conjunto de comandos se o nome da ramificação for principal e um conjunto diferente de comandos se o nome da ramificação for dev.

```
frontend:
  phases:
    build:
      commands:
```

```
        - if [ "${AWS_BRANCH}" = "main" ]; then echo "main branch"; fi
        - if [ "${AWS_BRANCH}" = "dev" ]; then echo "dev branch"; fi
```

Configuração de um comando para navegar até uma subpasta

Para monorepos, os usuários desejam poder fazer `cd` em uma pasta para executar a compilação. Depois de executar o comando `cd`, ele será aplicado a todos os estágios da compilação. Assim, não será necessário repetir o comando em fases separadas.

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

Implantação de backend com o frontend para uma aplicação Gen 1

> **Note**
>
> Esta seção se aplica somente a aplicações do Amplify Gen 1. Um backend Gen 1 é criado usando o Amplify Studio e a interface de linha de comando (CLI) do Amplify.

O comando `amplifyPush` é um script auxiliar que ajuda com as implantações de backend. As configurações de compilação abaixo determinam automaticamente o ambiente de backend correto a ser implantado para a ramificação atual.

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

```
  phases:
    build:
      commands:
        - amplifyPush --simple
```

Definir a pasta de saída

As configurações de criação a seguir definem o diretório de saída como a pasta pública.

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

Instalar pacotes como parte da compilação

É possível usar os comandos `npm` ou `yarn` para instalar pacotes durante a compilação.

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

Usar um registro privado de npm

É possível adicionar referências a um registro privado nas configurações de criação ou adicioná-lo como uma variável de ambiente.

```
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
```

Instalar pacotes do SO

A AL2023 imagem do Amplify executa seu código com um usuário não privilegiado chamado. `amplify` O Amplify concede a esse usuário privilégios para executar comandos do SO usando o comando `sudo` do Linux. Se você quiser instalar pacotes de sistema operacional para dependências ausentes, é possível usar comandos como `yum` e `rpm` com `sudo`.

O exemplo de seção de compilação a seguir demonstra a sintaxe para instalar um pacote de sistema operacional usando o comando `sudo`.

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

Configuração de armazenamento de chave-valor para cada compilação

O `envCache` fornece armazenamento de chave-valor no momento da compilação. Os valores armazenados no `envCache` só podem ser modificados durante uma compilação e podem ser reutilizados na próxima compilação. Com o `envCache`, é possível armazenar informações sobre o ambiente implantado e disponibilizá-lo para o contêiner de compilação em criações sucessivas. Ao contrário dos valores armazenados no `envCache`, as alterações nas variáveis do ambiente durante uma compilação não são mantidas para compilações futuras.

Exemplo de uso:

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

Ignorar a compilação para uma confirmação

Para pular uma compilação automática em um determinado commit, inclua o texto [skip-cd] no final da mensagem do commit.

Desativação das compilações automáticas em cada confirmação

É possível configurar o Amplify para desativar as compilações automáticas em cada confirmação de código. Para configurar, escolha Configurações da aplicação, Configurações de ramificação e, em seguida, localize a seção Ramificações que lista as ramificações conectadas. Selecione uma ramificação e escolha Ações, Desativar compilação automática. Novas confirmações para essa ramificação não iniciarão mais uma nova compilação.

Configuração de compilação e implantação de frontend baseado em diff

É possível configurar o Amplify para usar compilações de frontend baseadas em diff. Se ativado, no início de cada compilação, o Amplify tenta executar um diff na sua pasta `appRoot` ou na pasta `/src/` por padrão. Se o Amplify não encontrar nenhuma diferença, ele ignora as etapas de compilação, teste (se configurado) e implantação do frontend e não atualiza seu aplicativo hospedado.

Para configurar o frontend baseado em diff, criar e implantar

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual configurar a compilação e implantação de frontend com base em diff.
3. No painel de navegação, escolha Hospedagem, Variáveis de ambiente.
4. Na seção Variáveis de ambiente, escolha Gerenciar variáveis.
5. O procedimento para configurar a variável de ambiente varia dependendo se você está habilitando ou desabilitando a compilação e implantação de frontend com base em diff.
   - Para habilitar a compilação e implantação de frontend baseado em diff
     a. Na seção Gerenciar variáveis, em Variável, insira `AMPLIFY_DIFF_DEPLOY`.
     b. Em Valor, insira `true`.
   - Para desabilitar a compilação e implantação de frontend baseado em diff
     - Execute um destes procedimentos:
       - Na seção Gerenciar variáveis, localize `AMPLIFY_DIFF_DEPLOY`. Em Valor, insira `false`.
       - Remova a variável de ambiente `AMPLIFY_DIFF_DEPLOY`.
6. Escolha Salvar.

Opcionalmente, é possível definir a variável de ambiente `AMPLIFY_DIFF_DEPLOY_ROOT` para substituir o caminho padrão por um caminho relativo à raiz do seu repositório, como `dist`.

Configuração de compilações de backend baseadas em diff para uma aplicação Gen 1

> **Note**
>
> Esta seção se aplica somente a aplicações do Amplify Gen 1. Um backend Gen 1 é criado usando o Amplify Studio e a interface de linha de comando (CLI) do Amplify.

É possível configurar o Amplify Hosting para usar compilações de backend baseadas em diff usando a variável de ambiente `AMPLIFY_DIFF_BACKEND`. Quando você habilita compilações de backend baseadas em diff, no início de cada compilação, o Amplify tenta executar uma comparação na pasta em seu repositório `amplify`. Se o Amplify não encontrar nenhuma diferença, ele pula a etapa de compilação do backend e não atualiza seus atributos de backend. Se seu projeto não tiver uma pasta `amplify` no seu repositório, o Amplify ignorará o valor da variável de ambiente `AMPLIFY_DIFF_BACKEND`.

Se você atualmente tem comandos personalizados especificados nas configurações de compilação da sua fase de backend, as compilações condicionais de backend não funcionarão. Se quiser que esses comandos personalizados sejam executados, é necessário movê-los para a fase de frontend das configurações de compilação no arquivo `amplify.yml` do seu aplicativo.

Para configurar compilações de backend baseadas em diff

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual configurar as compilações de backend baseadas em diff.
3. No painel de navegação, escolha Hospedagem, Variáveis de ambiente.
4. Na seção Variáveis de ambiente, escolha Gerenciar variáveis.
5. O procedimento para configurar a variável de ambiente varia dependendo se você está habilitando ou desabilitando compilações de backend baseadas em diff.
   - Para habilitar compilações de backend baseadas em diff
     a. Na seção Gerenciar variáveis, em Variável, insira `AMPLIFY_DIFF_BACKEND`.
     b. Em Valor, insira `true`.
   - Para desativar as compilações de backend baseadas em diff

- Execute um destes procedimentos:
  - Na seção Gerenciar variáveis, localize `AMPLIFY_DIFF_BACKEND`. Em Valor, insira `false`.
  - Remova a variável de ambiente `AMPLIFY_DIFF_BACKEND`.

6. Escolha Salvar.

Definição de configurações de compilação monorepo

Quando você armazena vários projetos ou microsserviços em um único repositório, isso é chamado de monorepo. É possível usar o Amplify Hosting para implantar aplicativos em um monorepo sem criar várias configurações de compilação ou configurações de ramificação.

O Amplify suporta aplicativos em monorepos genéricos, bem como aplicativos em monorepos criados usando npm workspace, pnpm workspace, Yarn workspace, Nx e Turborepo. Quando você implanta seu aplicativo, o Amplify detecta automaticamente a ferramenta de compilação monorepo que você está usando. O Amplify aplica automaticamente as configurações da compilação para aplicativos em um espaço de trabalho npm, espaço de trabalho Yarn ou Nx. Os aplicativos Turborepo e pnpm requerem configuração adicional. Para obter mais informações, consulte Configurar aplicativos Turborepo e pnpm monorepo.

É possível salvar as configurações de compilação de um monorepo no console do Amplify ou baixar o arquivo `amplify.yml` e adicioná-lo à raiz do seu repositório. O Amplify aplica as configurações salvas no console a todas as suas ramificações, a menos que encontre um arquivo `amplify.yml` no seu repositório. Quando um arquivo `amplify.yml` está presente, suas configurações substituem todas as configurações de compilação salvas no console do Amplify.

Referência de sintaxe do YAML da especificação de compilação monorepo

A sintaxe YAML para uma especificação de compilação monorepo é diferente da sintaxe YAML para um repositório que contém um único aplicativo. Para um monorepo, você declara cada projeto em uma lista de aplicativos. É necessário fornecer a seguinte chave adicional `appRoot` para cada aplicativo declarado na especificação de compilação do monorepo:

appRoot

A raiz, dentro do repositório, na qual o aplicativo é iniciado. Essa chave deve existir e ter o mesmo valor da variável de ambiente `AMPLIFY_MONOREPO_APP_ROOT`. Para obter instruções

sobre como definir essa variável de ambiente, consulte Definindo a variável de ambiente AMPLIFY_MONOREPO_APP_ROOT.

O exemplo de especificação de compilação do monorepo a seguir demonstra como declarar vários aplicativos Amplify no mesmo repositório. Os dois aplicativos, `react-app` e `angular-app` estão declarados na lista `applications`. A chave `appRoot` de cada aplicativo indica que o aplicativo está localizado na pasta raiz `apps` do repositório.

O atributo `buildpath` está definido como `/` para executar e criar o aplicativo a partir da raiz do projeto monorepo. O atributo `baseDirectory` é o caminho relativo de `buildpath`.

Sintaxe YAML da especificação de compilação do Monorepo

```
version: 1
applications:
  - appRoot: apps/react-app
    env:
      variables:
        key: value
    backend:
      phases:
        preBuild:
          commands:
            - *enter command*
        build:
          commands:
            - *enter command*
        postBuild:
            commands:
            - *enter command*
    frontend:
      buildPath: / # Run install and build from the monorepo project root
      phases:
        preBuild:
          commands:
            - *enter command*
            - *enter command*
        build:
          commands:
            - *enter command*
      artifacts:
        files:
```

```
            - location
            - location
        discard-paths: yes
        baseDirectory: location
      cache:
        paths:
            - path
            - path
    test:
      phases:
        preTest:
          commands:
            - *enter command*
        test:
          commands:
            - *enter command*
        postTest:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        configFilePath: *location*
        baseDirectory: *location*
  - appRoot: apps/angular-app
    env:
      variables:
        key: value
    backend:
      phases:
        preBuild:
          commands:
            - *enter command*
        build:
          commands:
            - *enter command*
        postBuild:
            commands:
            - *enter command*
    frontend:
      phases:
        preBuild:
          commands:
```

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

Uma aplicação usando o exemplo de especificação de compilação a seguir será criada sob a raiz do projeto e os artefatos de construção estarão localizados em `/packages/nextjs-app/.next`.

```
applications:
  - frontend:
      buildPath: '/'  # run install and build from monorepo project root
      phases:
        preBuild:
          commands:
```

```
            - npm install
        build:
          commands:
            - npm run build --workspace=nextjs-app
      artifacts:
        baseDirectory: packages/nextjs-app/.next
        files:
          - '**/*'
      cache:
        paths:
          - node_modules/**/*
    appRoot: packages/nextjs-app
```

Definindo a variável de ambiente AMPLIFY_MONOREPO_APP_ROOT

Quando você implanta um aplicativo armazenado em um monorepo, a variável de ambiente `AMPLIFY_MONOREPO_APP_ROOT` do aplicativo deve ter o mesmo valor do caminho da raiz do aplicativo, em relação à raiz do seu repositório. Por exemplo, um monorepo nomeado `ExampleMonorepo` com uma pasta raiz chamada `apps`, que contém `app1`, `app2` e `app3` tem a seguinte estrutura de diretórios:

```
ExampleMonorepo
  apps
    app1
    app2
    app3
```

Neste exemplo, o valor da variável de ambiente `AMPLIFY_MONOREPO_APP_ROOT` para `app1` é `apps/app1`.

Quando você implanta um aplicativo monorepo usando o console do Amplify, o console define automaticamente a variável de ambiente `AMPLIFY_MONOREPO_APP_ROOT` usando o valor que você especifica para o caminho até a raiz do aplicativo. No entanto, se seu aplicativo monorepo já existe no Amplify ou é implantado usando AWS CloudFormation, você deve definir manualmente a variável de ambiente na seção Variáveis de `AMPLIFY_MONOREPO_APP_ROOT` ambiente no console do Amplify.

Configurar a variável de ambiente AMPLIFY_MONOREPO_APP_ROOT automaticamente durante a implantação

As instruções a seguir demonstram como implantar um aplicativo monorepo com o console do Amplify. Amplify define automaticamente a variável de ambiente `AMPLIFY_MONOREPO_APP_ROOT` e usando a pasta raiz do aplicativo que você especifica no console.

Para implantar um aplicativo monorepo com o console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha Criar nova aplicação no canto superior direito.
3. Na página Comece a desenvolver com o Amplify, escolha seu provedor de Git e escolha Avançar.
4. Na página Adicionar ramificação do repositório, faça o seguinte:
   a. Escolha o nome do seu repositório na lista.
   b. Escolha o nome da ramificação a ser usada.
   c. Selecione Minha aplicação é um monorepo
   d. Insira o caminho para seu aplicativo em seu monorepo, por exemplo, **`apps/app1`**.
   e. Escolha Próximo.
5. Na página Configurações da aplicação, é possível usar as configurações padrão ou personalizar as configurações de compilação da sua aplicação. Na seção Variáveis de ambiente, o Amplify define `AMPLIFY_MONOREPO_APP_ROOT` no caminho que você especificou na etapa 4d.
6. Escolha Próximo.
7. Na página Revisar, escolha Salvar e implantar.

Configurar a variável de ambiente AMPLIFY_MONOREPO_APP_ROOT para um aplicativo existente

Use as instruções a seguir para definir manualmente a variável de `AMPLIFY_MONOREPO_APP_ROOT` ambiente para um aplicativo que já está implantado no Amplify ou foi criado usando. CloudFormation

Para definir a variável de ambiente AMPLIFY_MONOREPO_APP_ROOT para um aplicativo existente

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o nome do aplicativo para o qual definir a variável de ambiente.
3. No painel de navegação, escolha Hospedagem, e, em seguida, Variáveis de ambiente.

4. Na página Variáveis de ambiente, selecione Gerenciar variáveis de ambiente.
5. Na seção Gerenciador de variáveis, faça o seguinte:
   a. Selecione Add new (Adicionar novo).
   b. Em Variável, insira a chave `AMPLIFY_MONOREPO_APP_ROOT`.
   c. Em Valor, insira o caminho para o aplicativo, por exemplo **`apps/app1`**.
   d. Para Ramificação, por padrão, o Amplify aplica a variável de ambiente a todas as ramificações.
6. Escolha Salvar.

Configurar aplicativos Turborepo e pnpm monorepo

As ferramentas de construção do Turborepo e pnpm workspace monorepo obtêm informações de configuração dos arquivos `.npmrc`. Ao implantar um aplicativo monorepo criado com uma dessas ferramentas, é necessário ter um arquivo `.npmrc` no diretório raiz do projeto.

No arquivo `.npmrc`, defina o vinculador para instalar os pacotes do Node em `hoisted`. Você tem as seguintes opções para fazer o seguinte:

```
node-linker=hoisted
```

Para obter mais informações sobre arquivos `.npmrc` e configurações, consulte pnpm .npmrc na documentação do pnpm.

O Pnpm não está incluído no contêiner de compilação padrão do Amplify. Para os aplicativos pnpm workspace e Turborepo, é necessário adicionar um comando para instalar o pnpm na fase `preBuild` das configurações de compilação do seu aplicativo.

O exemplo a seguir, trecho de uma especificação de compilação mostra uma fase `preBuild` com um comando para instalar o pnpm.

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

Personalização da imagem de compilação

É possível usar uma imagem de compilação personalizada para fornecer um ambiente de compilação personalizado para um aplicativo Amplify. Se você tiver dependências específicas que levam muito tempo para instalar durante uma compilação usando o contêiner padrão do Amplify, poderá criar sua própria imagem do Docker e fazer referência a ela durante uma compilação. As imagens podem ser hospedadas no Amazon Elastic Container Registry público.

Para que uma imagem de compilação personalizada funcione como uma imagem de compilação do Amplify, ela deve atender aos requisitos a seguir.

Requisitos de imagens de compilação personalizadas

1. Uma distribuição Linux que suporta a GNU C Library (glibc), como a Amazon Linux, compilada para a arquitetura x86-64.
2. cURL: quando ativamos sua imagem personalizada, baixamos o executor de compilação para o contêiner e, portanto, exigimos que cURL esteja presente. Se essa dependência estiver ausente, a compilação falha instantaneamente sem nenhuma saída, pois o executor de compilação não é capaz de produzir nenhuma saída.
3. Git: para clonar o repositório Git, exigimos que Git seja instalado na imagem. Se essa dependência estiver ausente, a etapa Clonar repositório falhará.
4. OpenSSH: para clonar seu repositório com segurança, é necessário que o OpenSSH configure a chave SSH temporariamente durante a compilação. O pacote OpenSSH fornece os comandos que o executor de compilação precisa para fazer isso.
5. Bash e The Bourne Shell: esses dois utilitários são usados para executar comandos durante a compilação. Se eles não estiverem instalados, suas compilações poderão falhar antes de serem iniciadas.
6. Node.JS+NPM: Nosso executor de compilação não instala o Node. Em vez disso, ele depende da instalação do nó e do NPM na imagem. Isso é necessário apenas para compilações que exigem pacotes NPM ou comandos específicos de Node. No entanto, é altamente recomendável instalá-los porque, quando estão presentes, o executor de compilação do Amplify pode usar essas ferramentas para melhorar a execução da compilação. O recurso de substituição de pacote do Amplify usa o NPM para instalar o Hugo-extended pacote quando você define uma substituição para o Hugo.

Os pacotes a seguir não são obrigatórios, mas sua instalação é altamente recomendada.

1. NVM (Node Version Manager): recomendamos que você instale esse gerenciador de versões se precisar administrar versões diferentes do Node. Quando você define uma substituição, o recurso de substituição de pacotes do Amplify é usado NVM para alterar Node.js as versões antes de cada compilação.
2. Wget: o Amplify pode usar o utilitário Wget para baixar arquivos durante o processo de compilação. Recomendamos que você o instale em sua imagem personalizada.
3. Tar: o Amplify pode usar o utilitário Tar para descompactar arquivos baixados durante o processo de compilação. Recomendamos que você o instale em sua imagem personalizada.

Configuração de uma imagem de compilação personalizada para uma aplicação

Use o procedimento a seguir para configurar uma imagem de compilação personalizada para uma aplicação no console do Amplify.

Para configurar uma imagem de compilação personalizada hospedada no Amazon ECR

1. Consulte Conceitos básicos no Guia do usuário público do Amazon ECR para configurar um repositório público do Amazon ECR com uma imagem do Docker.
2. Faça login Console de gerenciamento da AWS e abra o console do Amplify.
3. Escolha o aplicativo para o qual você quer configurar uma imagem de compilação personalizada.
4. No painel de navegação, escolha Hospedagem, Configurações de compilação.
5. Na página Configurações da compilação, na seção Configurações da compilação de imagem, escolha Editar.
6. Na página Editar configurações de imagem da compilação, expanda o menu Imagem de compilação e escolha Imagem de compilação personalizada.
7. Insira o nome do repositório público do Amazon ECR criado na Etapa 1. É aqui que sua imagem de compilação é hospedada. Por exemplo, se o nome do seu repositório for ecr-examplerepo, você digitaria **`public.ecr.aws/xxxxxxxx/ecr-examplerepo`**.
8. Escolha Salvar.

Uso de versões específicas de pacotes e dependências na imagem de compilação

As atualizações de pacote ao vivo possibilitam que você especifique versões de pacotes e dependências para uso na imagem de compilação padrão do Amplify. A imagem de compilação padrão é fornecida com vários pacotes e dependências pré-instalados (por exemplo, Hugo, CLI do Amplify, Yam etc.). Com atualizações de pacote ao vivo, é possível substituir a versão dessas dependências e especificar uma versão específica ou garantir que a versão mais recente esteja sempre instalada.

Se as atualizações de pacote ao vivo estiverem habilitadas, antes que a compilação seja executada, o executor de compilação primeiro atualiza (ou faz downgrade) as dependências especificadas. Isso aumenta o tempo de compilação proporcional ao tempo necessário para atualizar as dependências, mas o benefício é que é possível garantir que a mesma versão de uma dependência seja usada para criar o aplicativo.

> **Warning**
>
> Definir a Node.js versão como a mais recente faz com que as compilações falhem. Em vez disso, você deve especificar uma Node.js versão exata`20`, como`21.5`, ou`v0.1.2`.

Para configurar atualizações de pacote ao vivo

1. Faça login Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual você quer configurar as atualizações de pacotes ao vivo.
3. No painel de navegação, escolha Hospedagem, Configurações de compilação.
4. Na página Configurações da compilação, na seção Configurações da compilação de imagem, escolha Editar.
5. Na página Editar configurações de imagem de compilação, na lista de Atualizações de pacotes ao vivo, escolha Adicionar novo.
6. Em Pacote, selecione a dependência a ser substituída.
7. Em Versão, mantenha a versão padrão mais recente ou insira uma versão específica da dependência. Se você usar mais recente, a dependência sempre será atualizada para a versão mais recente disponível.
8. Escolha Salvar.

Configuração da instância de compilação de uma aplicação do Amplify

O Amplify Hosting oferece tamanhos de instância de compilação configuráveis que permitem que você forneça à instância de compilação da sua aplicação os recursos de CPU, memória e espaço em disco necessários. Antes do lançamento desse recurso, o Amplify fornecia uma configuração de instância de compilação de tamanho fixo de 8 GiB de memória e 4 v. CPUs

O Amplify oferece suporte a três tipos de instância de compilação: Standard, Large e XLarge. Se você não especificar um tipo de instância, o Amplify usará a instância padrão Standard. Você pode configurar o tipo de instância de compilação para um aplicativo usando o console do Amplify AWS CLI, o ou o. SDKs

O custo de cada tipo de instância de compilação é calculado por minuto de compilação. Para obter detalhes sobre os preços, consulte Preços do AWS Amplify.

A tabela a seguir descreve as especificações de computação para cada tipo de instância de compilação:

Tipo de instância de compilação	v CPUs	Memória	Espaço em disco
Standard	4 v CPUs	8 GiB	128 GB
Large	8 v CPUs	16 GiB	128 GB
XLarge	36 g CPUs	72 GiB	256 GB

Tópicos

Noções básicas dos tipos de instâncias de compilação

A configuração do tipo de instância de compilação é definida no nível da aplicação e se estende a todas as ramificações da aplicação. Os detalhes principais a seguir se aplicam aos tipos de instância de compilação:

- O tipo de instância de compilação que você configura para uma aplicação se aplica automaticamente às ramificações criadas automaticamente e às visualizações de solicitação de pull.
- A cota do serviço de trabalhos simultâneos se aplica a todos os tipos de instância de compilação em seu. Conta da AWS Por exemplo, se seu limite de Trabalhos simultâneos for cinco, será possível executar no máximo 5 compilações em todos os tipos de instância da sua Conta da AWS.
- O custo de cada tipo de instância de compilação é calculado por minuto de compilação. O processo de alocação de instâncias de compilação pode exigir mais tempo adicional antes do início da compilação. Especialmente para instâncias maiores XLarge, sua compilação pode apresentar latência antes do início da compilação, devido a esse tempo de sobrecarga. No entanto, você será cobrado somente pelo tempo real de construção, não pelo tempo adicional.

É possível configurar o tipo de instância de compilação ao criar uma nova aplicação ou atualizar o tipo de instância em uma aplicação existente. Para obter instruções sobre como definir essa configuração no console do Amplify, consulte Configuração do tipo de instância de compilação no console do Amplify. Você também pode atualizar essa configuração usando SDKs o. Para obter mais informações, consulte e UpdateApp APIs na Referência da API Amplify. CreateApp

Se você tem aplicações existentes em sua conta que foram criadas antes do lançamento do recurso de tipo de instância de compilação personalizável, elas estão usando o tipo de instância Standard padrão. Quando você atualiza o tipo de instância de compilação de uma aplicação existente, todas as compilações que estiverem na fila ou em andamento antes da atualização utilizarão o tipo de instância de compilação configurado anteriormente. Por exemplo, se você tiver uma aplicação existente com a ramificação `main` implantada no Amplify e atualizar seu tipo de instância de compilação de Standard para Large, todas as novas compilações que você iniciar a partir da ramificação `main` usarão o tipo de instância de compilação Large. No entanto, todas as compilações em andamento no momento em que você atualiza o tipo de instância de compilação continuarão sendo executadas na instância Standard.

Configuração do tipo de instância de compilação no console do Amplify

Siga o procedimento a seguir para configurar o tipo de instância de compilação ao criar uma nova aplicação do Amplify.

Para configurar o tipo de instância de compilação de uma nova aplicação

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todas as aplicações, escolha Criar nova aplicação.
3. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
4. Na página Adicionar ramificação do repositório, faça o seguinte:
   a. Na lista Repositórios atualizados recentemente, selecione o nome do repositório a ser conectado.
   b. Na lista Ramificação, selecione o nome da ramificação do repositório a ser conectada.
   c. Escolha Próximo.
5. Na página Configurações da aplicação, abra a seção Configurações avançadas.
6. Em Tipo de instância de compilação, escolha o tipo de instância desejada na lista.
7. Se você estiver implantando uma aplicação baseada em runtime do Node.js, configure o tamanho da memória heap para utilizar efetivamente um tipo de instância grande. É possível fazer isso na página Configurações da aplicação definindo uma variável de ambiente ou atualizando as configurações de compilação. Para obter mais informações, consulte Configuração da memória heap de uma aplicação para utilizar tipos de instâncias grandes.
   - Definição de uma variável de ambiente
     a. Na seção Configurações avançadas, Variáveis de ambiente, escolha Adicionar nova.
     b. Em Chave, insira **`NODE_OPTIONS`**.
     c. Em Valor, insira `--max-old-space-size=`*`memory_size_in_mb`*. *`memory_size_in_mb`*Substitua pelo tamanho de memória de pilha desejado em megabytes.
   - Atualização das configurações de compilação
     a. Na seção Configurações de compilação, escolha Editar arquivo YML.
     b. Execute o comando a seguir à fase `preBuild`. *`memory_size_in_mb`*Substitua pelo tamanho de memória de pilha desejado em megabytes.

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

     c. Escolha Salvar.

8. Escolha Próximo.
9. Na página Revisar, escolha Salvar e implantar.

Siga o procedimento a seguir para configurar o tipo de instância de compilação para uma aplicação existente do Amplify.

Para configurar o tipo de instância de compilação de uma aplicação existente

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha a aplicação para a qual você quer configurar o tipo de instância de compilação.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Configurações de compilação.
4. Na página Configurações da compilação, na seção Configurações avançadas, escolha Editar.
5. Na página Editar configurações, em Tipo de instância de compilação, escolha o tipo de instância desejada na lista.
6. Escolha Salvar. Essa alteração entrará em vigor na próxima vez que você implantar a aplicação.
7. (Opcional) Para implantar a aplicação atualizada imediatamente, faça o seguinte:

    a. No painel de navegação, selecione Visão geral.
    b. Na página de visão geral da sua aplicação, escolha a ramificação a ser reimplantada.
    c. Na página Implantação, escolha uma implantação, como a implantação mais recente. Em seguida, escolha Reimplantar esta versão. Uma nova implantação se iniciará.
    d. Quando a implantação for concluída, as configurações de compilação da aplicação mostrarão que a ramificação está usando o tipo de instância de compilação atualizado.

Configuração da memória heap de uma aplicação para utilizar tipos de instâncias grandes

Se você estiver criando aplicações com uso intenso de memória, use esta seção para entender como configurar sua aplicação para utilizar tipos de instâncias grandes. As linguagens e estruturas de programação geralmente dependem da alocação de memória dinâmica, também conhecida como memória heap, durante o runtime para gerenciar os requisitos de memória da aplicação. A memória heap é solicitada pelo ambiente de runtime e alocada pelo sistema operacional host. Por padrão, os ambientes de runtime impõem um limite máximo de tamanho de heap disponível para a aplicação. Isso significa que nenhuma memória adicional estará disponível para a aplicação além do tamanho

do heap, mesmo que o sistema operacional ou contêiner do host tenha uma quantidade maior de memória disponível.

Como exemplo, o ambiente de execução do JavaScript Node.js v8 impõe um limite de tamanho de pilha padrão que depende de vários fatores, incluindo o tamanho da memória do host. Como resultado, as instâncias de compilação Standard e Large têm um tamanho de heap Node.js padrão de 2096 MB e a instância XLarge tem um tamanho de heap padrão de 4144 MB. Portanto, criar um aplicativo com um requisito de memória de 6000 MB usando o tamanho de pilha padrão do Node.js em qualquer tipo de instância de compilação do Amplify resultará em uma falha na compilação devido a um erro. out-of-memory

Para contornar os limites padrão de memória do tamanho da heap do Node.js, utilize uma das opções a seguir:

- Defina a variável de ambiente `NODE_OPTIONS` em sua aplicação do Amplify com o valor `--max-old-space-size=`*`memory_size_in_mb`*. Em `memory_size_in_mb`, especifique o tamanho de memória heap desejado, em megabytes.

  Para instruções, consulte Configurar variáveis de ambiente.

- Adicione o comando a seguir à fase `preBuild` na especificação de compilação da sua aplicação do Amplify.

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

  É possível atualizar a especificação de compilação no console do Amplify ou no arquivo `amplify.yml` da sua aplicação no repositório do projeto. Para instruções, consulte Definição das configurações de compilação de uma aplicação do Amplify.

  No exemplo a seguir, a especificação de compilação do Amplify define um tamanho de memória heao do Node.js para 7000 MB para criar uma aplicação de frontend do React:

  ```
  version: 1
  frontend:
    phases:
      preBuild:
        commands:
          # Set the heap size to 7000 MB
          - export NODE_OPTIONS='--max-old-space-size=7000'
          # To check the heap size memory limit in MB
  ```

```
        - node -e "console.log('Total available heap size (MB):',
v8.getHeapStatistics().heap_size_limit / 1024 / 1024)"
        - npm ci --cache .npm --prefer-offline
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: build
    files:
      - '**/*'
  cache:
    paths:
      - .npm/**/*
```

Para utilizar tipos de instância grandes de forma eficaz, é importante ter um tamanho de memória heap suficiente configurado. A configuração de um pequeno tamanho de heao para uma aplicação com uso intenso de memória provavelmente resultará em uma falha de compilação. Os registros de compilação do aplicativo podem não indicar diretamente um out-of-memory erro, pois o tempo de execução do aplicativo pode falhar inesperadamente. Configurar um tamanho de heap tão grande quanto a memória do host pode fazer com que o sistema operacional do host alterne ou encerre outros processos e, potencialmente, interrompa seu processo de compilação. Como referência, o Node.js recomenda definir um tamanho máximo de heap de 1536 MB em uma máquina com aproximadamente 2000 MB de memória para deixar alguma memória para outros usos.

O tamanho ideal do heap depende das necessidades da sua aplicação e do uso de recursos. Se você encontrar out-of-memory erros, comece com um tamanho de pilha moderado e aumente gradualmente conforme necessário. Como diretriz, recomendamos começar com 6.000 MB para um tipo de instância Standard, 12.000 MB para um tipo de instância Large e 60.000 MB para um tipo de instância XLarge.

Criação de um webhook de entrada para iniciar uma compilação

Configure um webhook de entrada no console do Amplify para iniciar uma compilação sem confirmar o código no seu repositório Git. É possível usar webhooks com ferramentas de CMS descentralizado (como o Contentful ou o GraphCMS) para iniciar uma compilação sempre que o conteúdo for alterado ou para executar compilações diárias usando serviços como Zapier.

Para criar um webhook de entrada

1. Faça login Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual você deseja criar um webhook.
3. No painel de navegação, escolha Hospedagem, e, em seguida, Configurações de compilação.
4. Na página Configurações de compilação, role para baixo até a seção Webhooks de entrada e escolha Criar webhook.
5. Na caixa de diálogo Criar webhook, faça o seguinte:
   a. Em Nome do webhook, insira um nome para o webhook.
   b. Para Ramificação para compilação, selecione a ramificação a ser criada com base nas solicitações de webhook recebidas.
   c. Escolha Criar webhook.
6. Na seção Webhooks de entrada, execute uma das seguintes ações:
   - Copie o URL do webhook e forneça-o a uma ferramenta de CMS descentralizado ou outro serviço para iniciar as compilações.
   - Execute o comando curl em uma janela de terminal para iniciar uma nova compilação.

Configuração de notificações por e-mail para compilações

Você pode configurar notificações por e-mail para um AWS Amplify aplicativo para alertar as partes interessadas ou membros da equipe quando uma construção for bem-sucedida ou falhar. Amplify Hosting cria um tópico do Amazon Simple Notification Service (SNS) na sua conta e o usa para configurar notificações por e-mail. As notificações podem ser configuradas para serem aplicadas a todas as filiais ou ramificações específicas de um aplicativo Amplify.

Configuração de notificações por e-mail

Use os procedimentos a seguir para configurar notificações por e-mail para todas as filiais ou filiais específicas de um aplicativo Amplify.

Para configurar notificações por e-mail para um aplicativo do Amplify

1. Faça login Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual você deseja configurar as notificações por e-mail.

3. No painel de navegação, escolha Hospedagem, Notificações de compilação. Na página Notificações de compilação, selecione Gerenciar notificações.
4. Na página Gerenciar notificações, escolha Adicionar nova.
5. Execute um destes procedimentos:
   - Para enviar notificações para uma única filial, em E-mail, insira o endereço de e-mail para o qual enviar notificações. Em Filial, selecione o nome da filial para a qual enviar notificações.
   - Para enviar notificações para todas as filiais conectadas, em E-mail, insira o endereço de e-mail para o qual enviar notificações. Em Filial, escolha Todas as filiais.
6. Escolha Salvar.

Conexão de um domínio personalizado

É possível conectar um aplicativo que você implantou com o Amplify Hosting a um domínio personalizado. Quando você usa o Amplify para implantar sua aplicação da Web, o Amplify a hospeda para você no domínio padrão `amplifyapp.com` com um URL tal como `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`. Quando você conecta seu aplicativo a um domínio personalizado, os usuários veem que seu aplicativo está hospedado em um URL personalizado, como `https://www.example.com`.

Você pode comprar um domínio personalizado por meio de um registrador de domínio credenciado, como Amazon Route 53 ou GoDaddy. O Route 53 é o web service de Sistema de Nomes de Domínio (DNS) da Amazon. Para obter mais informações sobre o uso do Route 53, consulte O que é o Amazon Route 53. Para obter uma lista de registradores de domínios credenciados terceirizados, consulte o Diretório de registradores credenciados no site da ICANN.

Ao configurar seu domínio personalizado, é possível usar o certificado gerenciado padrão que o Amplify provisiona para você ou seu próprio certificado personalizado. É possível alterar o certificado usado pelo domínio a qualquer momento. Para obter informações detalhadas sobre o gerenciamento de certificados, consulte Usando SSL/TLS certificados.

Antes de continuar com a configuração de um domínio personalizado, verifique se você cumpriu os pré-requisitos descritos abaixo.

- Você possui um nome de domínio registrado.
- Você tem um certificado emitido ou importado para AWS Certificate Manager.
- Você implantou sua aplicação no Amplify Hosting.

  Para obter mais informações sobre a conclusão dessa etapa, consulte Noções básicas da implantação de uma aplicação no Amplify Hosting.
- Você tem conhecimento básico de domínios e terminologia de DNS.

  Para obter mais informações sobre domínios e DNS, consulte Entender a terminologia e os conceitos do DNS.

> **Warning**
>
> Ao iniciar uma DomainAssociation solicitação para um aplicativo do Amplify com um domínio que já está ou foi anteriormente associado a diferentes aplicativos do Amplify em outras contas da AWS na mesma região, isso é considerado uma associação de domínio entre contas. Solicitações de associação de domínio entre contas exigem verificação manual. Se você quiser continuar com uma associação de domínio entre contas, entre em contato com o suporte da AWS para obter ajuda.

Tópicos

Entender a terminologia e os conceitos do DNS

Se você não estiver familiarizado com os termos e conceitos associados ao Sistema de Nomes de Domínio (DNS), os tópicos a seguir podem ajudá-lo a entender os procedimentos para adicionar domínios personalizados.

Terminologia DNS

Veja a seguir uma lista de termos comuns ao DNS. Eles podem ajudar você a entender os procedimentos para adicionar domínios personalizados.

CNAME

Um nome de registro canônico (CNAME) é um tipo de registro DNS que mascara o domínio para um conjunto de páginas da web e faz com que pareçam estar localizadas em outro lugar. Um CNAME aponta para um subdomínio para um nome de domínio totalmente qualificado (FQDN). Por exemplo, é possível criar registro CNAME para mapear o subdomínio www.example.com, em que www é o subdomínio, para o domínio FQDN branch-name.d1m7bkiki6tdw1.cloudfront.net atribuído ao seu aplicativo no console do Amplify.

ANAME

Um registro ANAME é como um registro CNAME, mas no nível raiz. Um ANAME aponta à raiz do seu domínio para um FQDN. Esse FQDN aponta para um endereço IP.

Servidor de nomes

Um servidor de nome é um servidor na Internet que é especializado no tratamento de consultas sobre a localização de vários serviços de um nome de domínio. Se você configurar seu domínio no Amazon Route 53, uma lista de servidores de nomes já está atribuída ao seu domínio.

Registro NS

Um registro NS aponta para servidores de nomes que pesquisam os detalhes do seu domínio.

Verificação de DNS

Um Sistema de Nomes de Domínio (DNS) é como uma lista telefônica que traduz nomes de domínio legíveis por humanos em endereços IP fáceis de usar no computador. Quando você digita **`https://google.com`** em um navegador, uma operação de pesquisa é executada no provedor de DNS para encontrar o endereço IP do servidor que hospeda o site.

Os provedores de DNS contêm registros de domínios e seus endereços IP correspondentes. Os registros DNS mais usados são os registros CNAME, ANAME e NS.

O Amplify usa um registro CNAME para verificar se você possui o domínio personalizado. Se você hospedar seu domínio com o Route 53, a verificação será feita em seu nome. No entanto, se você hospedar seu domínio com um provedor terceirizado GoDaddy, como, precisará atualizar manualmente as configurações de DNS do seu domínio e adicionar um novo registro CNAME fornecido pelo Amplify.

Processo de ativação de domínios personalizados

> **Warning**
>
> Ao iniciar uma DomainAssociation solicitação para um aplicativo do Amplify com um domínio que já está ou foi anteriormente associado a diferentes aplicativos do Amplify em outras contas da AWS na mesma região, isso é considerado uma associação de domínio entre contas. Solicitações de associação de domínio entre contas exigem verificação manual. Se você quiser continuar com uma associação de domínio entre contas, entre em contato com o suporte da AWS para obter ajuda.

Quando você conecta sua aplicação do Amplify a um domínio personalizado no console do Amplify, há várias etapas que o Amplify deve concluir antes que você possa visualizar sua aplicação usando seu domínio personalizado. A lista a seguir descreve cada etapa do processo de configuração e ativação do domínio.

Criação de SSL/TLS

: Se você estiver usando um certificado gerenciado, AWS Amplify emita um certificado SSL/TLS para configurar um domínio personalizado seguro.

Configuração e verificação SSL/TLS

: Antes de emitir um certificado gerenciado, o Amplify verifica se você é o proprietário do domínio. Para domínios gerenciados pelo Amazon Route 53, o Amplify atualiza automaticamente a verificação de registros DNS. Para domínios gerenciados fora do Route 53, será necessário adicionar manualmente o registro de verificação de DNS fornecido pelo Amplify no console do Amplify em seu domínio com um provedor de DNS terceirizado.

  Se estiver usando um certificado personalizado, você será responsável por validar a propriedade do domínio.

Ativação do domínio

: O domínio foi verificado com sucesso. Para domínios gerenciados fora do Route 53, será necessário adicionar manualmente os registros CNAME fornecidos no console do Amplify em seu domínio com um provedor de DNS terceirizado.

Usando SSL/TLS certificados

Um SSL/TLS certificado é um documento digital que permite que os navegadores da Web identifiquem e estabeleçam conexões de rede criptografadas com sites usando o SSL/TLS protocolo seguro. Ao configurar seu domínio personalizado, é possível usar o certificado gerenciado padrão que o Amplify provisiona para você ou seu próprio certificado personalizado.

Com um certificado gerenciado, o Amplify emite um SSL/TLS certificado para todos os domínios conectados ao seu aplicativo para que todo o tráfego seja protegido por meio de HTTPS/2. O certificado padrão gerado pelo AWS Certificate Manager (ACM) é válido por 13 meses e é renovado automaticamente, desde que seu aplicativo esteja hospedado no Amplify.

> **Warning**
>
> O Amplify não poderá renovar o certificado se o registro de verificação CNAME tiver sido modificado ou excluído nas configurações de DNS com seu provedor de domínio. É necessário excluir e adicionar o domínio novamente no console do Amplify.

Para usar um certificado personalizado, é necessário primeiro obter um certificado da autoridade de certificação terceirizada de sua escolha. O Amplify Hosting oferece suporte a dois tipos de certificados: RSA (Rivest-Shamir-Adleman) e ECDSA (Elliptic Curve Digital Signature Algorithm). Cada tipo de certificado deve estar de acordo com os requisitos descritos a seguir.

Certificados RSA

- O Amplify Hosting oferece suporte a chaves RSA de 1.024 bits, 2.048 bits, 3.072 bits e 4.096 bits.
- AWS Certificate Manager (ACM) emite certificados RSA com chaves de até 2048 bits.
- Para usar um certificado RSA de 3.072 bits ou 4.096 bits, obtenha o certificado externamente e importe-o para o ACM. Ele estará então disponível para uso com o Amplify Hosting.

Certificados ECDSA

- O Amplify Hosting oferece suporte a chaves de 256 bits.
- Use a curva elíptica prime256v1 para obter um certificado ECDSA para o Amplify Hosting.

Depois de obter um certificado, importe-o para AWS Certificate Manager o. O ACM é um serviço que permite provisionar, gerenciar e implantar facilmente SSL/TLS certificados públicos e privados para uso com Serviços da AWS seus recursos internos conectados. Certifique-se de solicitar ou importar o certificado na região Leste dos EUA (Norte da Virgínia) (us-east-1).

Certifique-se de que seu certificado personalizado cubra todos os subdomínios que você planeja adicionar. É possível usar um caractere curinga no início do seu nome de domínio para cobrir vários subdomínios. Por exemplo, se o seu domínio for `example.com`, é possível incluir o domínio curinga `*.example.com`. Isso abrangerá subdomínios como `product.example.com` e `api.example.com`.

Depois que seu certificado personalizado estiver disponível no ACM, será possível selecioná-lo durante o processo de configuração do domínio. Para obter mais informações sobre a importação de certificados no AWS Certificate Manager, consulte Importação de certificados no AWS Certificate Manager no Guia do usuário do AWS Certificate Manager .

Se você renovar ou reimportar seu certificado personalizado no ACM, o Amplify atualizará os dados do certificado associados ao seu domínio personalizado. No caso de certificados importados, o ACM não gerenciará as renovações automaticamente. Você é responsável por renovar seus certificados personalizados e importá-los novamente.

É possível alterar o certificado usado por um domínio a qualquer momento. Por exemplo, é possível mudar do certificado gerenciado padrão para um certificado personalizado ou mudar de um certificado personalizado para um certificado gerenciado. Além disso, é possível alterar o certificado personalizado em uso para um certificado personalizado diferente. Para obter instruções sobre como atualizar certificados, consulte Atualizar o SSL/TLS certificado de um domínio.

Adição de um domínio personalizado gerenciado pelo Amazon Route 53

O Amazon Route 53 é um serviço de DNS altamente disponível e escalável. Para obter mais informações, consulte Amazon Route 53 no Guia do desenvolvedor do Amazon Route 53. Se você já possuir um domínio do Route 53, use as instruções a seguir para conectar seu domínio personalizado à sua aplicação do Amplify.

Para adicionar um domínio personalizado gerenciado pelo Route 53

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.

2. Escolha o aplicativo ao qual você deseja conectar a um domínio personalizado.
3. No painel de navegação, escolha Hospedagem, Domínios personalizados.
4. Na página Domínios personalizados, escolha Adicionar domínio.
5. Insira o nome do seu domínio raiz. Por exemplo, se o nome do seu domínio for https://example.com, insira**`example.com`**.

   Quando você começa a digitar, todos os domínios raiz que você já gerencia no Route 53 aparecem na lista. É possível escolher o domínio ao qual deseja conectar-se na lista. Se você ainda não possui o domínio e ele está disponível, é possível comprar o domínio no Amazon Route 53.
6. Depois de inserir seu nome de domínio, escolha Configurar domínio.
7. Por padrão, o Amplify cria automaticamente duas entradas de subdomínio para seu domínio. Por exemplo, se seu nome de domínio for exemplo.com, você verá os subdomínios https://www.example.come https://example.comum redirecionamento configurado do domínio raiz para o subdomínio www.

   (Opcional) É possível modificar a configuração padrão se quiser adicionar apenas subdomínios. Para alterar a configuração padrão, escolha Regravações e redirecionamentos no painel de navegação, e, em seguida, configure seu domínio.
8. Escolha o SSL/TLS certificado a ser usado. Você pode usar o certificado gerenciado padrão que o Amplify provisiona para você ou um certificado personalizado de terceiros para o qual você importou. AWS Certificate Manager
   - Use o certificado gerenciado padrão do Amplify.
     - Escolha Certificado gerenciado do Amplify.
   - Use um certificado personalizado terceirizado.
     a. Escolha Certificado SSL personalizado.
     b. Selecione o certificado a ser usado na lista.
9. Escolha Adicionar domínio.

> **Note**
>
> Pode levar até 24 horas para o DNS propagar e emitir o certificado. Para obter ajuda na resolução de erros que ocorrem, consulte Solucionar problemas de domínios personalizados.

Adição de um domínio personalizado gerenciado por um provedor de DNS terceirizado

Se você não estiver usando o Amazon Route 53 para gerenciar seu domínio, é possível adicionar um domínio personalizado gerenciado por um provedor de DNS terceirizado ao seu aplicativo implantado com o Amplify.

Se você estiver usando GoDaddy, consulte the section called “Atualizando registros DNS para um domínio gerenciado pelo GoDaddy” para obter instruções específicas para esse provedor.

Adicionar um domínio personalizado gerenciado por um provedor DNS de terceiros

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo ao qual você deseja adicionar um domínio personalizado.
3. No painel de navegação, escolha Hospedagem, Domínios personalizados.
4. Na página Domínios personalizados, escolha Adicionar domínio.
5. Insira o nome do seu domínio raiz. Por exemplo, se o nome do seu domínio for https://example.com, insira**`example.com`**.
6. O Amplify detecta que você não está usando um domínio do Route 53 e oferece a opção de criar uma zona hospedada no Route 53.
   - Para criar uma zona hospedada no Route 53
     a. Escolha Criar zona hospedada no Route 53.
     b. Escolha Configurar domínio.
     c. Os servidores de nomes de zonas hospedadas são exibidos no console. Vá ao site do seu provedor de DNS e adicione os servidores de nomes às suas configurações de DNS.
     d. Selecione Eu adicionei os servidores de nomes acima ao meu registro de domínio.

e. Prossiga para a etapa sete.

- Para continuar com a configuração manual

  a. Escolha Configuração manual

  b. Escolha Configurar domínio.

  c. Prossiga para a etapa sete.

7. Por padrão, o Amplify cria automaticamente duas entradas de subdomínio para seu domínio. Por exemplo, se seu nome de domínio for exemplo.com, você verá os subdomínios https://www.example.come https://example.comum redirecionamento configurado do domínio raiz para o subdomínio www.

   (Opcional) É possível modificar a configuração padrão se quiser adicionar apenas subdomínios. Para alterar a configuração padrão, escolha Regravações e redirecionamentos no painel de navegação e configure seu domínio.

8. Escolha o SSL/TLS certificado a ser usado. Você pode usar o certificado gerenciado padrão que o Amplify provisiona para você ou um certificado personalizado de terceiros para o qual você importou. AWS Certificate Manager

   - Use o certificado gerenciado padrão do Amplify.

     - Escolha Certificado gerenciado do Amplify.

   - Use um certificado personalizado terceirizado.

     a. Escolha Certificado SSL personalizado.

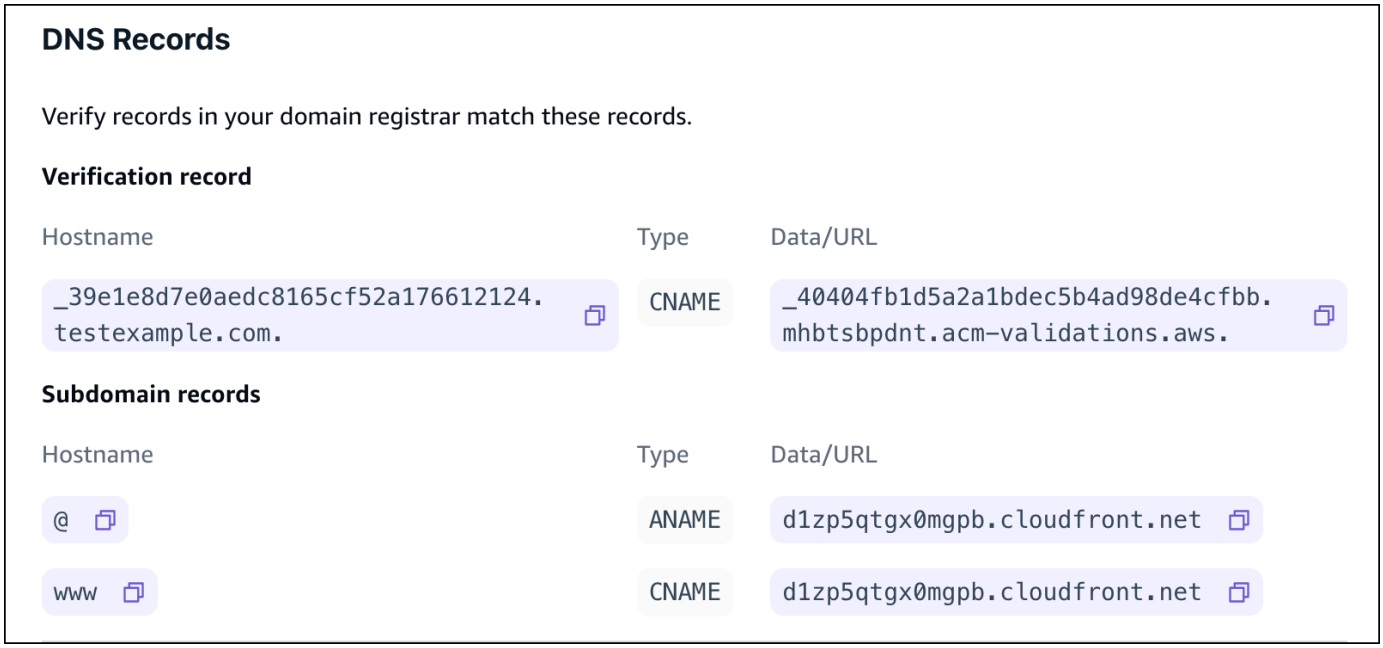
     b. Selecione o certificado a ser usado na lista.

9. Escolha Adicionar domínio.

10. Se você escolheu Criar zona hospedada no Route 53 na etapa seis, vá para a etapa 15.

    Se você escolheu Configuração manual na etapa seis, será necessário atualizar seus registros de DNS com seu provedor de domínio terceirizado.

    No menu Ações, escolha Exibir registros DNS. A captura de tela a seguir mostra os registros de DNS exibidos no console.



11. Execute um destes procedimentos:

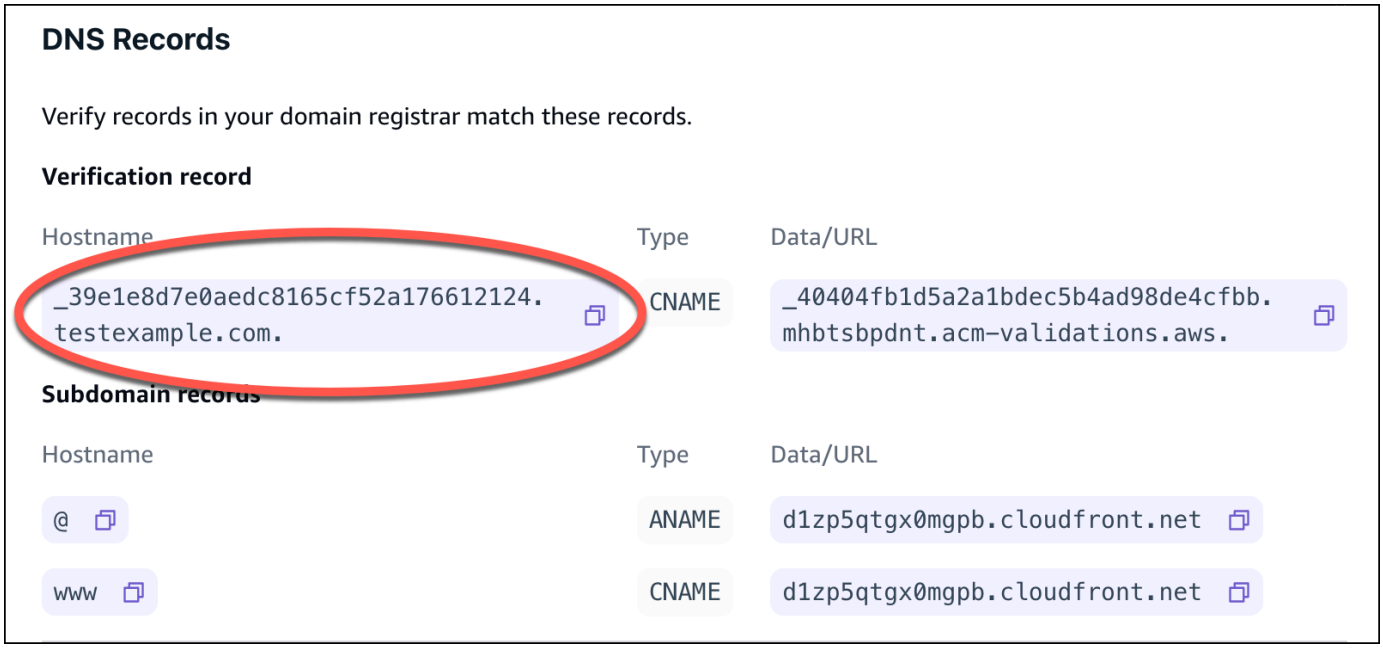
    - Se você estiver usando GoDaddy, acesseAtualizando registros DNS para um domínio gerenciado pelo GoDaddy.
    - Se você estiver usando um provedor de DNS terceirizado diferente, vá para a próxima etapa deste procedimento.

12. Acesse o site do seu provedor de DNS, faça login na sua conta e localize as configurações de gerenciamento de DNS do seu domínio. Você configurará dois registros CNAME.

13. Configure o primeiro registro CNAME para apontar seu subdomínio para o servidor de AWS validação.

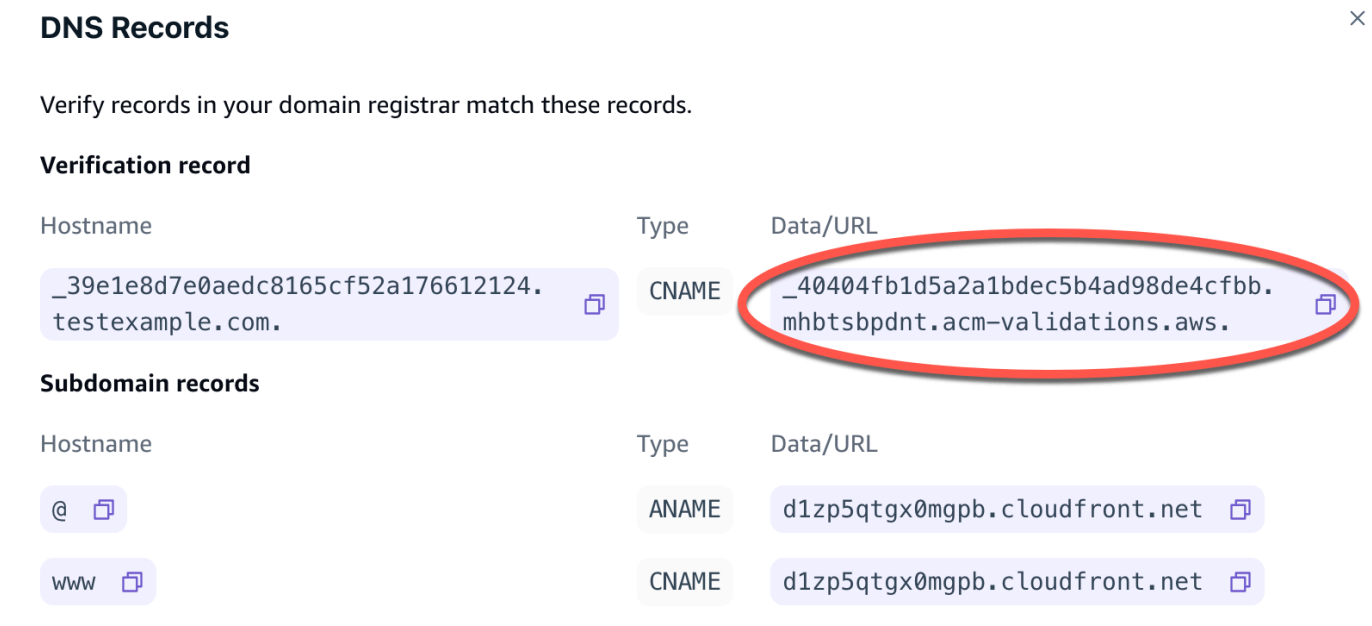
    Se o console do Amplify exibir um registro de DNS para verificar a propriedade do seu subdomínio, como _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, insira somente **`_c3e2d7eaf1e656b73f46cd6980fdc0e`** como nome do subdomínio do registro CNAME.

    A captura de tela a seguir mostra a localização do registro de verificação a ser usado.



Se o console do Amplify exibir um registro do servidor de validação do ACM, como _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, insira **`_cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`** como valor do registro CNAME.

A captura de tela a seguir mostra a localização do registro de verificação do ACM a ser usado.



O Amplify usa essas informações para verificar a propriedade do seu domínio e gerar um SSL/TLS certificado para seu domínio. Depois que o Amplify validar a propriedade do seu domínio, todo o tráfego será servido usando HTTPS/2.

> **Note**
>
> O certificado padrão do Amplify gerado pelo AWS Certificate Manager (ACM) é válido por 13 meses e é renovado automaticamente, desde que seu aplicativo esteja hospedado com o Amplify. O Amplify não pode renovar o certificado se o registro de verificação CNAME tiver sido modificado ou excluído. É necessário excluir e adicionar o domínio novamente no console do Amplify.

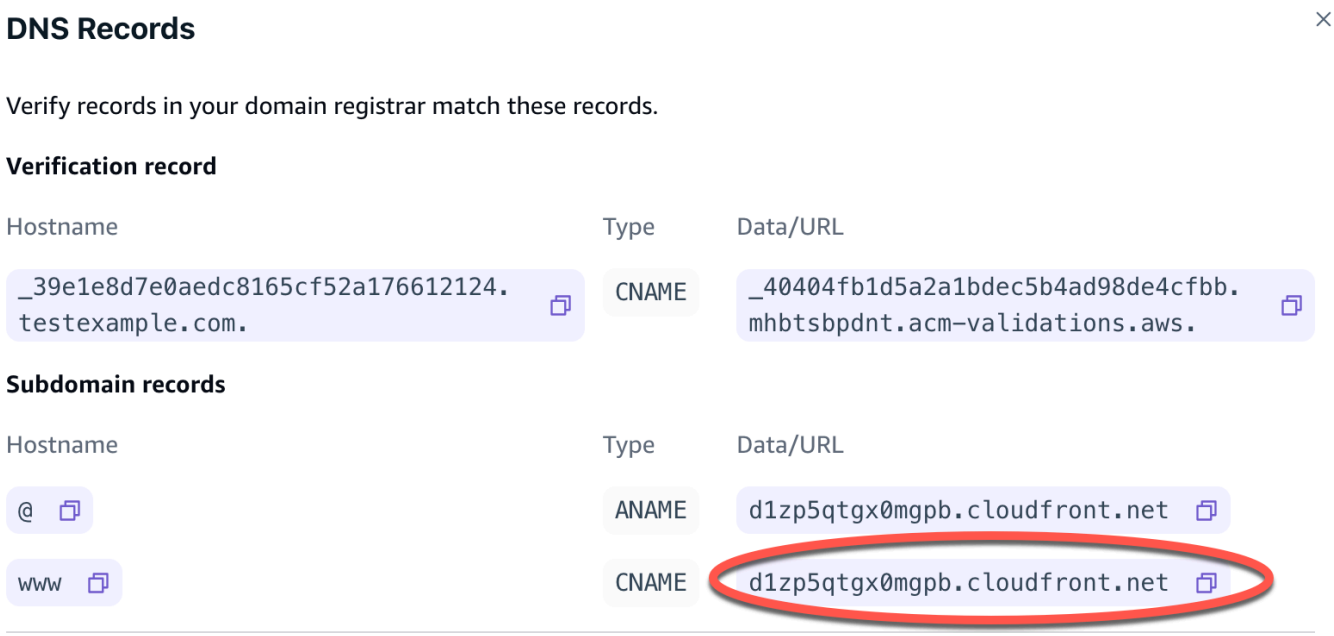
> **Important**
>
> É importante que você execute essa etapa logo após adicionar seu domínio personalizado no console do Amplify. O AWS Certificate Manager (ACM) começa imediatamente a tentar verificar a propriedade. Com o tempo, as verificações se tornam menos frequentes. Se você adicionar ou atualizar seus registros CNAME algumas horas depois de criar seu aplicativo, isso pode fazer com que seu aplicativo fique preso no estado de verificação pendente.

14. Configure um segundo registro CNAME para apontar seus subdomínios para o domínio Amplify. Por exemplo, se seu subdomínio for www.example.com, insira www como nome do subdomínio.

    Se o console do Amplify exibir o domínio da sua aplicação como d111111abcdef8.cloudfront.net, insira **`d111111abcdef8.cloudfront.net`** como domínio do Amplify.

    Se você tiver tráfego de produção, é recomendável atualizar o registro CNAME depois que o status do domínio mostrar AVAILABLE no console do Amplify.

    A captura de tela a seguir mostra a localização do registro de nome de domínio a ser usado.



15. Configure o ANAME/ALIAS registro para apontar para o domínio raiz do seu aplicativo (por exemplo https://example.com). Um registro ANAME permite apontar a raiz do domínio para um nome de host. Se você tiver tráfego de produção, é recomendável atualizar o registro ANAME depois que o status do domínio mostrar AVAILABLE no console. Para provedores de DNS que não têm ANAME/ALIAS suporte, é altamente recomendável migrar seu DNS para o Route 53. Para obter mais informações, consulte Como configurar o Amazon Route 53 como seu serviço de DNS.

> **Note**
>
> A verificação de propriedade de domínio e a propagação de DNS para domínios de terceiros pode levar até 48 horas. Para obter ajuda na resolução de erros que ocorrem, consulte Solução de problemas de domínios personalizados.

Atualizando registros DNS para um domínio gerenciado pelo GoDaddy

Se GoDaddy for seu provedor de DNS, use as instruções a seguir para atualizar seus registros DNS na GoDaddy interface do usuário e concluir a conexão do aplicativo Amplify ao seu domínio. GoDaddy

Para adicionar um domínio personalizado gerenciado por GoDaddy

1. Antes de atualizar seus registros DNS com GoDaddy, conclua as etapas de um a nove do procedimentothe section called “Adição de um domínio personalizado gerenciado por um provedor de DNS terceirizado”.
2. Faça login na sua GoDaddy conta.
3. Na sua lista de domínios, encontre o domínio a ser adicionado e escolha Gerenciar DNS.
4. Na página DNS, GoDaddy exibe uma lista de registros do seu domínio na seção Registros DNS. Você precisa adicionar dois novos registros CNAME.
5. Crie o primeiro registro CNAME para direcionar seus subdomínios para o domínio Amplify.

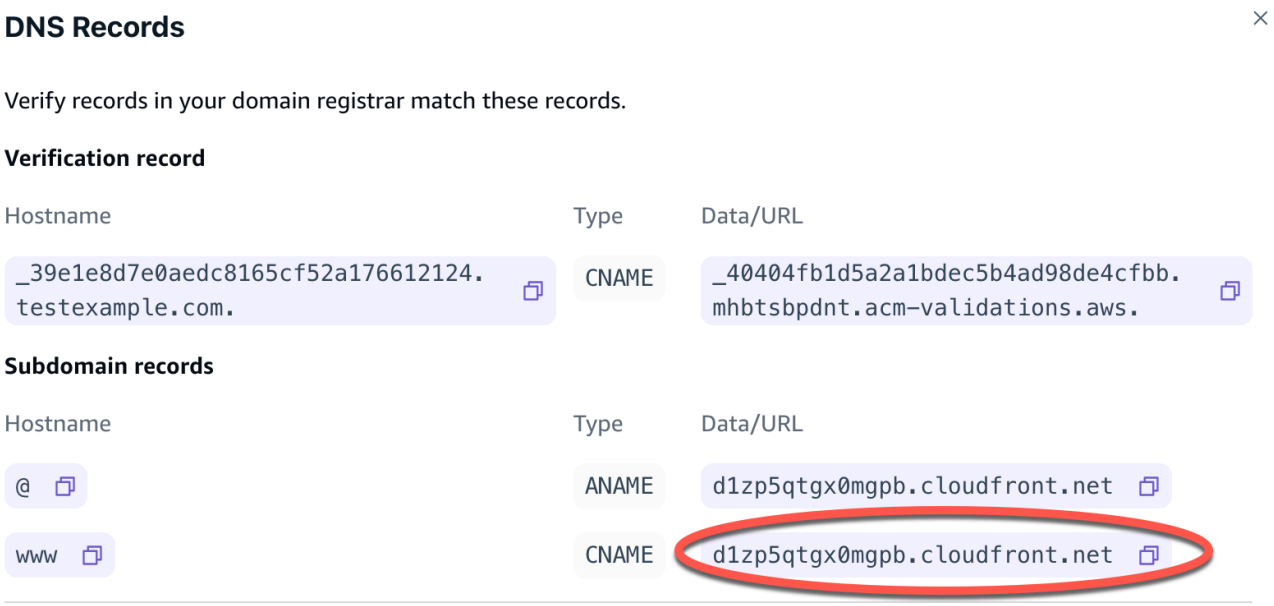
   a. Na seção Registros de DNS, escolha Adicionar novo registro.

   b. Para Tipo, escolha CNAME.

   c. Em Nome, insira somente o subdomínio. Por exemplo, se seu subdomínio for www.exemplo.com, insira www em Nome.

   d. Em Value, veja seus registros DNS no console do Amplify e insira o valor. Se o console do Amplify exibir o domínio da sua aplicação como d111111abcdef8.cloudfront.net, insira **`d111111abcdef8.cloudfront.net`** como valor.

      A captura de tela a seguir mostra a localização do registro de nome de domínio a ser usado.



   e. Escolha Salvar.

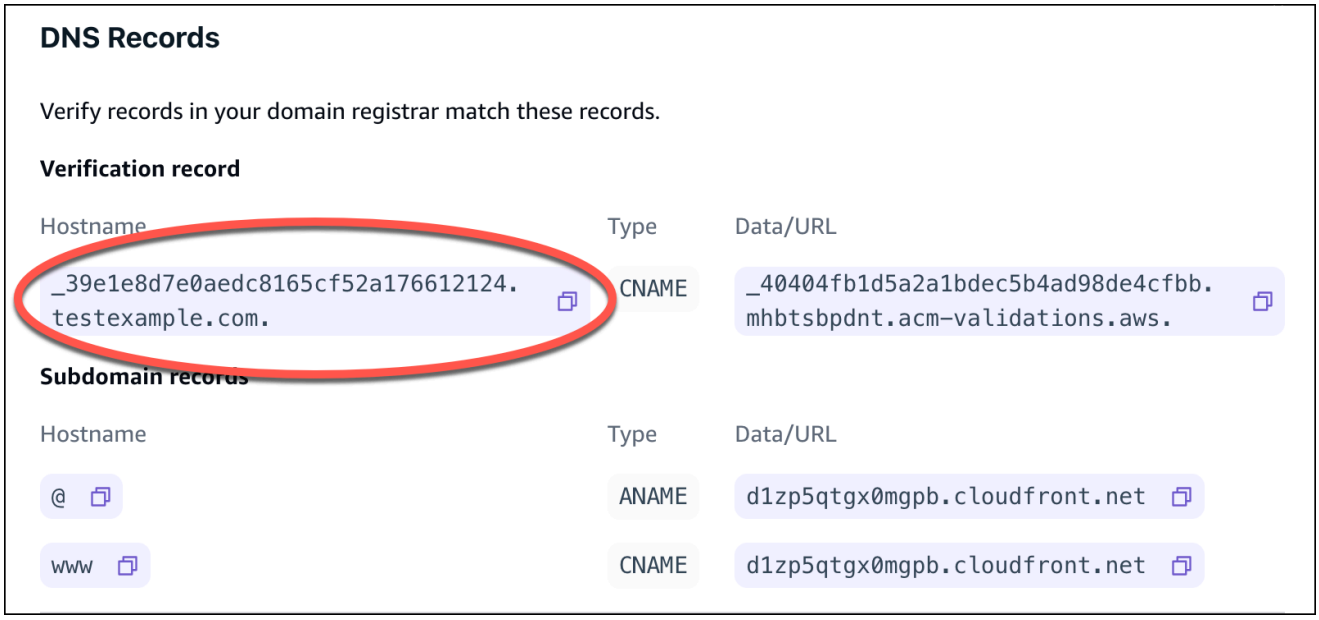
6. Crie o segundo registro CNAME para apontar para o servidor de validação AWS Certificate Manager (ACM). Um único ACM validado gera um SSL/TLS certificado para seu domínio.

   a. Para Tipo, escolha CNAME.

   b. Em Nome, insira o subdomínio.

      Por exemplo, se o registro DNS no console do Amplify para verificar a propriedade do seu subdomínio for _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, digite somente **`_c3e2d7eaf1e656b73f46cd6980fdc0e`** como Nome.

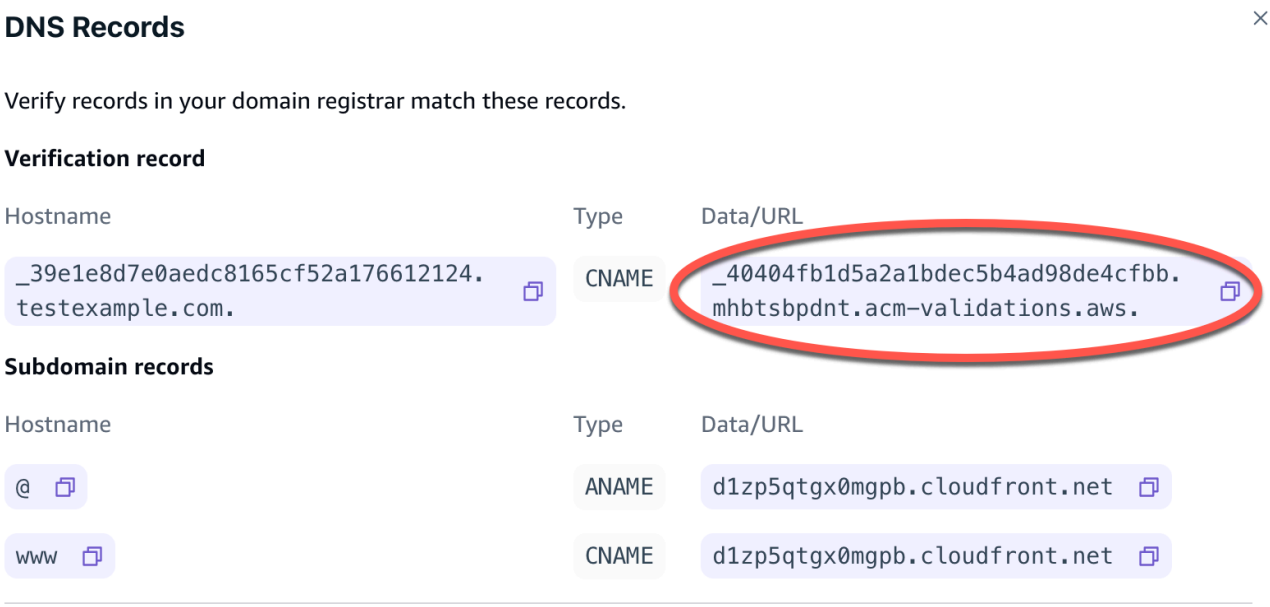
      A captura de tela a seguir mostra a localização do registro de verificação a ser usado.



   c. Em Valor, insira o certificado de validação do ACM.

      Por exemplo, se o servidor de validação for _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, digite _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws para Valor.

      A captura de tela a seguir mostra a localização do registro de verificação do ACM a ser usado.



d. Escolha Salvar.

> **Note**
>
> O certificado padrão do Amplify gerado pelo AWS Certificate Manager (ACM) é válido por 13 meses e é renovado automaticamente, desde que seu aplicativo esteja hospedado com o Amplify. O Amplify não pode renovar o certificado se o registro de verificação CNAME tiver sido modificado ou excluído. É necessário excluir e adicionar o domínio novamente no console do Amplify.

7. Essa etapa não é necessária para subdomínios. GoDaddy não oferece ANAME/ALIAS records. For DNS providers that do not have ANAME/ALIAS suporte, é altamente recomendável migrar seu DNS para o Amazon Route 53. Para obter mais informações, consulte Como configurar o Amazon Route 53 como seu serviço de DNS.

   Se você quiser se manter GoDaddy como seu provedor e atualizar o domínio raiz, adicione Encaminhamento e configure um encaminhamento de domínio:

   a. Na página DNS, localize o menu na parte superior da página e escolha Encaminhamento.
   b. Na seção Domínio, escolha Adicionar encaminhamento.
   c. Escolha http:// e, em seguida, insira o nome do seu subdomínio ao qual encaminhar (por exemplo, www.example.com) como URL de destino.
   d. Em Tipo de encaminhamento, escolha Temporário (302).

e. Escolha Salvar.

Atualizando o SSL/TLS certificado de um domínio

Você pode alterar o SSL/TLS certificado que está em uso para um domínio a qualquer momento. Por exemplo, é possível deixar de usar um certificado gerenciado para usar um certificado personalizado. Isso é útil se você quiser gerenciar o certificado e suas notificações de expiração. É possível alterar o certificado personalizado em uso pelo domínio. Fazer alterações no certificado SSL não causará nenhum tempo de inatividade para seu domínio ativo. Para obter mais informações sobre certificados, consulte Uso de certificados SSL/TLS.

Use o procedimento a seguir para atualizar o tipo de certificado ou o certificado personalizado que está sendo usado para um domínio.

Para atualizar um certificado de domínio

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha a aplicação que você deseja atualizar.
3. No painel de navegação, escolha Hospedagem, Domínios personalizados.
4. Na página Domínios personalizados, escolha Configuração de domínio.
5. Na página de detalhes do seu domínio, localize a seção Certificado SSL personalizado. O procedimento para atualizar seu certificado varia de acordo com o tipo de alteração que você deseja fazer.
   - Para mudar de um certificado personalizado para o certificado gerenciado padrão do Amplify
     - Escolha Certificado gerenciado do Amplify.
   - Para mudar de um certificado gerenciado para um certificado personalizado
     a. Escolha Certificado SSL personalizado.
     b. Selecione o certificado a ser usado na lista.
   - Para mudar um certificado personalizado para um certificado personalizado diferente
     - Em Certificado SSL personalizado, selecione o novo certificado a ser usado na lista.

6. Escolha Salvar. Os detalhes de status do domínio indicarão que o Amplify iniciou o processo de criação de SSL para um certificado gerenciado ou o processo de configuração para um certificado personalizado.

Gerenciar subdomínios

Um subdomínio é a parte do seu URL que aparece antes do nome do seu domínio. Por exemplo, www é o subdomínio de www.amazon.com e aws é o subdomínio de aws.amazon.com. Se você já tem um site de produção, talvez queira conectar apenas um subdomínio. Os subdomínios também podem ser multiníveis, por exemplo, beta.alpha.example.com tem o subdomínio multinível beta.alpha.

Para adicionar somente um subdomínio

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo ao qual você deseja adicionar um subdomínio.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Domínios personalizados.
4. Na página Domínios personalizados, escolha Adicionar domínio.
5. Insira o nome do seu domínio raiz e escolha Configurar domínio. Por exemplo, se o nome do seu domínio for https://example.com, digite example.com.
6. Escolha Excluir raiz e modifique o nome do subdomínio. Por exemplo, se o domínio for exemplo.com, é possível modificá-lo para adicionar somente o subdomínio alfa.
7. Escolha Adicionar domínio.

Para adicionar um subdomínio de vários níveis

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo ao qual você deseja adicionar um subdomínio de vários níveis.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Domínios personalizados.
4. Na página Domínios personalizados, escolha Adicionar domínio.
5. Insira o nome de um domínio com um subdomínio, escolha Excluir raiz e modifique o subdomínio para adicionar um novo nível.

Por exemplo, se você tiver um domínio chamado alpha.example.com e quiser criar um subdomínio de vários níveis beta.alpha.example.com, será necessário inserir beta como valor do subdomínio.

6. Escolha Adicionar domínio.

Para adicionar ou editar um subdomínio

Depois de adicionar um domínio personalizado a um aplicativo, é possível editar um subdomínio existente ou adicionar um novo subdomínio.

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual você deseja gerenciar subdomínios.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Domínios personalizados.
4. Na página Domínios personalizados, escolha Configuração de domínio.
5. Na seção Subdomínios, é possível editar seus subdomínios existentes conforme necessário.
6. (Opcional) Para adicionar um novo subdomínio, escolha Adicionar novo.
7. Escolha Salvar.

Configuração de subdomínios curinga

O Amplify Hosting agora oferece suporte a subdomínios curinga. Um subdomínio curinga é um subdomínio abrangente que permite direcionar subdomínios existentes e não existentes para uma ramificação específica do seu aplicativo. Ao usar um caractere curinga para associar todos os subdomínios em um aplicativo a uma ramificação específica, é possível veicular o mesmo conteúdo aos usuários do seu aplicativo em qualquer subdomínio e evitar configurar cada subdomínio individualmente.

Para criar um subdomínio curinga, especifique um asterisco (*) como nome do subdomínio. Por exemplo, se você especificar o subdomínio curinga `*.example.com` para uma ramificação específica do seu aplicativo, qualquer URL que termine com example.com será roteada para a ramificação. Nesse caso, as solicitações para `dev.example.com` e `prod.example.com` serão encaminhadas para o subdomínio `*.example.com`.

Observe que o Amplify oferece suporte a subdomínios curinga somente para um domínio personalizado. Você não pode usar esse atributo com o domínio padrão `amplifyapp.com`.

Os seguintes requisitos se aplicam aos subdomínios curinga:

- O nome do subdomínio deve ser especificado somente com um asterisco (*).
- Não é possível usar um curinga para substituir parte de um nome de subdomínio, como em *domain.example.com.
- Não é possível substituir um subdomínio no meio de um nome de domínio, como em subdomínio.*.exemplo.com.
- Por padrão, todos os certificados provisionados do Amplify abrangem todos os subdomínios de um domínio personalizado.

Para adicionar ou excluir um subdomínio curinga

Depois de adicionar um domínio personalizado a um aplicativo, é possível adicionar um subdomínio curinga para uma ramificação do aplicativo.

1. Faça login Console de gerenciamento da AWS e abra o console do Amplify Hosting.
2. Escolha o aplicativo para o qual você deseja gerenciar subdomínios curinga.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Domínios personalizados.
4. Na página Domínios personalizados, escolha Configuração de domínio.
5. Na seção Subdomínios, é possível adicionar ou excluir subdomínios curinga.
   - Para adicionar um novo subdomínio curinga
     a. Selecione Add new (Adicionar novo).
     b. Para o subdomínio, insira um *****.
     c. Para a ramificação do seu aplicativo, selecione o nome da ramificação na lista.
     d. Escolha Salvar.
   - Para excluir um subdomínio curinga
     a. Escolha Remover ao lado do nome do subdomínio. O tráfego para um subdomínio que não está explicitamente configurado é interrompido e o Amplify Hosting retorna um código de status 404 para essas solicitações.

b. Escolha Salvar.

Configuração de subdomínios automáticos para um domínio personalizado do Amazon Route 53

Depois que um aplicativo é conectado a um domínio personalizado no Route 53, o Amplify permite que você crie subdomínios automaticamente para filiais recém-conectadas. Por exemplo, se você conectar sua ramificação dev, o Amplify poderá criar automaticamente dev.exampledomain.com. Quando você exclui uma ramificação, todos os subdomínios associados são excluídos automaticamente.

Para configurar a criação automática de subdomínios para filiais recém-conectadas

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha um aplicativo conectado a um domínio personalizado gerenciado no Route 53.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Domínios personalizados.
4. Na página Domínios personalizados, escolha Configuração de domínio.
5. Na seção Criação automática de subdomínios, ative o recurso.

> **Note**
>
> Esse atributo está disponível somente para domínios raiz, por exemplo, exampledomain.com. O console do Amplify não exibirá essa caixa de seleção se seu domínio já for um subdomínio, como dev.exampledomain.com.

Pré-visualizações da Web com subdomínios

Depois de habilitar a Criação automática de subdomínios usando as instruções anteriores, as pré-visualizações da Web de solicitação pull da sua aplicação também estarão acessíveis com os subdomínios criados automaticamente. Quando uma solicitação pull é fechada, a ramificação e o subdomínio associados são excluídos automaticamente. Para obter mais informações sobre como configurar visualizações na web para uma solicitação pull, consulte Pré-visualizações na web para solicitações pull.

Solucionar problemas de domínios personalizados

Se você encontrar problemas ao adicionar um domínio personalizado a uma aplicação no console do AWS Amplify , consulte Solucionar problemas de domínios personalizados no capítulo de solução de problemas do Amplify. Se você não encontrar uma solução para seu problema ali, entre em contato com o Suporte. Para obter mais informações, consulte Criação de um caso de suporte no Guia do usuário do AWS Support .

Suporte de firewall para sites hospedados pelo Amplify

O suporte de firewall para sites hospedados pelo Amplify permite que você proteja seus aplicativos da web com uma integração direta com. AWS WAF AWS WAF permite configurar um conjunto de regras, chamado de lista de controle de acesso à web (Web ACL), que permite, bloqueia ou monitora (conta) solicitações da web com base nas regras e condições de segurança da web personalizáveis que você define. Ao integrar seu aplicativo Amplify AWS WAF, você ganha mais controle e visibilidade do tráfego HTTP aceito pelo seu aplicativo. Para saber mais AWS WAF, consulte Como AWS WAF funciona no Guia do AWS WAF desenvolvedor.

O suporte ao firewall está disponível em todas as áreas Regiões da AWS em que a Amplify Hosting opera. Essa integração se enquadra em um recurso AWS WAF global, semelhante CloudFront a. A Web ACLs pode ser conectada a vários aplicativos do Amplify Hosting, mas eles devem residir na mesma região.

Você pode usar AWS WAF para proteger seu aplicativo Amplify de explorações comuns da web, como injeção de SQL e scripts entre sites. Isso pode afetar a disponibilidade e a performance da sua aplicação, comprometer a segurança ou consumir recursos excessivos. Por exemplo, é possível criar regras para permitir ou bloquear solicitações de intervalos de endereços IP especificados, solicitações de blocos CIDR, solicitações originárias de um país ou região específico, solicitações que contenham código SQL inesperado ou scripting.

Você também pode criar regras que correspondam a uma string especificada ou um padrão de expressão regular em cabeçalhos HTTP, método, URI, string de consulta e o corpo da solicitação (limitados aos primeiros 8 KB). Além disso, é possível criar regras para bloquear eventos de agentes de usuário específicos, bots e extratores de conteúdo. Por exemplo, podem ser utilizadas regras baseadas em intervalos para especificar o número de solicitações da web que são permitidas por cada IP do cliente no final de um período de cinco minutos em atualização contínua.

Para saber mais sobre os tipos de regras compatíveis e os AWS WAF recursos adicionais, consulte o Guia do AWS WAF desenvolvedor e a Referência da AWS WAF API.

> **Important**
>
> A segurança é uma responsabilidade compartilhada entre você AWS e você. AWS WAF não é a solução para todos os problemas de segurança da Internet e você deve configurá-la para atender aos seus objetivos de segurança e conformidade. Para ajudar você a entender como

aplicar o modelo de responsabilidade compartilhada ao usar AWS WAF, consulte Segurança no uso do AWS WAF serviço.

Tópicos

- AWS WAF Habilitando um aplicativo Amplify no Console de gerenciamento da AWS
- Como desassociar uma ACL da Web de uma aplicação do Amplify
- AWS WAF Habilitando um aplicativo Amplify usando o AWS CDK
- Como o Amplify se integra com AWS WAF
- Preços de firewall para aplicação do Amplify

AWS WAF Habilitando um aplicativo Amplify no Console de gerenciamento da AWS

Você pode ativar AWS WAF as proteções para um aplicativo do Amplify no console do Amplify ou no console. AWS WAF

- Console do Amplify — Você pode ativar os recursos do Firewall para um aplicativo Amplify existente associando uma ACL AWS WAF da web ao seu aplicativo no console do Amplify. Use a proteção de um clique para criar uma ACL da Web com regras pré-configuradas que consideramos as práticas recomendadas para a maioria das aplicações. Você tem a opção de personalizar o acesso por endereço IP e país. As instruções apresentadas nesta seção descrevem como configurar proteções de um clique.
- AWS WAF console — Use uma Web ACL pré-configurada que você cria no AWS WAF console ou usando o. AWS WAF APIs Você deve criar uma web ACLs que deseja associar a um aplicativo Amplify na região Global (CloudFront). A web regional ACLs pode já existir no seu Conta da AWS, mas não é compatível com o Amplify. Para obter instruções de introdução, consulte Configuração AWS WAF e seus componentes no Guia do AWS WAF desenvolvedor.

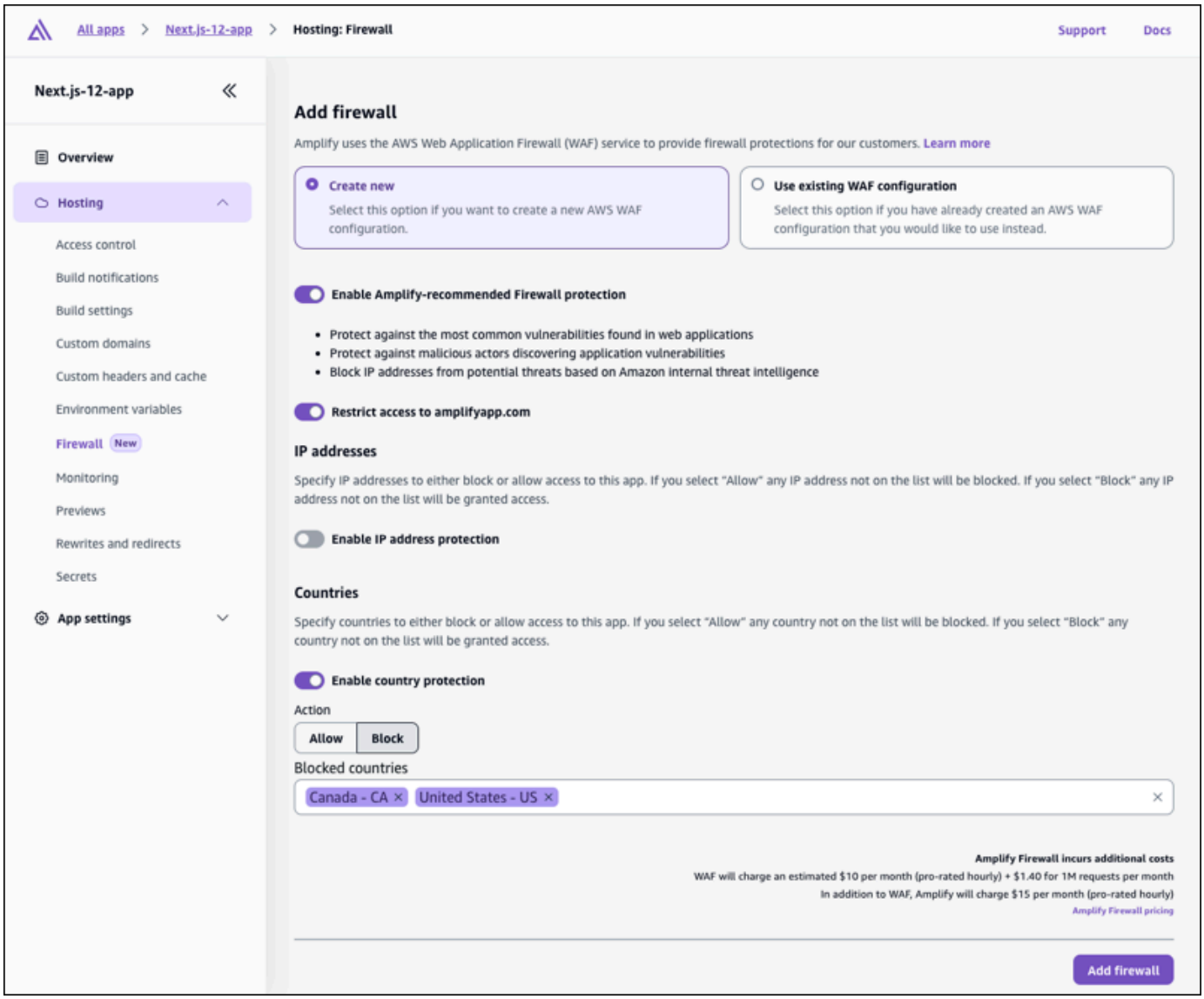
Use o procedimento a seguir AWS WAF para habilitar um aplicativo existente no console do Amplify.

Habilitar AWS WAF para um aplicativo Amplify existente

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/

2. Na página Todas as aplicações, escolha o nome da aplicação implantada para ativar o recurso do Firewall.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Firewall.

   A captura de tela a seguir mostra como navegar até a página Adicionar firewall no console do Amplify.



4. Na página Adicionar firewall, suas ações dependerão de você querer criar uma nova AWS WAF configuração ou usar uma existente.
   - Crie uma nova AWS WAF configuração.
     a. Selecione Create new (Criar novo).
     b. Como opção, habilite qualquer uma das configurações a seguir:

i. Ative Habilitar a proteção de firewall recomendada pelo Amplify.

ii. Ative Restringir acesso a amplifyapp.com para impedir o acesso à sua aplicação no domínio padrão do Amplify.

iii. Em endereços IP, ative Habilitar proteções de endereço IP.

   A. Em Ação, escolha Permitir se quiser especificar os endereços IP que terão acesso, e todos os outros serão bloqueados. Escolha Bloquear se quiser especificar os endereços IP que serão bloqueados, e todos os outros terão acesso.

   B. Para a versão IP, selecione IPV4ou IPV6.

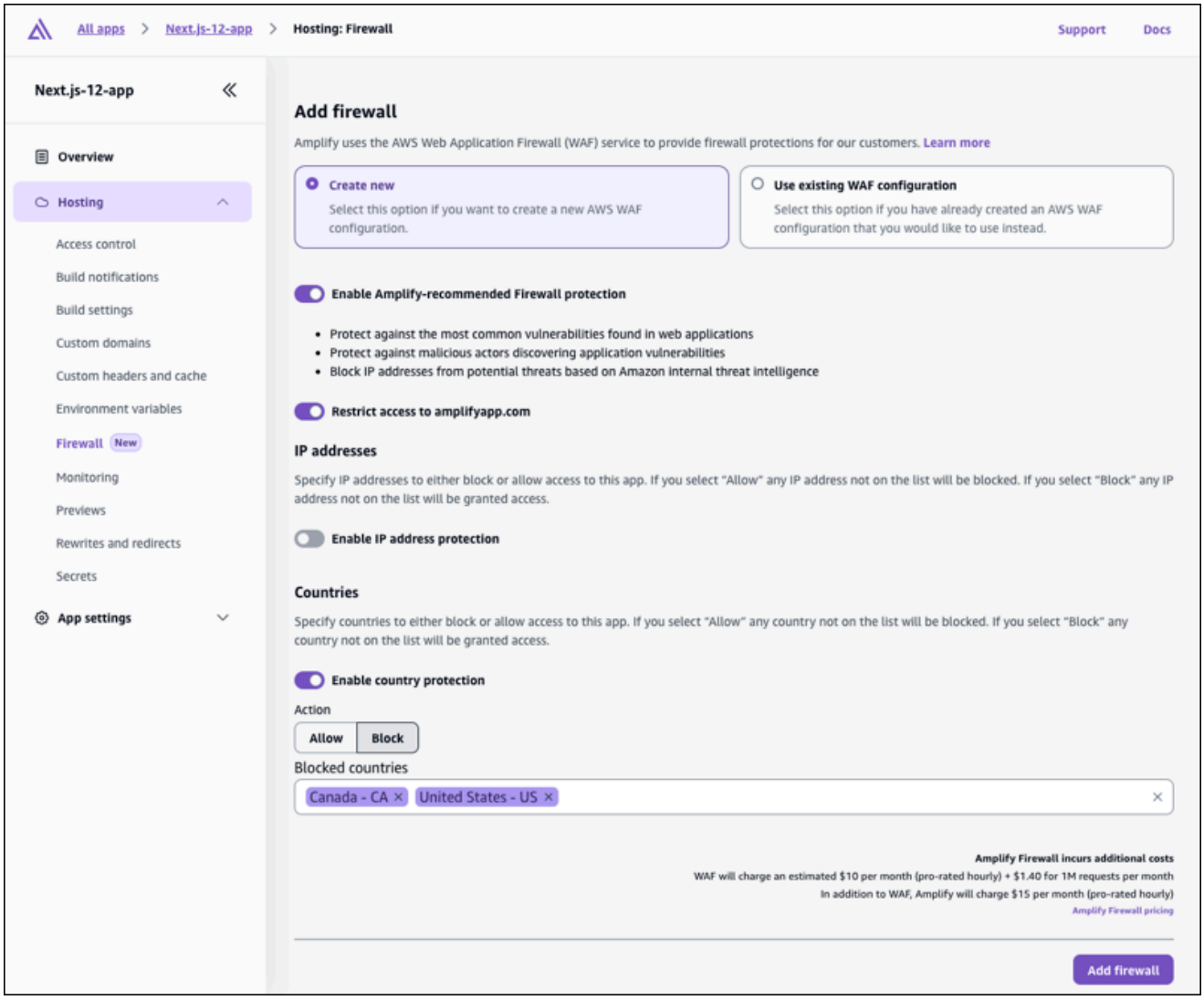
   C. Na caixa de texto Endereços IP, insira seus endereços IP permitidos ou bloqueados, um por linha, no formato CIDR.

iv. Em Países, ative a opção Habilitar a proteção do país.

   A. Em Ação, escolha Permitir se quiser especificar os países que terão acesso, e todos os outros serão bloqueados. Escolha Bloquear se quiser especificar os países que serão bloqueados, e todos os outros terão acesso.

   B. Em Países, selecione seus países permitidos ou bloqueados na lista.

A captura de tela a seguir demonstra como habilitar uma nova AWS WAF configuração para um aplicativo.



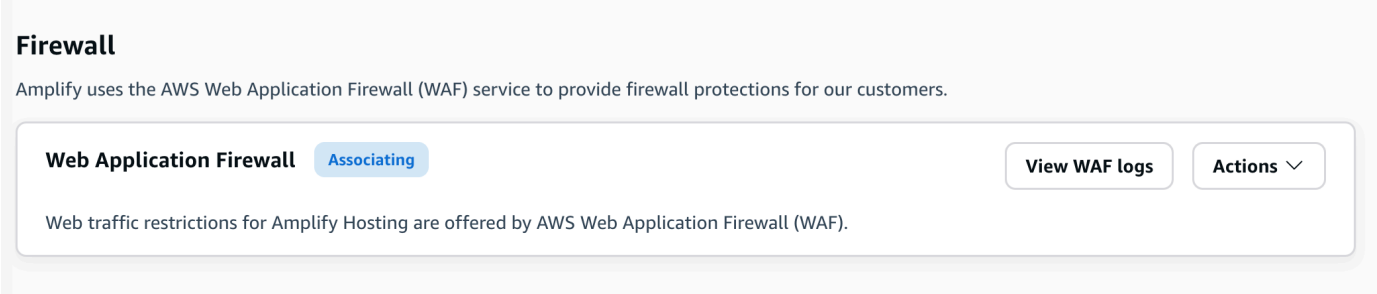
- Use uma AWS WAF configuração existente.

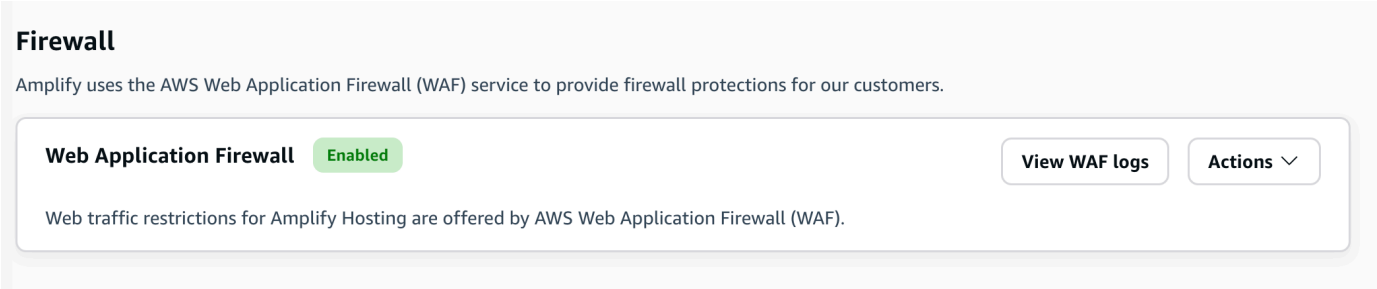
  a. Escolha Usar AWS WAF configuração existente.

  b. Selecione uma configuração salva na lista de sites ACLs AWS WAF em seu Conta da AWS. A ACL da web que você associa ao seu aplicativo Amplify deve ser criada na região Global CloudFront (). A web regional ACLs pode já existir no seu Conta da AWS, mas não é compatível com o Amplify.

5. Escolha Adicionar firewall.
6. Na página Firewall, o status de associação é exibido para indicar que as AWS WAF configurações estão sendo propagadas. Quando o processo for concluído, o status mudará para Habilitado.

As capturas de tela a seguir mostram o status do progresso do firewall no console do Amplify, indicando quando AWS WAF a configuração está associada e ativada.





Como desassociar uma ACL da Web de uma aplicação do Amplify

Não é possível excluir uma ACL da Web associada a uma aplicação do Amplify. Primeiro, é necessário desassociar a ACL da Web da aplicação no console do Amplify. Em seguida será possível excluí-la no console do AWS WAF .

Para desassociar uma ACL da Web de uma aplicação do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/
2. Na página Todas as aplicações, escolha o nome da aplicação da qual desassociar uma ACL da Web.
3. No painel de navegação, escolha Hospedagem, e, em seguida, escolha Firewall.
4. Na página Firewall, escolha Ações e, em seguida, escolha Desassociar firewall.
5. No modal de confirmação, insira **`disassociate`**, e, em seguida, escolha Desassociar.firewall.
6. Na página Firewall, o status de Desassociação é exibido para indicar que as AWS WAF configurações estão sendo propagadas.

   Quando o processo estiver concluído, você poderá excluir a Web ACL no AWS WAF console.

AWS WAF Habilitando um aplicativo Amplify usando o AWS CDK

Você pode usar o AWS Cloud Development Kit (AWS CDK) AWS WAF para habilitar um aplicativo Amplify. Para saber mais sobre o uso do CDK, consulte O que é o CDK? no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

O exemplo de TypeScript código a seguir demonstra como criar um AWS CDK aplicativo com duas pilhas de CDK: uma para Amplify e outra para. AWS WAF Observe que a AWS WAF pilha deve ser implantada na região Leste dos EUA (Norte da Virgínia) (us-east-1). A pilha de aplicações do Amplify pode ser implantada em uma região diferente. Você deve criar a ACL da web que deseja associar ao aplicativo Amplify na região Global CloudFront (). A web regional ACLs pode já existir no seu Conta da AWS, mas não é compatível com o Amplify.

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

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

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

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

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

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

const app = new cdk.App();

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

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

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

Como o Amplify se integra com AWS WAF

A lista a seguir fornece detalhes específicos sobre como o suporte ao Firewall é integrado AWS WAF e as restrições a serem consideradas ao criar sites ACLs e associá-los aos aplicativos Amplify.

- Você pode ativar AWS WAF qualquer tipo de aplicativo Amplify. Isso inclui qualquer estrutura compatível, aplicativos renderizados do lado do servidor (SSR) e sites totalmente estáticos. AWS WAF é compatível com os aplicativos Amplify Gen 1 e Gen 2.
- Você deve criar uma web ACLs que deseja associar a um aplicativo Amplify na região Global (CloudFront). A web regional ACLs pode já existir no seu Conta da AWS, mas não é compatível com o Amplify.

- A ACL da Web e a aplicação do Amplify devem ser criadas na mesma Conta da AWS. Você pode usar AWS Firewall Manager para replicar AWS WAF regras e simplificar a manutenção das regras da organização centralizadas e distribuídas entre várias. Contas da AWS Contas da AWS Para saber mais, consulte AWS Firewall Manager no Guia do desenvolvedor do AWS WAF .
- É possível compartilhar a mesma ACL da Web em várias aplicações do Amplify na mesma Conta da AWS. Todas as aplicações devem estar na mesma região.
- Quando você associa uma ACL da Web a uma aplicação do Amplify, a ACL da Web é anexada a todas as ramificações da aplicação por padrão. Quando você criar novas ramificações, elas terão a ACL da Web.
- Quando você associar uma ACL da Web a uma aplicação do Amplify, ela será automaticamente associada a todos os domínios da aplicação. No entanto, é possível configurar regras que se apliquem a um único nome de domínio usando regras de correspondência de cabeçalhos de host.
- Não é possível excluir uma ACL da Web associada a uma aplicação do Amplify. Antes de excluir uma Web ACL no AWS WAF console, você precisa desassociá-la do aplicativo.

Política de recursos de ACLs da Web do Amplify

Para permitir que o Amplify acesse sua ACL da Web, uma política de recursos é anexada à ACL da Web durante a associação. O Amplify constrói essa política de recursos automaticamente, mas você pode visualizá-la usando a API. AWS WAFV2 GetPermissionPolicy As permissões do IAM a seguir são necessárias para associar uma ACL da Web a uma aplicação do Amplify.

- amplificar: ACL AssociateWeb
- onda 2: ACL AssociateWeb
- onda 2: PutPermissionPolicy
- onda 2: GetPermissionPolicy

Preços de firewall para aplicação do Amplify

O custo de implementação AWS WAF em um aplicativo Amplify é calculado com base nos dois componentes a seguir:

- AWS WAF uso — Você será cobrado pelo seu AWS WAF uso de acordo com o modelo de AWS WAF preços. AWS WAF as cobranças são baseadas nas listas de controle de acesso à web (web ACLs) que você cria, no número de regras que você adiciona por ACL da web e no número de

solicitações da web que você recebe. Para obter detalhes sobre os preços, consulte Preços do AWS WAF.

- Custo de integração do Amplify Hosting: há uma taxa de 15,00 USD por mês, por aplicação, quando você anexa uma ACL da Web a uma aplicação do Amplify. Isso é rateado por hora.

Implantações de ramificação de atributo e fluxos de trabalho da equipe

O Amplify Hosting foi projetado para funcionar com ramificações de recursos e GitFlow fluxos de trabalho. O Amplify usa ramificações do Git para criar uma nova implantação sempre que você conecta uma nova ramificação ao seu repositório. Depois de conectar sua primeira ramificação, você cria ramificações de atributos adicionais.

Para adicionar uma ramificação a uma aplicação

1. Escolha a aplicação à qual você deseja adicionar a ramificação.
2. Escolha Configurações da aplicação e, em seguida, Configurações da ramificação.
3. Na página Configurações da ramificação, escolha Adicionar ramificação.
4. Selecione uma ramificação do seu repositório.
5. Escolha Adicionar ramificação.
6. Reimplante sua aplicação.

Depois de adicionar uma ramificação, seu aplicativo tem duas implantações disponíveis nos domínios padrão do Amplify, como e. https://main.appid.amplifyapp.comhttps://dev.appid.amplifyapp.com Isso pode variar team-to-team, mas normalmente a ramificação principal rastreia o código de lançamento e é sua ramificação de produção. A ramificação de desenvolvimento é usada como uma ramificação de integração para testar novos atributos. Isso permite aos testadores da versão beta o teste de atributos não lançados na implantação da ramificação de desenvolvimento, sem afetar nenhum usuário final de produção na implantação da ramificação principal.

Tópicos

- Use backends do Amplify em todas as aplicações (somente aplicações Gen 1)

Fluxos de trabalho de equipe com aplicações Amplify Gen 2 full-stack

AWS O Amplify Gen 2 apresenta uma experiência de desenvolvedor TypeScript baseada em código que prioriza a definição de back-ends. Para saber mais sobre fluxos de trabalho full-stack com as aplicações do Amplify Gen 2, consulte Fluxos de trabalho full-stack nos Documentos do Amplify.

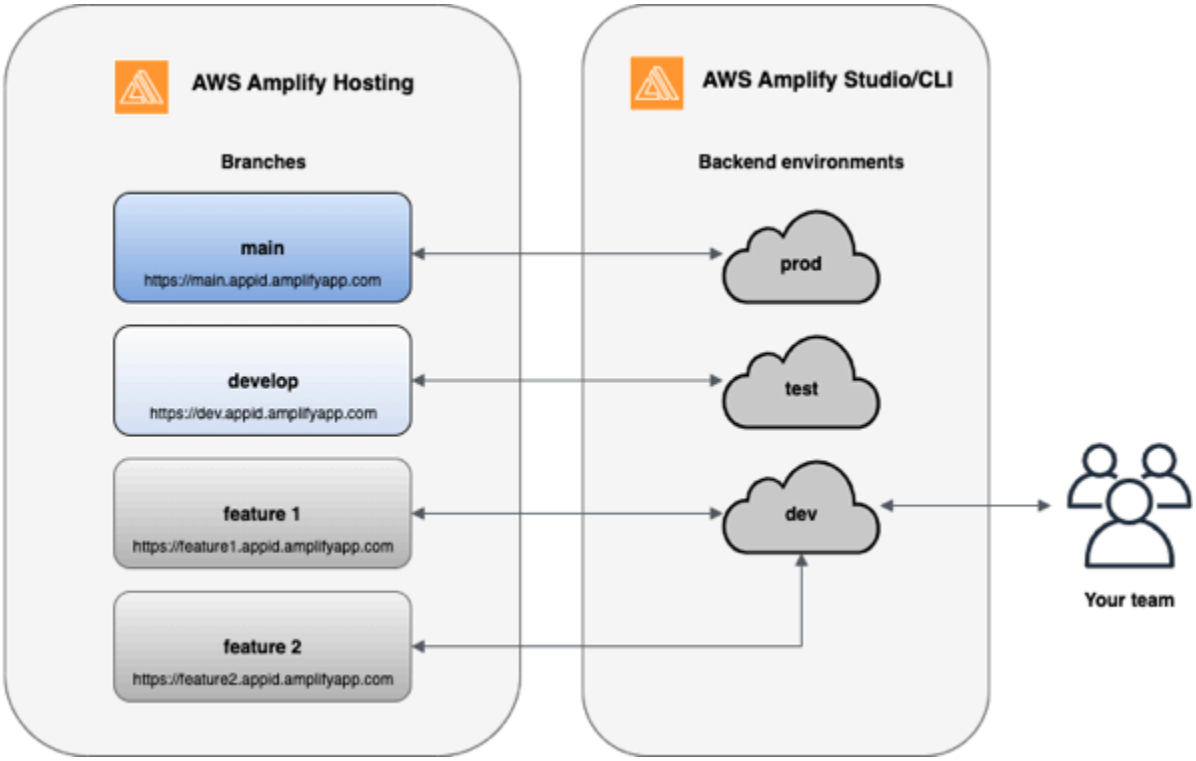
Fluxos de trabalho de equipe com aplicações Amplify Gen 1 full-stack

A implantação de uma ramificação de atributos consiste em um ambiente de frontend e um ambiente de backend opcional. O frontend é construído e implantado em uma rede global de entrega de conteúdo (CDN), enquanto o backend é implantado pelo Amplify Studio ou pela CLI do Amplify para AWS. Para saber como configurar esse cenário de implantação, consulte Compilação de um backend para uma aplicação.

O Amplify Hosting implanta continuamente recursos de back-end, como funções GraphQL APIs e Lambda, com suas implantações de ramificações de recursos. É possível usar os seguintes modelos de ramificação para implantar seu backend e frontend com Amplify Hosting.

Fluxo de trabalho de ramificação de atributo

- Crie os ambientes de backend prod, test e dev com a Amplify CLI.
- Mapeie o backend do produto para a ramificação principal.
- Mapeie o backend de teste para a ramificação de desenvolvimento.
- Os membros da equipe podem usar o ambiente de backend de desenvolvimento para testar ramificações de atributos individuais.



1. Instale a Amplify CLI para inicializar um novo projeto do Amplify.

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

2. Inicialize um ambiente de backend prod para o seu projeto. Se você não tiver um projeto, crie um usando ferramentas de bootstrap como create-react-app ou 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. Adicione os ambientes de backend test e dev.

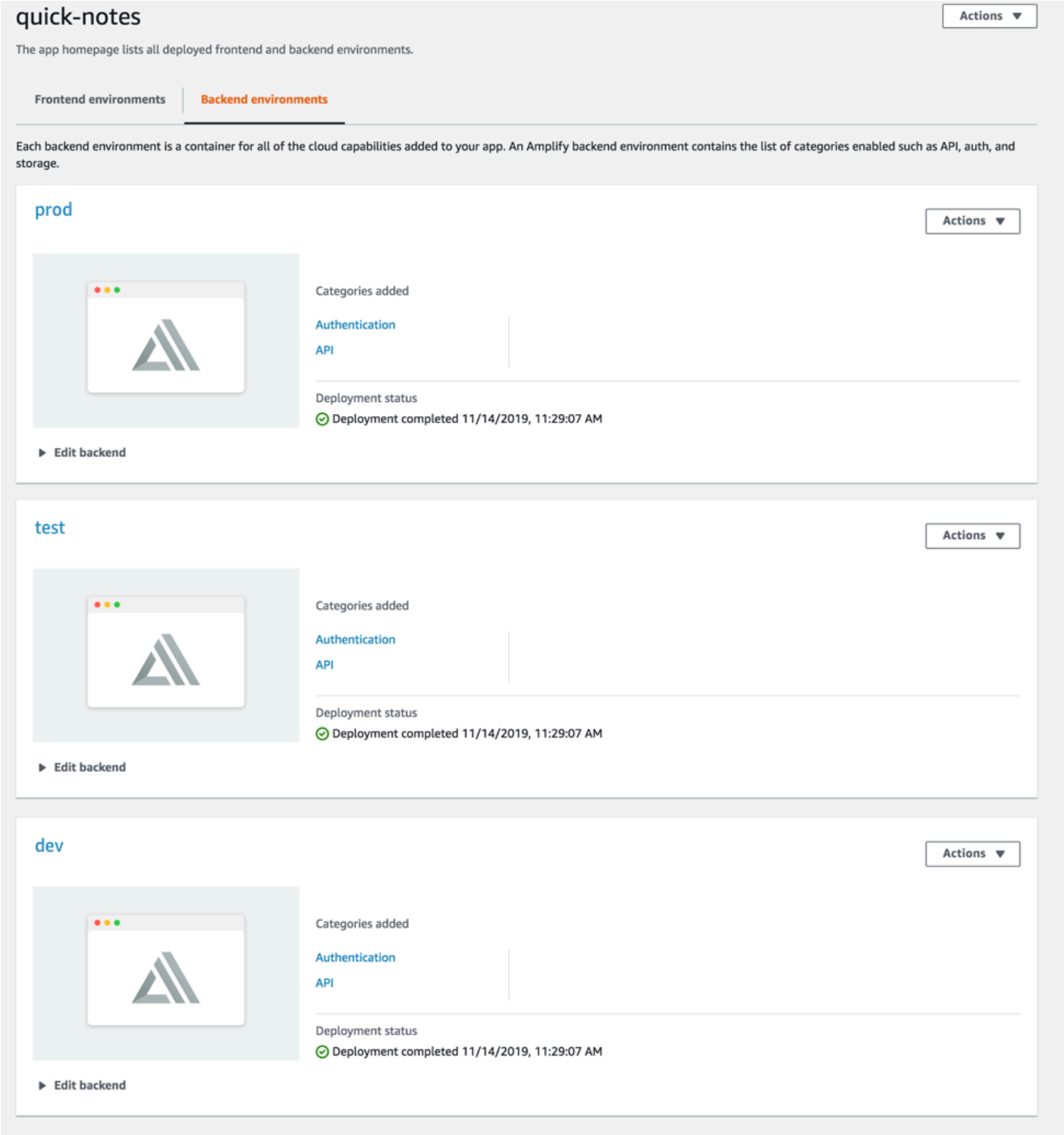
```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: test
...
amplify push
```

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

4. Envie o código para um repositório Git de sua escolha (neste exemplo, vamos supor que você enviou para o principal).

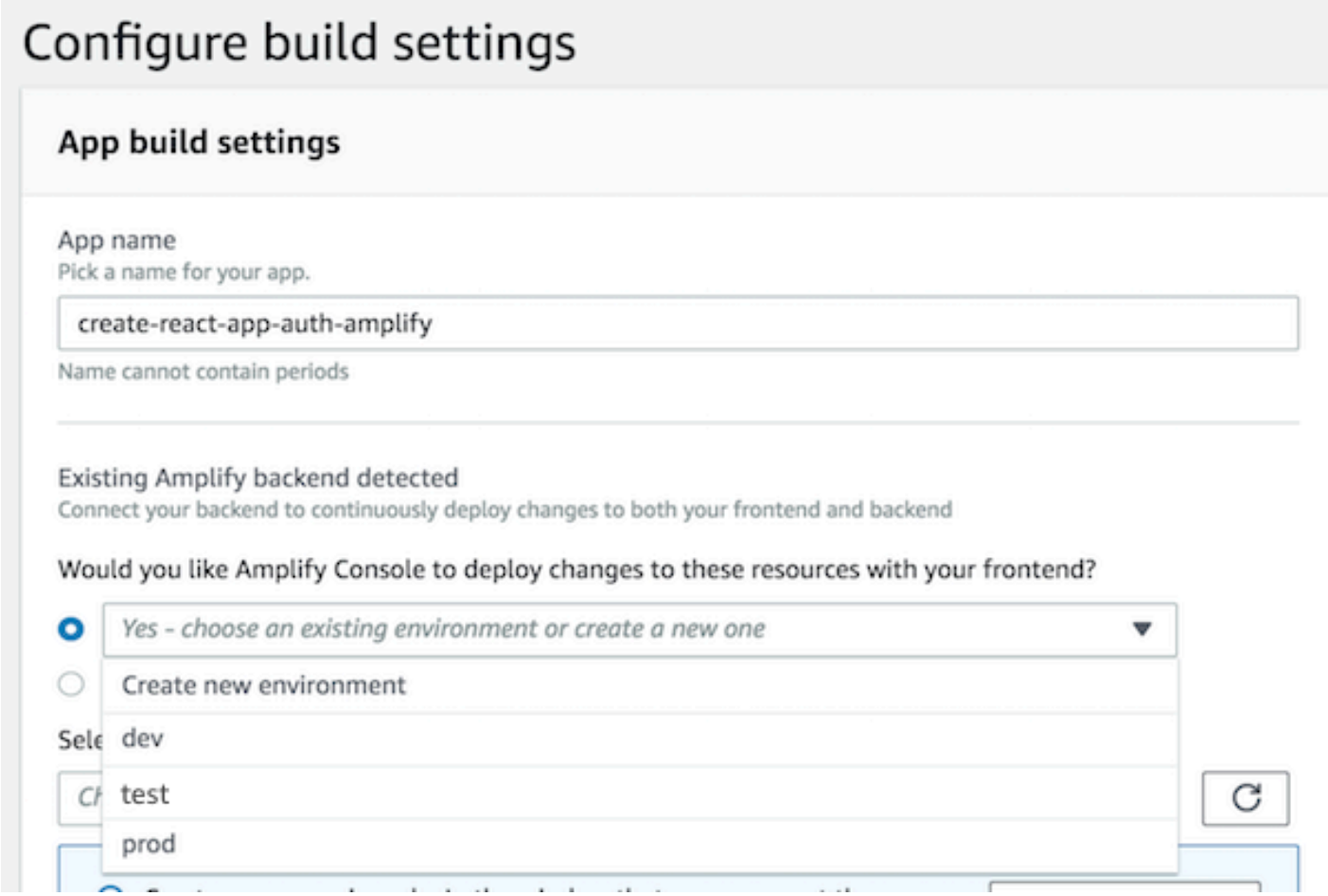
```
git commit -am 'Added dev, test, and prod environments'
git push origin main
```

5. Visite Amplify no Console de gerenciamento da AWS para ver seu ambiente de back-end atual. Navegue um nível acima no rastro de navegação para ver uma lista de todos os ambientes de backend criados na guia Ambientes de backend.

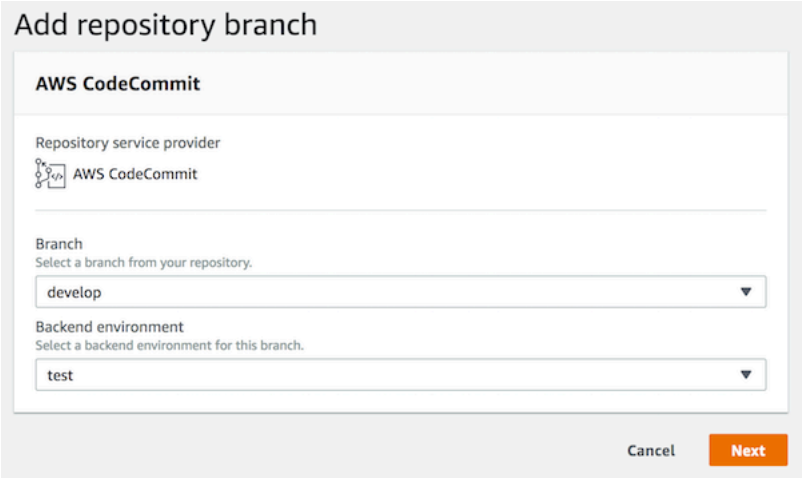


6. Mude para a guia Ambientes de frontend e conecte seu provedor de repositório e a ramificação principal.
7. Na página de configurações de compilação, selecione um ambiente de backend existente para configurar a implantação contínua com a ramificação principal. Escolha prod na lista e conceda

o perfil de serviço para o Amplify. Escolha Salvar e implantar. Após a conclusão da compilação, você terá uma implantação da filial principal disponível em https://main.appid.amplifyapp.com.



8. Conecte a ramificação develop no Amplify (presuma que as ramificações develop e principal são as mesmas nesse ponto). Escolha o ambiente de backend test.



9. O Amplify agora está configurado. É possível começar a trabalhar em novos atributos em um atributo de atributo. Adicione a funcionalidade de backend usando o ambiente de backend dev na sua estação de trabalho local.

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

10. Quando terminar de trabalhar no atributo, confirme seu código, crie uma solicitação pull para revisar internamente.

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

11. Para visualizar como serão as alterações, acesse o console do Amplify e conecte a ramificação de atributo. Nota: Se você tiver o AWS CLI instalado em seu sistema (não na CLI do Amplify), você pode conectar uma ramificação diretamente do seu terminal. Para encontrar o ID do aplicativo, vá até Configurações do aplicativo > Geral > ARN do aplicativo: 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. Seu recurso estará acessível em https://newinternet.appid.amplifyapp.compara compartilhar com seus colegas de equipe. Se tudo estiver aparentemente correto, mescle o PR à ramificação de desenvolvimento.

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

13. Isso iniciará uma compilação que atualizará o back-end e o front-end no Amplify com uma implantação de ramificação em. https://dev.appid.amplifyapp.com É possível compartilhar esse link com partes interessadas, para que possam revisar o novo atributo.

14. Exclua sua ramificação de atributos do Git, Amplify e remova o ambiente de backend da nuvem (você sempre pode criar executando ‘amplify env checkout prod’ e executando ‘amplify env add’).

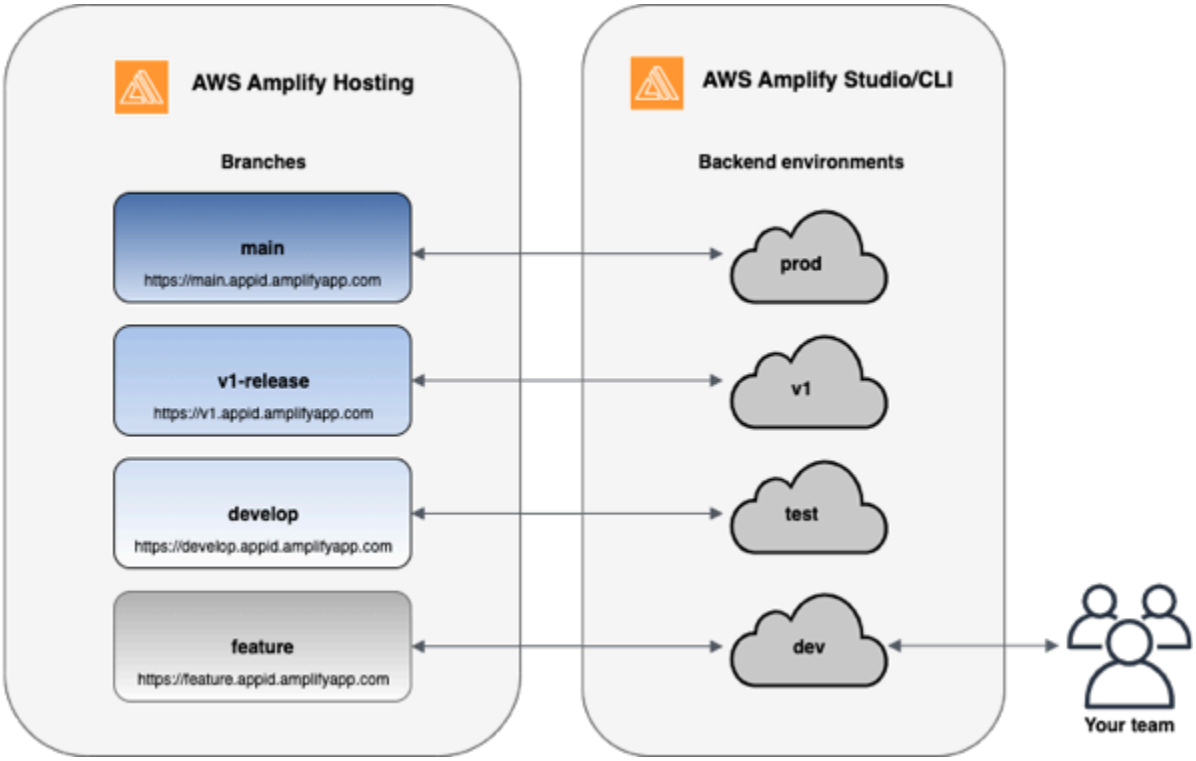
```
git push origin --delete newinternet
```

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

GitFlow fluxo de trabalho

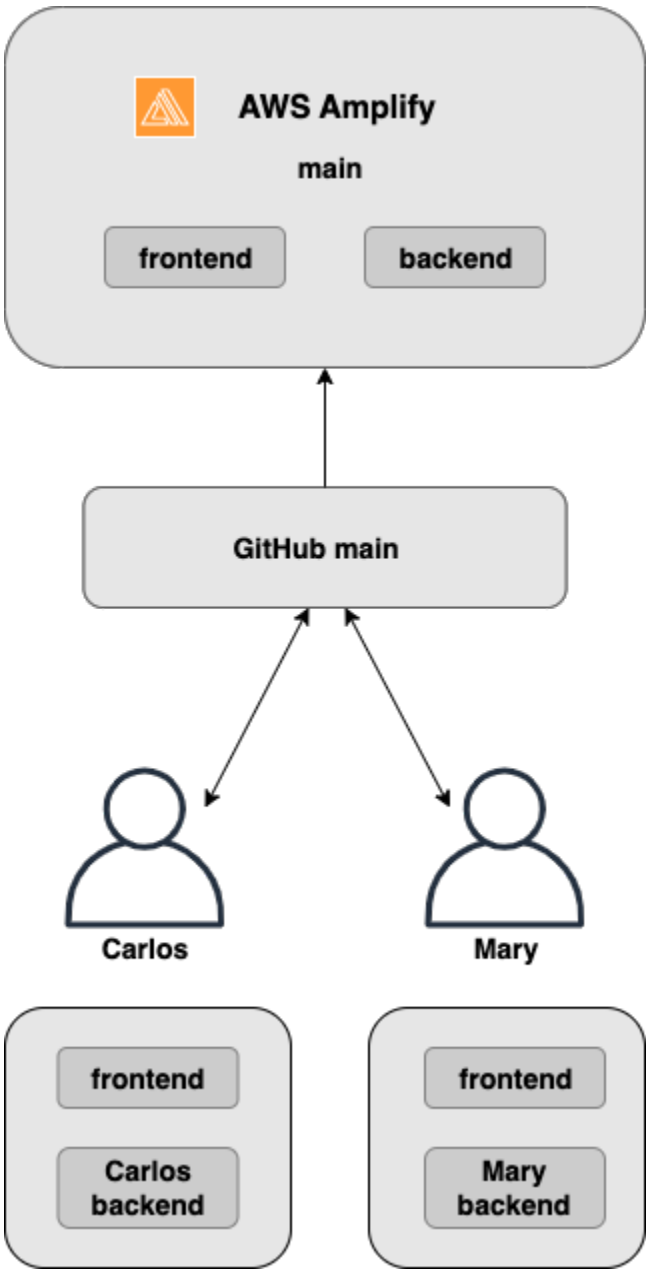
GitFlow usa duas ramificações para registrar o histórico do projeto. A ramificação principal rastreia somente o código de lançamento, e a ramificação de desenvolvimento é usada como uma ramificação de integração para novos recursos. GitFlow simplifica o desenvolvimento paralelo ao isolar o novo desenvolvimento do trabalho concluído. O novo desenvolvimento (como atributos e correções de erros não emergenciais) é feito em ramificações de atributo. Quando o desenvolvedor estiver satisfeito com o código e entender que ele está pronto para o lançamento, a ramificação de atributo será mesclada com a ramificação de desenvolvimento de integração. As únicas confirmações para a ramificação principal são as mesclagens de ramificações release e ramificações hotfix (para corrigir erros emergenciais).

O diagrama abaixo mostra uma configuração recomendada com GitFlow. É possível seguir o mesmo processo descrito na seção de fluxo de trabalho de ramificação de atributo acima.



Sandbox de desenvolvedor

- Cada desenvolvedor em uma equipe cria um ambiente de sandbox na nuvem, separado do computador local. Isso permite que os desenvolvedores trabalhem em isolamento uns dos outros sem substituir as alterações de outros membros da equipe.
- Cada ramificação no Amplify tem seu próprio backend. Isso garante que o Amplify use o repositório do Git como uma fonte única da verdade a partir da qual implantar alterações, em vez de depender que os desenvolvedores da equipe enviem manualmente o backend ou frontend para a produção a partir do computador local deles.



1. Instale a Amplify CLI para inicializar um novo projeto do Amplify.

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

2. Inicialize um ambiente de backend mary para o seu projeto. Se você não tiver um projeto, crie um usando ferramentas de bootstrap como create-react-app ou 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. Envie o código para um repositório Git de sua escolha (neste exemplo, vamos supor que você enviou para o principal).

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

4. Conecte seu repo > principal ao Amplify.
5. O console do Amplify detecta ambientes de backend criados pela Amplify CLI. Escolha Criar novo ambiente no menu suspenso e conceda a perfil de serviço para o console do Amplify. Escolha Salvar e implantar. Após a conclusão da compilação, você terá uma implantação de ramificação principal disponível em https://main.appid.amplifyapp.comum novo ambiente de back-end vinculado à ramificação.
6. Conecte a ramificação desenvolver no Amplify (presuma que as ramificações desenvolver e principal são as mesmas nesse ponto) e escolha Criar

Implantações de ramificação de atributo com base em padrão

As implantações de ramificação com base em padrão permitem que você implante automaticamente ramificações que associem um padrão específico ao Amplify. As equipes de produto que usam ramificações de recursos ou GitFlow fluxos de trabalho para seus lançamentos agora podem definir padrões, como **`release**`** implantar automaticamente ramificações do Git que começam com 'release' em uma URL compartilhável.

1. Escolha Configurações da aplicação e, em seguida, Configurações da ramificação.
2. Na página Configurações da ramificação, selecione Editar.
3. Selecione Detecção automática de ramificações para conectar automaticamente as ramificações ao Amplify que correspondam a um conjunto de padrões.
4. Na caixa Detecção automática de ramificações - padrões, insira os padrões para implantação automática de ramificações.
   - **`*`** – Implantar todas as ramificações no seu repositório.
   - **`release*`**: implanta todas as ramificações iniciadas pela palavra "release".
   - **`release*/`** – Implantar todas as ramificações que correspondem a um padrão “release /”.

- Especifique vários padrões em uma lista separada por vírgulas. Por exemplo, **`.release*, feature*`**

5. Configure a proteção de senha automática para todas as ramificações que são criadas automaticamente selecionando Controle de acesso de detecção automática da ramificações.
6. Para aplicações Gen 1 integradas a um backend do Amplify, é possível optar por criar um ambiente ou apontar todas as ramificações para um backend já existente.
7. Escolha Salvar.

Implantações de ramificações de atributos baseadas em padrões para um aplicativo conectado a um domínio personalizado

É possível usar implantações de ramificações de atributos baseadas em padrões para um aplicativo conectado a um domínio personalizado do Amazon Route 53.

- Para obter instruções de configuração de implantações de ramificações de atributos baseadas em padrões, consulte Configuração de subdomínios automáticos para um domínio personalizado do Amazon Route 53
- Para obter instruções sobre como conectar um aplicativo Amplify a um domínio personalizado gerenciado no Route 53, consulte Adição de um domínio personalizado gerenciado pelo Amazon Route 53
- Para obter mais informações sobre o uso do Route 53, consulte O que é o Amazon Route 53.

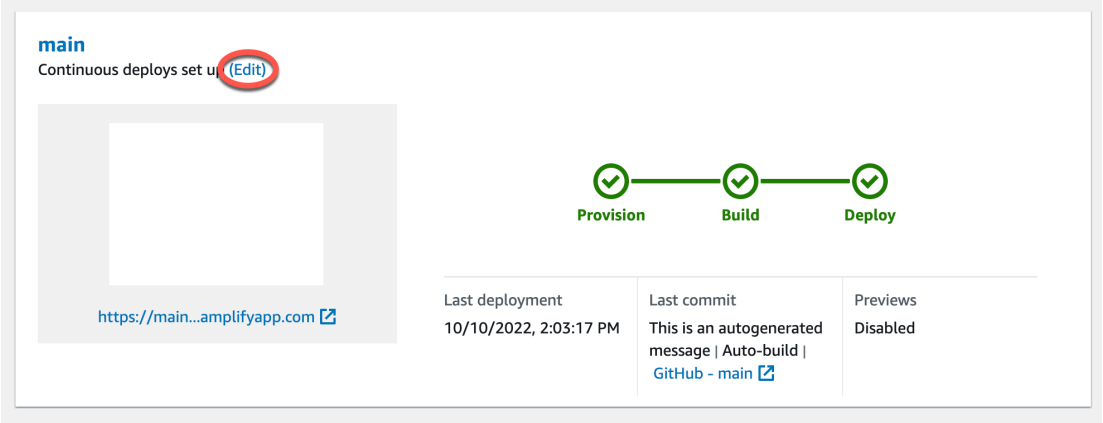
Geração automática em tempo de compilação da configuração do Amplify (somente aplicações Gen 1)

> **Note**
>
> As informações nesta seção são somente para aplicações Gen 1. Se você quiser implantar automaticamente alterações de infraestrutura e código de aplicação a partir de ramificações de atributos para uma aplicação Gen 2, consulte Implantações de ramificações full-stack na Documentação do Amplify

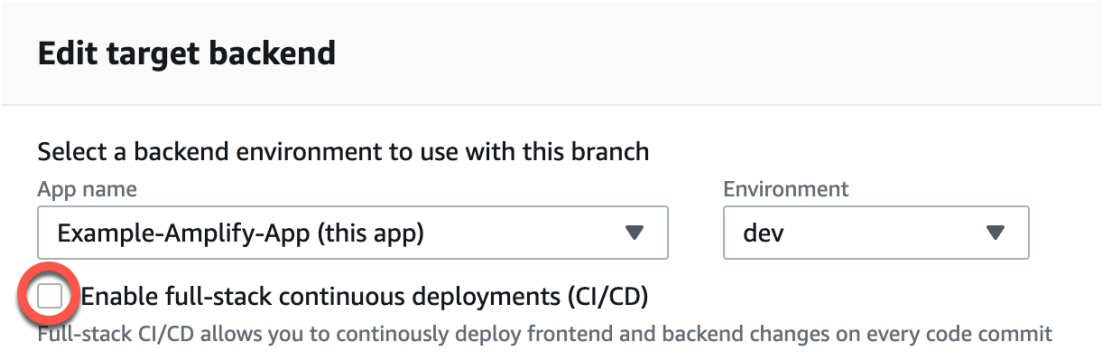
O Amplify oferece suporte à geração automática em tempo de compilação do arquivo de configuração `aws-exports.jsdo` Amplify para aplicações Gen 1. Ao desativar as implantações de CI/CD full-stack, você permite que seu aplicativo gere automaticamente o arquivo `aws-exports.js` e garante que as atualizações não sejam feitas em seu backend no momento da compilação.

Para gerar automaticamente **`aws-exports.js`** no momento da construção

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para editar.
3. Escolha a guia Ambientes de hospedagem.
4. Localize a ramificação a ser editada e escolha Editar.



5. Na página Editar backend de destino, desmarque Habilitar implantações contínuas full-stack (CI/CD) para desativar a CI/CD full-stack para esse backend.



6. Selecione um perfil de serviço existente para dar ao Amplify as permissões necessárias para fazer alterações no backend do seu aplicativo. Se precisar criar um perfil de serviço, escolha Criar novo perfil. Para obter mais informações sobre como criar um perfil de serviço, consulte Adição de um perfil de serviço com permissões para implantar recursos de backend.
7. Escolha Salvar. O Amplify aplica essas alterações na próxima vez que você criar o aplicativo.

Compilações condicionais de backend (somente aplicações Gen 1)

> **Note**
>
> As informações nesta seção são somente para aplicações Gen 1. O Amplify Gen 2 apresenta uma experiência de desenvolvedor TypeScript baseada e que prioriza o código. Portanto, esse atributo não é necessário para backends Gen 2.

O Amplify oferece suporte a construções condicionais de backend em todas as ramificações de uma aplicação Gen 1. Para configurar compilações de backend condicionais, defina a variável de ambiente `AMPLIFY_DIFF_BACKEND` como `true`. Habilitar compilações condicionais de backend ajudará a acelerar as compilações em que as alterações são feitas somente no frontend.

Quando você habilita compilações de backend baseadas em diff, no início de cada compilação, o Amplify tenta executar uma comparação na pasta em seu repositório `amplify`. Se o Amplify não encontrar nenhuma diferença, ele pula a etapa de compilação do backend e não atualiza seus atributos de backend. Se seu projeto não tiver uma pasta `amplify` no seu repositório, o Amplify ignorará o valor da variável de ambiente `AMPLIFY_DIFF_BACKEND`. Para obter instruções sobre como definir a variável de ambiente `AMPLIFY_DIFF_BACKEND`, consulte Configuração de compilações de backend baseadas em diff para uma aplicação Gen 1.

Se você atualmente tem comandos personalizados especificados nas configurações de compilação da sua fase de backend, as compilações condicionais de backend não funcionarão. Se quiser que esses comandos personalizados sejam executados, é necessário movê-los para a fase de frontend das configurações de compilação no arquivo `amplify.yml` do seu aplicativo. Para obter mais informações sobre como atualizar o arquivo `amplify.yml`, consulte Referência de especificação de compilação.

Use backends do Amplify em todas as aplicações (somente aplicações Gen 1)

> **Note**
>
> As informações nesta seção são somente para aplicações Gen 1. Se você quiser compartilhar recursos de backend para uma aplicação Gen 2, consulte Compartilhar recursos entre ramificações nos Documentos do Amplify

O Amplify permite reutilizar facilmente os ambientes de backend existentes em todas as suas aplicações Gen 1 em uma determinada região. É possível fazer isso ao criar um aplicativo, conectar uma nova ramificação a um aplicativo existente ou atualizar um frontend existente para apontar para um ambiente de backend diferente.

Reutilize backends ao criar um aplicativo

Para reutilizar um backend ao criar um aplicativo Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Para criar um backend para usar neste exemplo, faça o seguinte:
   a. No painel de navegação, selecione Todos os apps.
   b. Escolha Novo aplicativo, Compile um aplicativo.
   c. Insira um nome para o aplicativo, como **`Example-Amplify-App`**.
   d. Escolha Confirmar implantação.
3. Para conectar um frontend ao seu novo backend, escolha a guia Ambientes de hospedagem.
4. Escolha seu provedor git e escolha Conectar ramificação.
5. Na página Adicionar ramificação do repositório, em Repositórios atualizados recentemente, escolha o nome do seu repositório. Para Ramificação, selecione a ramificação do seu repositório para se conectar.
6. Na página Configurações, faça o seguinte:
   a. Em Nome do aplicativo, selecione o aplicativo a ser usado para adicionar um ambiente de backend. É possível escolher o aplicativo atual ou qualquer outro aplicativo na região atual.

b. Em Ambiente, selecione o nome do ambiente de backend a ser adicionado. É possível usar um ambiente existente ou criar outro.

c. Por padrão, a pilha completa CI/CD está desativada. Desativar o CI/CD de full-stack faz com que o aplicativo seja executado no modo pull only. No momento da compilação, o Amplify gerará automaticamente somente o arquivo `aws-exports.js`, sem modificar seu ambiente de backend.

d. Selecione um perfil de serviço existente para dar ao Amplify as permissões necessárias para fazer alterações no backend do seu aplicativo. Se precisar criar um perfil de serviço, escolha Criar novo perfil. Para obter mais informações sobre como criar um perfil de serviço, consulte Adição de um perfil de serviço com permissões para implantar recursos de backend.

e. Escolha Próximo.

7. Escolha Salvar e implantar.

Reutilize backends ao conectar uma ramificação a um aplicativo existente

Para reutilizar um backend ao conectar uma ramificação a um aplicativo Amplify existente

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo ao qual conectar uma nova filial.
3. No painel de navegação, em Configurações do aplicativo, selecione Geral.
4. Na seção Ramificações, escolha Conectar uma ramificação.
5. Na página Adicionar ramificação do repositório, em Ramificação, selecione a ramificação do seu repositório para se conectar.
6. Em Nome do aplicativo, selecione o aplicativo a ser usado para adicionar um ambiente de backend. É possível escolher o aplicativo atual ou qualquer outro aplicativo na região atual.
7. Em Ambiente, selecione o nome do ambiente de backend a ser adicionado. É possível usar um ambiente existente ou criar outro.
8. Se você precisar configurar um perfil de serviço para dar ao Amplify as permissões necessárias para fazer alterações no backend do seu aplicativo, o console solicitará que você execute essa tarefa. Para obter mais informações sobre como criar um perfil de serviço, consulte Adição de um perfil de serviço com permissões para implantar recursos de backend.
9. Por padrão, a pilha completa CI/CD está desativada. Desativar o full-stack CI/CD faz com que o aplicativo seja executado no modo pull only. No momento da compilação, o Amplify

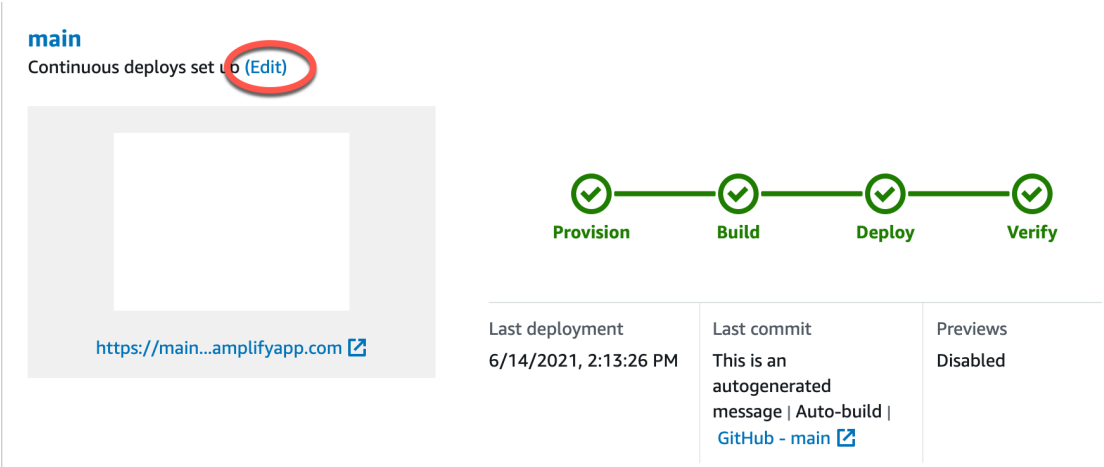
gerará automaticamente somente o arquivo `aws-exports.js`, sem modificar seu ambiente de backend.

10. Escolha Próximo.
11. Escolha Salvar e implantar.

Edite um frontend existente para apontar para um backend diferente

Para editar um aplicativo Amplify de frontend para apontar para um backend diferente

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual editar o backend.
3. Escolha a guia Ambientes de hospedagem.
4. Localize a ramificação a ser editada e escolha Editar.



5. Na página Selecionar um ambiente de backend para usar com essa ramificação, em Nome do aplicativo, selecione o aplicativo de frontend para o qual você deseja editar o ambiente de backend. É possível escolher o aplicativo atual ou qualquer outro aplicativo na região atual.
6. Em Ambiente de backend, selecione o nome do ambiente de backend a ser adicionado.
7. Por padrão, o full-stack CI/CD está habilitado. Desmarque essa opção para desativar a pilha completa CI/CD desse back-end. Desativar o full-stack CI/CD faz com que o aplicativo seja executado no modo pull only. No momento da compilação, o Amplify gerará automaticamente somente o arquivo `aws-exports.js`, sem modificar o ambiente de backend.
8. Escolha Salvar. O Amplify aplica essas alterações na próxima vez que você criar o aplicativo.

Compilação de um backend para uma aplicação

Com AWS Amplify você pode criar um aplicativo fullstack com dados, autenticação, armazenamento e hospedagem de front-end implantados em. AWS

AWS O Amplify Gen 2 apresenta uma TypeScript-based experiência de desenvolvedor que prioriza o código para definir back-ends. Para saber como usar o Amplify Gen 2 para compilar e conectar um backend à sua aplicação, consulte Compilação e criação de backends nos documentos do Amplify.

Se você estiver buscando pela documentação para compilar um backend para uma aplicação Gen 1 usando a CLI e o Amplify Studio, consulte Compilação e conexão de backends nos documentos do Amplify Gen 1.

Tópicos

Crie um backend para uma aplicação Gen 2

Para ver um tutorial que orienta você nas etapas de criação de um aplicativo fullstack do Amplify Gen 2 com um TypeScript-based back-end, consulte Primeiros passos na documentação do Amplify.

Crie um backend para uma aplicação Gen 1

Neste tutorial, você configurará um CI/CD fluxo de trabalho completo com o Amplify. Você implantará um aplicativo de frontend no Amplify Hosting. Em seguida, você criará um backend usando o Amplify Studio. Por fim, você conectará o backend da nuvem ao aplicativo de frontend.

Pré-requisitos

Antes de começar este tutorial, conclua os pré-requisitos a seguir.

Crie um repositório Git

> O Amplify suporta GitHub Bitbucket e. GitLab AWS CodeCommit Envie sua aplicação para seu repositório Git.

Instalação da interface de linha de comandos (CLI) do Amplify

Para obter instruções, consulte Instalar a Amplify CLI na documentação do Amplify Framework.

Etapa 1: Implantar um frontend

Se você tiver um aplicativo de frontend existente em um repositório Git que deseja usar neste exemplo, siga as instruções para implantar um aplicativo de frontend.

Se você precisar criar uma nova aplicação de frontend para usar neste exemplo, siga as instruções em Criar aplicação do React na documentação Criar aplicação do React.

Para implantar um aplicativo de frontend

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todos os aplicativos, escolha Novo aplicativo e, em seguida, Hospedar aplicativo web no canto superior direito.
3. Selecione seu provedor GitHub, Bitbucket ou AWS CodeCommit repositório e escolha Continuar. GitLab
4. O Amplify autoriza o acesso ao seu repositório git. Para GitHub repositórios, o Amplify agora usa o recurso Apps para autorizar GitHub o acesso ao Amplify.

   Para obter mais informações sobre como instalar e autorizar o GitHub aplicativo, consulteConfigurando o acesso do Amplify aos repositórios GitHub.
5. Na página Adicionar ramificação do repositório, faça o seguinte:

   a. Na lista Repositórios atualizados recentemente, selecione o nome do repositório a ser conectado.

   b. Na lista Ramificação, selecione o nome da ramificação do repositório a ser conectada.

   c. Escolha Próximo.
6. Na página Definir configurações de segurança, escolha Próximo.
7. Na página Revisar, escolha Salvar e implantar. Depois de concluir a implantação, seu aplicativo poderá ser visualizado no domínio padrão `amplifyapp.com`.

> **Note**
>
> Para aumentar a segurança de seus aplicativos do Amplify, o domínio amplifyapp.com é registrado na Lista Pública de Sufixos (PSL). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para seus aplicativos do Amplify. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a Set-Cookiepágina na Rede de Desenvolvedores da Mozilla.

Etapa 2: criar um backend

Agora que você implantou um aplicativo de frontend no Amplify Hosting, é possível criar um backend. Use as instruções a seguir para criar um backend com um banco de dados simples e endpoint da API do GraphQL.

Para criar um backend

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todos os aplicativos, selecione o aplicativo que você criou na Etapa 1.
3. Na página inicial do aplicativo, escolha a guia Ambientes de backend e escolha Conceitos básicos. Isso inicia o processo de configuração de um ambiente de teste padrão.
4. Depois que a configuração for concluída, escolha o Iniciar Studio para acessar o ambiente de backend de teste padrão no Amplify Studio.

O Amplify Studio é uma interface visual para criar e gerenciar seu backend e acelerar o desenvolvimento de sua interface de usuário de frontend. Para obter mais informações sobre como usar o Amplify Studio, consulte a documentação do Amplify Studio.

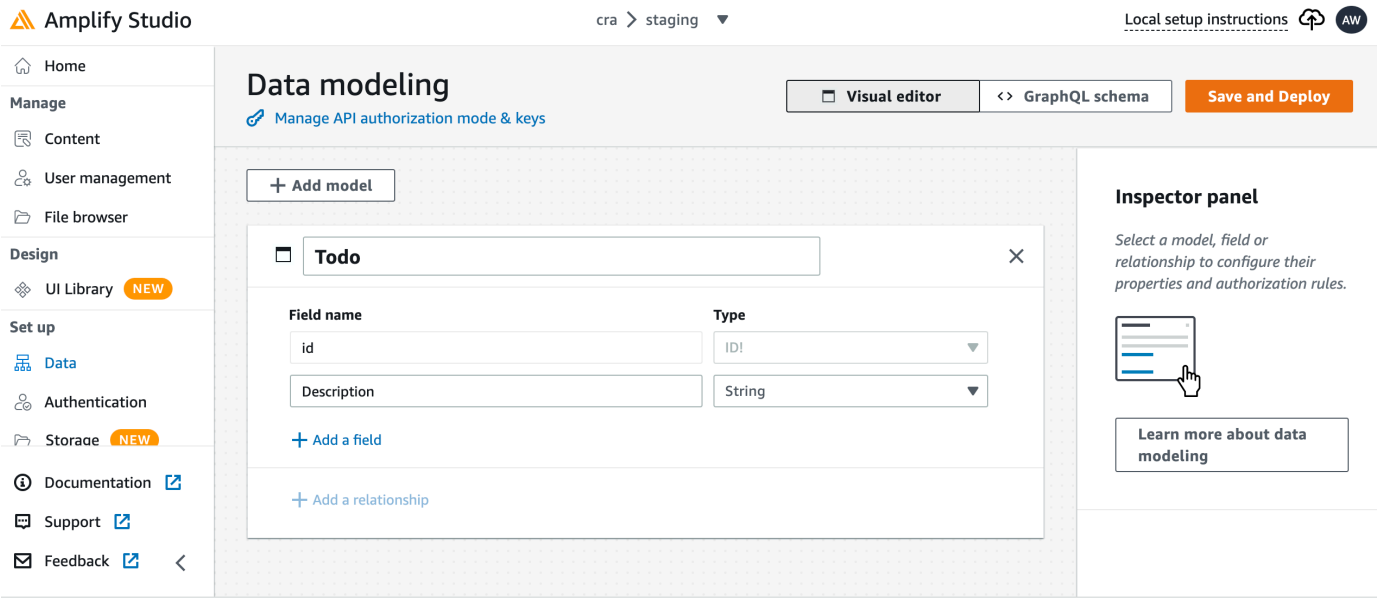
Use as instruções a seguir para criar um banco de dados simples usando a interface do construtor visual de backend do Amplify Studio.

Criar um modelo de dados

1. Na página inicial do ambiente de teste padrão do seu aplicativo, escolha Criar modelo de dados. Isso abre o designer do modelo de dados.
2. Na página Modelagem de dados, escolha Adicionar modelo.

3. No título, insira **`Todo`**.
4. Escolha Adicionar um campo.
5. Em Nome do campo, insira **`Description`**.

   A captura de tela a seguir é um exemplo de como seu modelo de dados será exibido no designer.



6. Escolha Salvar e implantar.
7. Retorne ao console do Amplify Hosting e a implantação do ambiente de teste padrão estará em andamento.

Durante a implantação, o Amplify Studio cria todos os AWS recursos necessários no back-end, incluindo uma API GraphQL para acessar dados e uma AWS AppSync tabela do Amazon DynamoDB para hospedar os itens do Todo. O Amplify usa CloudFormation para implantar seu back-end, o que permite que você armazene sua definição de back-end como infraestrutura como código.

Etapa 3: Conectar o backend ao frontend

Agora que você implantou um frontend e criou um backend em nuvem que contém um modelo de dados, você precisa conectá-los. Use as instruções a seguir para reduzir sua definição de backend ao seu projeto de aplicativo local com a Amplify CLI.

Para conectar um backend de nuvem a um frontend local

1. Abra uma janela de terminal e navegue até o diretório raiz do seu projeto local.

2. Execute o comando a seguir na janela do terminal, substituindo o texto em vermelho pelo ID exclusivo do aplicativo e pelo nome do ambiente de backend do seu projeto.

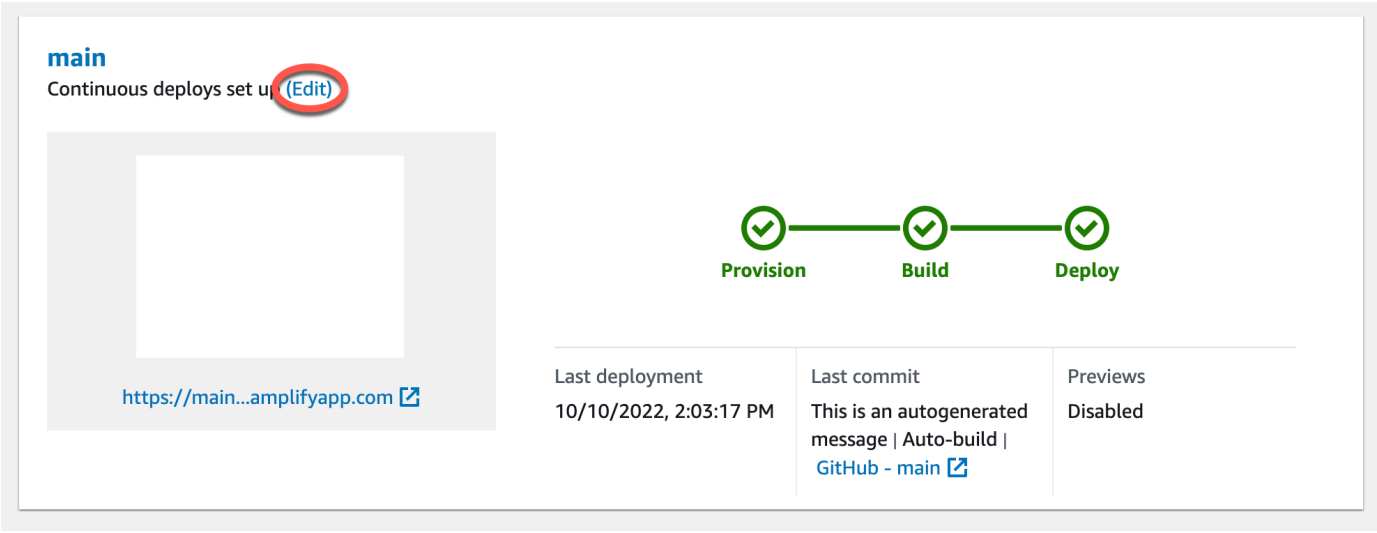
```
amplify pull --appId abcd1234 --envName staging
```

3. Siga as instruções na janela do terminal para concluir a configuração do projeto.

Agora é possível configurar o processo de compilação para adicionar o backend ao fluxo de trabalho de implantação contínua. Use as instruções a seguir para conectar uma ramificação de frontend a um backend no console do Amplify Hosting.

Para conectar uma ramificação de aplicativos de frontend e um backend de nuvem

1. Na página inicial do aplicativo, escolha a guia Ambientes de hospedagem.
2. Localize a ramificação principal e escolha Editar.



3. Na janela Editar backend de destino, em Ambiente, selecione o nome do backend a ser conectado. Neste exemplo, escolha o backend de teste padrão que você criou na Etapa 2.

   Por padrão, o full-stack CI/CD está habilitado. Desmarque essa opção para desativar a pilha completa CI/CD desse back-end. Desativar o full-stack CI/CD faz com que o aplicativo seja executado no modo pull only. No momento da compilação, o Amplify gerará automaticamente somente o arquivo `aws-exports.js`, sem modificar seu ambiente de backend.

4. Em seguida, você deve configurar um perfil de serviço para dar ao Amplify as permissões necessárias para fazer alterações no backend do seu aplicativo. É possível usar um perfil de

serviço existente ou criar outro. Para instruções, consulte Adição de um perfil de serviço com permissões para implantar recursos de backend.

5. Depois de adicionar um perfil de serviço, retorne à janela Editar backend de destino e escolha Salvar.
6. Para concluir a conexão do backend de teste padrão à ramificação principal do aplicativo de frontend, execute uma nova compilação do seu projeto.

   Execute um destes procedimentos:

   - Do seu repositório git, envie algum código para iniciar uma compilação no console do Amplify.
   - No console do Amplify, navegue até a página de detalhes da versão do aplicativo e escolha Reimplantar esta versão.

Próximas etapas

Configurar implantações de ramificação de atributos

Siga nosso fluxo de trabalho recomendado para configurar implantações de ramificações de atributos com vários ambientes de backend.

Crie uma interface de usuário de frontend no Amplify Studio

Use o Studio para criar sua UI de frontend com um conjunto de componentes de UI prontos para uso e, em seguida, conecte-o ao backend do seu aplicativo. Para obter mais informações e tutoriais, consulte o guia do usuário do Amplify Studio na documentação do Amplify Framework.

Recursos avançados de implantação

Este capítulo aborda recursos avançados de implantação que aprimoram seu fluxo de trabalho do Amplify Hosting. Esses recursos fornecem controles e capacidades adicionais para ajudar as equipes a gerenciarem as implantações com mais eficiência, garantirem a qualidade do código e manterem a segurança durante todo o ciclo de vida do desenvolvimento.

Saiba como proteger suas ramificações de recursos com autenticação por senha para restringir o acesso a recursos não lançados. Ative as visualizações na web para pull requests para revisar as alterações em uma visualização exclusiva URLs antes de mesclar o código às ramificações de produção. Configure o end-to-end teste usando a estrutura Cypress para capturar regressões antes de enviar o código para a produção. Embora o recurso do botão Implantar no Amplify não esteja mais disponível, você ainda pode implantar facilmente as aplicações diretamente do seu repositório usando o Amplify Hosting.

Tópicos

Restrição de acesso a ramificações de uma aplicação do Amplify

Se você estiver trabalhando em atributos não lançados, poderá proteger com senha as ramificações de atributos para restringir o acesso. Quando o controle de acesso é definido em uma ramificação, os usuários são solicitados a fornecer um nome de usuário e uma senha quando tentam acessar o URL da ramificação.

É possível definir uma senha que se aplique a uma ramificação individual ou globalmente a todas as ramificações conectadas. Quando o controle de acesso estiver habilitado no nível global e na ramificação, a senha no nível da ramificação terá precedência sobre uma senha de nível global (aplicação).

O Amplify controla a utilização de solicitações com falha que estejam tentando acessar recursos protegidos por senha. Esse comportamento protege as aplicações contra ataques de dicionário ou outras tentativas de ler dados por trás dos controles de acesso.

Use o procedimento a seguir para definir uma senha para restringir o acesso às ramificações de uma aplicação do Amplify.

Para definir senhas em ramificações de atributos

1. Faça login Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo no qual você deseja definir as senhas da ramificação de atributos.
3. No painel de navegação, escolha Hospedagem e escolha Controle de acesso.
4. Na seção Configurações de controle de acesso, escolha Gerenciar acesso.
5. Na página Gerenciar controle de acesso, faça uma das ações a seguir.
   - Para definir um nome de usuário e uma senha que se apliquem a todas as ramificações conectadas
     - Ative Gerenciar acesso para todas as ramificações. Por exemplo, se você tiver as ramificações main, dev, e feature conectadas, poderá aplicar o mesmo nome de usuário e senha para todas as ramificações.
   - Para definir um nome de usuário e uma senha que se apliquem a uma ramificação individual
     a. Desative Gerenciar acesso para todas as ramificações.
     b. Localize a ramificação que você deseja gerenciar. Em Configurações de acesso, escolha Senha restrita necessária.
     c. Em Nome de usuário, insira um nome de usuário.
     d. Em Senha, insira uma senha.
   - Escolha Salvar.
6. Se você estiver gerenciando o controle de acesso para um aplicativo renderizado do lado do servidor (SSR), reimplante o aplicativo executando uma nova compilação a partir do seu repositório Git. Essa etapa é necessária para permitir que o Amplify aplique suas configurações de controle de acesso.

Pré-visualizações na web para solicitações pull

As visualizações na Web oferecem às equipes de desenvolvimento e garantia de qualidade (QA) uma maneira de visualizar as alterações das pull requests (PRs) antes de mesclar o código a uma ramificação de produção ou integração. As solicitações pull permitem que você informe outras

pessoas sobre as alterações que você enviou para uma ramificação em um repositório. Depois que uma solicitação pull é aberta, é possível discutir e revisar as possíveis alterações com os colaboradores e adicionar confirmações de acompanhamento antes que suas alterações sejam mescladas na ramificação base.

Uma visualização prévia da web implanta cada solicitação pull feita em seu repositório em um URL de visualização exclusiva que é completamente diferente do URL que seu site principal usa. Para aplicações com ambientes de backend provisionados usando a CLI do Amplify ou o Amplify Studio, cada solicitação pull (somente repositórios Git privados) cria um backend temporário que é excluído quando a PR é fechada.

Quando as pré-visualizações na Web são ativadas para sua aplicação, cada PR conta para a cota do Amplify de 50 ramificações por aplicação. Para evitar exceder essa cota, certifique-se de fechar sua. PRs Para obter mais informações sobre cotas, consulte Service quotas do Amplify Hosting.

> **Note**
>
> Atualmente, a variável de `AWS_PULL_REQUEST_ID` ambiente não está disponível quando usada AWS CodeCommit como seu provedor de repositório.

Segurança de pré-visualização na Web

Para fins de segurança, é possível ativar as pré-visualizações na Web em todas as aplicações com repositórios privados, mas não em todas as aplicações com repositórios públicos. Se seu repositório Git for público, é possível configurar visualizações somente para aplicativos que não exigem um perfil de serviço do IAM. Por exemplo, aplicativos com backends e aplicativos implantados na plataforma de `WEB_COMPUTE` hospedagem exigem um perfil de serviço do IAM. Portanto, não é possível habilitar visualizações na web para esses tipos de aplicativos se o repositório deles for público. O Amplify aplica essa restrição para impedir que terceiros enviem código arbitrário que seria executado usando as permissões de perfil do IAM do seu aplicativo.

Quando as pré-visualizações na Web são habilitadas para uma aplicação em um repositório público com um perfil SSR Compute, você precisa gerenciar cuidadosamente quais ramificações podem acessar a função. Recomendamos que você não use um perfil no nível da aplicação. Em vez disso, é necessário anexar um perfil de computação no nível da ramificação. Isso permite que você conceda permissões somente às ramificações que exijam acesso a recursos específicos. Para obter mais informações, consulte Adicionar uma função SSR Compute para permitir o acesso aos recursos AWS.

Habilitação de pré-visualizações na Web para solicitações de pull

Para aplicativos armazenados em um GitHub repositório, as visualizações na web usam o GitHub aplicativo Amplify para acesso ao repositório. Se você estiver habilitando visualizações na web em um aplicativo Amplify existente que você implantou anteriormente a partir de GitHub um repositório OAuth usando para acesso, primeiro você deve migrar o aplicativo para usar o aplicativo Amplify. GitHub Para obter instruções de migração, consulte Migração de um OAuth aplicativo existente para o aplicativo Amplify GitHub.

Para habilitar visualizações na web para solicitações pull

1. Escolha Hospedagem e, em seguida, Pré-visualizações.

   > **Note**
   >
   > As visualizações são visíveis no menu Configurações do aplicativo somente quando um aplicativo é configurado para implantação contínua e conectado a um repositório git. Para obter instruções sobre esse tipo de implantação, consulte Introdução ao código existente.

2. Somente para GitHub repositórios, faça o seguinte para instalar e autorizar o GitHub aplicativo Amplify em sua conta:

   a. Na janela Instalar GitHub aplicativo para ativar visualizações, escolha Instalar GitHub aplicativo.

   b. Selecione a GitHub conta na qual você deseja configurar o aplicativo Amplify GitHub.

   c. Uma página é aberta em GitHub.com para configurar as permissões do repositório para sua conta.

   d. Execute um destes procedimentos:

      - Para aplicar a instalação a todos os repositórios, escolha Todos os repositórios.
      - Para limitar a instalação aos repositórios específicos que você selecionar, escolha Somente selecionar repositórios. Certifique-se de incluir o repositório do aplicativo para o qual você está habilitando as visualizações da web nos repositórios que você selecionar.

   e. Escolha Salvar

3. Depois de ativar as visualizações para seu repositório, retorne ao console do Amplify para ativar as visualizações de ramificações específicas. Na página Pré-visualizações, selecione uma ramificação na lista e escolha Editar configurações.
4. Na página Gerenciar configurações de pré-visualização, ative Pré-visualizações de solicitações pull. Depois, selecione Confirm (Confirmar).
5. Para aplicativos full-stack, realize uma das seguintes ações:
   - Escolha Criar um novo ambiente de backend para cada solicitação pull. Essa opção permite testar as alterações sem afetar a produção.
   - Escolha Apontar todas as solicitações pull dessa ramificação para um ambiente existente.
6. Escolha Confirmar.

Na próxima vez que você enviar uma solicitação pull para a filial, o Amplify cria e implanta seu PR em um URL de pré-visualização. Depois que a solicitação pull é fechada, o URL de visualização é excluído e qualquer ambiente de backend temporário vinculado à solicitação pull é excluído. Somente para GitHub repositórios, você pode acessar uma prévia do seu URL diretamente da pull request em sua GitHub conta.

Acesso à pré-visualização na web com subdomínios

As pré-visualizações na Web para solicitações pull podem ser acessadas com subdomínios para uma aplicação do Amplify conectada a um domínio personalizado gerenciado pelo Amazon Route 53. Quando a solicitação pull é fechada, as ramificações e subdomínios associados à solicitação pull são excluídos automaticamente. Esse é o comportamento padrão para visualizações na web depois de configurar implantações de ramificações de atributos baseadas em padrões para seu aplicativo. Para obter instruções de configuração de subdomínios automáticos, consulte Configuração de subdomínios automáticos para um domínio personalizado do Amazon Route 53.

Configurando testes end-to-end Cypress para seu aplicativo Amplify

Você pode executar testes end-to-end (E2E) na fase de teste do seu aplicativo Amplify para capturar regressões antes de enviar o código para produção. A fase de teste pode ser configurada na especificação de compilação YAML. Atualmente, é possível executar apenas a estrutura de teste do Cypress durante uma compilação.

O Cypress é uma estrutura de teste JavaScript baseada que permite executar testes E2E em um navegador. Para um tutorial que demonstra como configurar testes E2E, consulte a postagem do blog Executando testes end-to-end Cypress para sua implantação completa com o Amplify. CI/CD

Adição de testes do Cypress a uma aplicação do Amplify existente

É possível adicionar testes Cypress a um aplicativo existente atualizando as configurações da compilação do aplicativo no console do Amplify. O YAML de especificação de compilação contém uma coleção de comandos de compilação e configurações relacionadas que o Amplify usa para executar a compilação. Use a etapa `test` para executar qualquer comando de teste no momento da compilação. Para testes E2E, o Amplify Hosting oferece uma integração mais profunda com o Cypress, que permite gerar um relatório de interface do usuário para seus testes.

A lista a seguir descreve as configurações do teste e como elas são usadas.

pré-teste

Instale as dependências necessárias para executar os testes do Cypress. O Amplify Hosting usa o mochawesome para gerar um relatório para visualizar os resultados do teste e aguardar a configuração do servidor localhost durante a compilação.

teste

Execute comandos cypress para realizar testes usando mochawesome.

pós-teste

O relatório mochawesome é gerado a partir do JSON de saída. Observe que, se você estiver usando o Yarn, deverá executar esse comando no modo silencioso para gerar o relatório mochawesome. Para Yarn, é possível usar o seguinte comando:

```
yarn run --silent mochawesome-merge cypress/report/mochawesome-report/
mochawesome*.json > cypress/report/mochawesome.json
```

artifacts>baseDirectory

O diretório a partir do qual os testes são executados.

artefatos> configFilePath

Os dados do relatório de teste gerados.

artifacts>files

Os artefatos gerados (capturas de tela e vídeos) estão disponíveis para download.

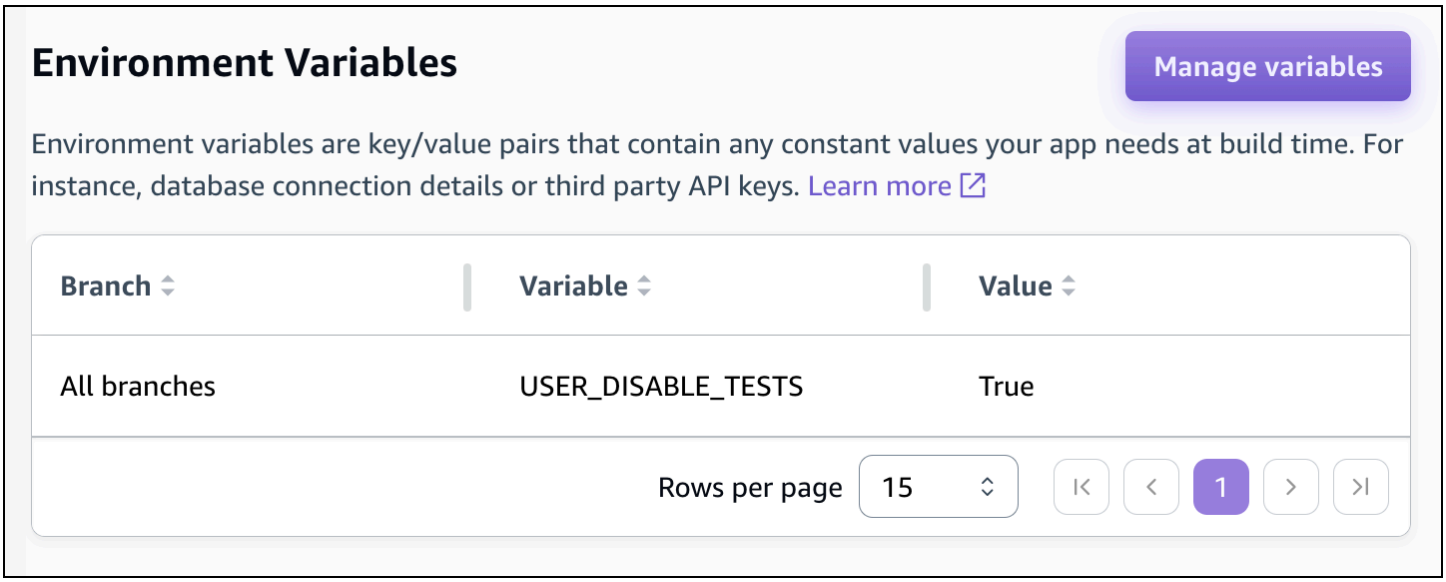
O exemplo a seguir, trecho de um arquivo `amplify.yml` de especificação de compilação mostra como adicionar testes Cypress ao seu aplicativo.

```
test:
  phases:
    preTest:
      commands:
        - npm ci
        - npm install -g pm2
        - npm install -g wait-on
        - npm install mocha mochawesome mochawesome-merge mochawesome-report-generator
        - pm2 start npm -- start
        - wait-on http://localhost:3000
    test:
      commands:
        - 'npx cypress run --reporter mochawesome --reporter-options
 "reportDir=cypress/report/mochawesome-
report,overwrite=false,html=false,json=true,timestamp=mmddyyyy_HHMMss"'
    postTest:
      commands:
        - npx mochawesome-merge cypress/report/mochawesome-report/mochawesome*.json >
 cypress/report/mochawesome.json
        - pm2 kill
  artifacts:
    baseDirectory: cypress
    configFilePath: '**/mochawesome.json'
    files:
      - '**/*.png'
      - '**/*.mp4'
```

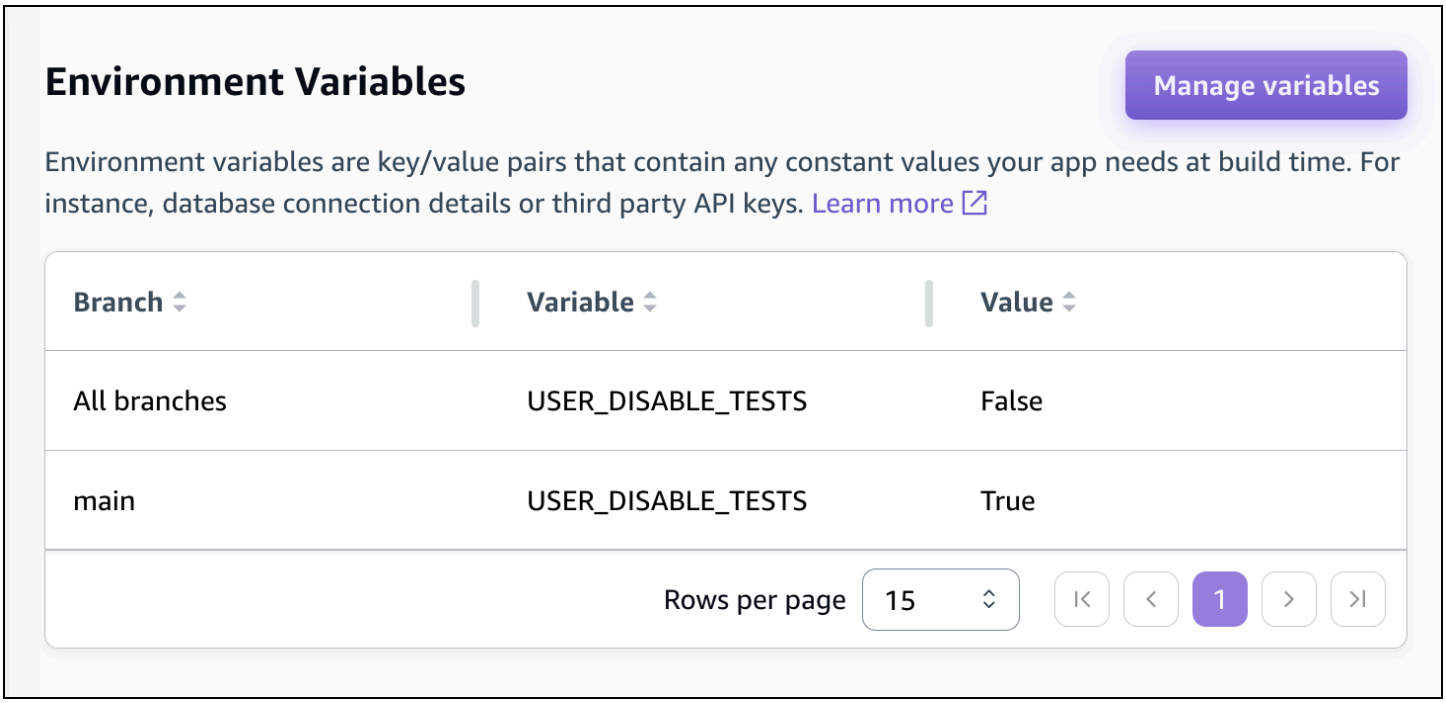
Desativação de testes para uma aplicação ou ramificação do Amplify

Depois que a configuração de teste for adicionada às suas configurações da compilação `amplify.yml`, a etapa `test` será executada em cada compilação, em cada ramificação. Se você quiser desabilitar globalmente a execução de testes ou executar testes apenas para ramificações específicas, é possível usar a variável de ambiente USER_DISABLE_TESTS sem modificar suas configurações da compilação.

Para desativar globalmente os testes para todas as ramificações, adicione a variável de USER_DISABLE_TESTS ambiente com um valor de `true` para todas as ramificações. A captura de tela a seguir mostra a seção Variáveis de ambiente no console do Amplify com os testes desativados para todas as ramificações.



Para desativar os testes para uma ramificação específica, adicione a variável de USER_DISABLE_TESTS ambiente com um valor de `false` para todas as ramificações e, em seguida, adicione uma substituição para cada ramificação que você deseja desativar com um valor de `true`. Na captura de tela a seguir, os testes são desativados na ramificação principal e habilitados para todas as outras ramificações.



A desativação dos testes com essa variável fará com que a etapa de teste seja totalmente ignorada durante uma compilação. Para reativar os testes, defina esse valor como ou exclua a variável de ambiente `false`.

Usando o botão Implantar para Amplificar para compartilhar um projeto GitHub

> **Important**
>
> A implantação de um clique usando o botão Implantar no Amplify Hosting não está mais disponível. Para implantar a partir de um repositório, crie uma nova aplicação no Amplify Hosting. Para instruções, consulte Noções básicas da implantação de uma aplicação no Amplify Hosting.

O botão Implantar no Amplify Hosting permite que você compartilhe GitHub projetos publicamente ou dentro de sua equipe. A seguir está uma imagem do botão:

Adição do botão Implantar no Amplify Hosting a um repositório ou blog

Adicione o botão ao seu arquivo GitHub README.md, postagem do blog ou qualquer outra página de marcação que renderize HTML. O botão tem os seguintes componentes:

1. Uma imagem SVG localizada no URL `https://oneclick.amplifyapp.com/button.svg`
2. O URL do console do Amplify com um link para seu GitHub repositório. É possível copiar a URL do seu repositório, por exemplo `https://github.com/username/repository`, ou fornecer um link direto para uma pasta específica, como `https://github.com/username/repository/tree/branchname/folder`. O Amplify Hosting implantará a ramificação padrão no seu repositório. Ramificações adicionais podem ser conectadas depois que o aplicativo é conectado.

Use o exemplo a seguir para adicionar o botão a um arquivo markdown, como seu GitHub README.md. Substitua `https://github.com/username/repository` pelo URL do seu repositório.

```
[![amplifybutton](https://oneclick.amplifyapp.com/button.svg)](https://
console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/username/
repository)
```

Use o exemplo a seguir para adicionar o botão a qualquer documento HTML. Substitua `https://github.com/username/repository` pelo URL do seu repositório.

```
<a href="https://console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/
username/repository">
    <img src="https://oneclick.amplifyapp.com/button.svg" alt="Deploy to Amplify
 Hosting">
</a>
```

Configuração de redirecionamentos e regravações para uma aplicação do Amplify

Os redirecionamentos permitem que um servidor web redirecione a navegação de um URL para outro. As razões comuns para usar redirecionamentos incluem: personalizar a aparência de um URL, evitar links quebrados, mover o local de hospedagem de um aplicativo ou site sem alterar seu endereço e alterar um URL solicitado para a forma exigida por um aplicativo web.

Noções básicas dos redirecionamentos com suporte no Amplify

O Amplify suporta os seguintes tipos de redirecionamento no console.

Redirecionamento permanente (301)

Os redirecionamentos 301 são destinados a alterações duradouras ao destino de um endereço web. O histórico de classificação do mecanismo de busca do endereço original se aplica ao novo endereço de destino. O redirecionamento ocorre do lado do cliente, portanto, a barra de navegação de um navegador exibirá o endereço de destino após o redirecionamento.

Os motivos comuns para usar redirecionamentos 301 incluem:

- Evitar um link quebrado quando o endereço de uma página é alterado.
- Evitar um link quebrado quando um usuário comete um erro de digitação previsível em um endereço.

Redirecionamento temporário (302)

Os redirecionamentos 302 são destinados a alterações temporárias ao destino de um endereço web. O histórico de classificação do mecanismo de busca do endereço original não se aplica ao novo endereço de destino. O redirecionamento ocorre do lado do cliente, portanto, a barra de navegação de um navegador exibirá o endereço de destino após o redirecionamento.

Os motivos comuns para usar redirecionamentos 302 incluem:

- Fornecer um destino de desvio enquanto estão sendo feitos reparos no endereço original.
- Fornecer páginas de teste para A/B comparação de uma interface de usuário.

> **Note**
>
> Se seu aplicativo estiver retornando uma resposta 302 inesperada, o erro provavelmente foi causado por alterações que você fez no redirecionamento e na configuração do cabeçalho personalizado do seu aplicativo. Para resolver esse problema, verifique se seus cabeçalhos personalizados são válidos e reative a regra de regravação 404 padrão para seu aplicativo.

Regravação (200)

Os redirecionamentos 200 (regravações) são destinados a mostrar conteúdo do endereço de destino como se ele fosse fornecido pelo endereço original. O histórico de classificação do mecanismo de busca continua a ser aplicado ao endereço original. O redirecionamento ocorre do lado do servidor, portanto, a barra de navegação de um navegador exibirá o endereço original após o redirecionamento. Os motivos comuns para usar redirecionamentos 200 incluem:

- Redirecionar todo um site para um novo local de hospedagem sem alterar o endereço do site.
- Redirecionar todo o tráfego para um aplicativo de única página web (SPA) para sua página index.html para manipulação por uma função de roteamento do lado do cliente.

Não encontrado (404)

Os redirecionamentos 404 ocorrem quando uma solicitação aponta para um endereço que não existe. A página de destino de um 404 é exibida em vez da página solicitada. Os motivos comuns para que ocorra um redirecionamento 404 incluem:

- Evitar uma mensagem de link quebrado quando um usuário entra em um URL inválido.
- Apontar solicitações de páginas não existentes de um aplicativo web para sua página index.html para manipulação por uma função de roteamento do lado do cliente.

Noções básicas da ordem dos redirecionamentos

Os redirecionamentos são aplicados da parte superior para a parte inferior da lista. Verifique se a sua solicitação tem o efeito pretendido. Por exemplo, a seguinte ordem de redirecionamentos faz com que todas as solicitações de determinado caminho em /docs/ sejam redirecionadas ao mesmo

caminho em /documents/, exceto /docs/specific-filename.html, que é redirecionado para documents/different-filename.html:

```
/docs/specific-filename.html /documents/different-filename.html 301
/docs/<*> /documents/<*>
```

A seguinte ordem de redirecionamentos ignora o redirecionamento de specific-filename.html para different-filename.html:

```
/docs/<*> /documents/<*>
/docs/specific-filename.html /documents/different-filename.html 301
```

Noções básicas de como o Amplify encaminha os parâmetros de consulta

É possível usar parâmetros de consulta para ter mais controle sobre suas correspondências de URL. O Amplify encaminha todos os parâmetros de consulta para o caminho de destino dos redirecionamentos 301 e 302, com as seguintes exceções:

- Se o endereço original incluir uma sequência de caracteres de consulta definida com um valor específico, o Amplify não encaminhará os parâmetros de consulta. Nesse caso, o redirecionamento se aplica somente às solicitações para o URL de destino com o valor de consulta especificado.
- Se o endereço de destino da regra correspondente tiver parâmetros de consulta, os parâmetros de consulta não serão encaminhados. Por exemplo, se o endereço de destino do redirecionamento for `https://example-target.com?q=someParam`, os parâmetros de consulta não serão transmitidos.

Criação e edição de redirecionamentos no console do Amplify

É possível criar e editar redirecionamentos para uma aplicação no console do Amplify. Antes de começar, você precisará das seguintes informações sobre as partes de um redirecionamento.

Um endereço original

O endereço solicitado pelo usuário.

Um endereço de destino

O endereço que realmente fornece o conteúdo que o usuário vê.

Um tipo de redirecionamento

Os tipos incluem um redirecionamento permanente (301), um redirecionamento temporário (302), uma regravação (200) ou não encontrado (404).

Um código de país de duas letras (opcional)

Um valor que é possível incluir para segmentar a experiência do usuário do seu aplicativo por região geográfica.

Para criar um redirecionamento no console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual você deseja criar um redirecionamento.
3. No painel de navegação, escolha Hospedagem e, em seguida, escolha Regravações e redirecionamentos.
4. Na página Regravações e redirecionamentos, escolha Gerenciar redirecionamentos.
5. Adicione ou atualize manualmente os redirecionamentos no editor JSON de Regravações e redirecionamentos.
   a. Em `source`, especifique o endereço que o usuário solicitou.
   b. Em `status`, especifique o tipo de redirecionamento.
   c. Em `target`, especifique o endereço de destino que renderiza o conteúdo para o usuário.
   d. (Opcional) Em `condition`, insira uma condição de código do país de duas letras.
6. Escolha Salvar.

Referência de exemplo de redirecionamentos e regravações

Esta seção fornece exemplos para vários cenários comuns de redirecionamento.

> **Important**
>
> Os redirecionamentos específicos do domínio não oferecem suporte a componentes de caminho no campo de origem.

Suportado:

- `"source": "https://example.com"` (caminhos anexados automaticamente)

Sem suporte:

- `"source": "https://example.com/specific-path"`
- No momento, não há suporte para regras com combinações `domain+path`.

Padrões alternativos
Para redirecionamentos de caminho específicos do domínio, use:

1. Regras separadas somente de domínio (os caminhos são anexados automaticamente)
2. Regras somente de caminho com lógica condicional
3. Várias combinações de regras

É possível usar esses exemplos para entender a sintaxe de JSON para a criação dos seus próprios redirecionamentos e reescritas no editor de JSON do console do Amplify.

> **Note**
>
> A correspondência de domínio do endereço original não diferencia maiúsculas de minúsculas.

Tópicos

Redirecionamentos e regravações simples

É possível usar o exemplo a seguir para redirecionar permanentemente uma página específica a um novo endereço.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`/original.html`	`/destination.html`	`permanent redirect (301)`	

Formato JSON

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

É possível usar o exemplo a seguir para redirecionar qualquer caminho em uma pasta para o mesmo caminho em uma pasta diferente.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

Formato JSON

```
[
  {
    "source": "/docs/<*>",
    "status": "301",
    "target": "/documents/<*>",
```

```
    "condition": null
  }
]
```

É possível usar o exemplo a seguir para redirecionar todo o tráfego para index.html como uma reescrita. Nesse cenário, a regravação faz parecer para o usuário que ele está no endereço original.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
/<*>	/index.html	rewrite (200)	

Formato JSON

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

É possível usar o exemplo a seguir para usar uma reescrita para alterar o subdomínio que é exibido ao usuário.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
https://mydomain.com	https://www.mydomain.com	rewrite (200)	

Formato JSON

```
[
  {
    "source": "https://mydomain.com",
```

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

É possível usar o exemplo a seguir para redirecionar para um domínio diferente com um prefixo de caminho.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`https://mydomain.com`	`https://www.mydomain.com/documents`	`temporary redirect (302)`	

Formato JSON

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

É possível usar o exemplo a seguir para redirecionar caminhos em uma pasta que não podem ser encontrados para uma página 404 personalizada.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`/<*>`	`/404.html`	`not found (404)`	

Formato JSON

```
[
```

```
  {
    "source": "/<*>",
    "status": "404",
    "target": "/404.html",
    "condition": null
  }
]
```

> **Important**
>
> Não há suporte para os componentes de caminho em regras de origem baseadas em domínio (como "`https://domain.com/path`"), e eles farão com que a regra seja ignorada sem erros.

Redireciona para aplicativos de única página web (SPA)

A maioria dos frameworks SPA suporta HTML5 history.pushState () para alterar a localização do navegador sem iniciar uma solicitação do servidor. Isso funciona para usuários que começam a jornada a partir da raiz (ou /index.html), mas falha para usuários que navegam diretamente para qualquer outra página.

O exemplo a seguir usa expressões regulares para configurar uma reescrita 200 para todos os arquivos em index.html, exceto para as extensões de arquivo especificadas na expressão regular.

Endereço original	Endereço de destino	Tipo de redirecio namento	Código do país
`</^[^.]+$\|\.(?!(css\|gif\|ico\|jpg\|js\|png\|txt\|svg\|woff\|woff2\|ttf\|map\|json\|webp)$)([^.]+$)/>`	`/index.html`	`200`	

Formato JSON

```
[
  {
    "source": "</^[^.]+$|\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|
webp)$)([^.]+$)/>",
    "status": "200",
    "target": "/index.html",
    "condition": null
  }
]
```

Regravação de proxy reverso

O exemplo a seguir usa uma regravação para criar um proxy de conteúdo de outro local para que pareça ao usuário que o domínio não foi alterado. O HTTPS é o único protocolo com suporte para proxies reversos.

Endereço original	Endereço de destino	Tipo de redirecio namento	Código do país
`/images/<*>`	`https://images.otherdomain.com/<*>`	`rewrite (200)`	

Formato JSON

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

Arrastando, cortando e limpando URLs

Para criar estruturas limpas de URL, como about, em vez de about.html, geradores de site estático, como Hugo, geram diretórios para páginas com um index.html (/about/index.html). O Amplify cria

automaticamente uma barra limpa URLs adicionando uma barra final quando necessário. A tabela a seguir destaca diferentes cenários:

Entradas do usuário no navegador	URL na barra de endereços	Documento fornecido
`/about`	`/about`	`/about.html`
`/about (when about.html returns 404)`	`/about/`	`/about/index.html`
`/about/`	`/about/`	`/about/index.html`

Espaços reservados

É possível usar o exemplo a seguir para redirecionar caminhos em uma estrutura de pastas a uma estrutura correspondente em outra pasta.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`/docs/<year>/<month>/<date>/<itemid>`	`/documents/<year>/<month>/<date>/<itemid>`	`permanent redirect (301)`	

Formato JSON

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

Strings de consulta e parâmetros de caminho

> **Warning**
>
> Não inclua segredos, credenciais ou dados confidenciais URLs como caminho ou parâmetros de consulta. Esses valores podem ser visualizados em texto simples nos logs de acesso da sua aplicação do Amplify.

É possível usar o exemplo a seguir para redirecionar um caminho para uma pasta com um nome que corresponde ao valor de um elemento de string de consulta no endereço original:

Endereço original	Endereço de destino	Tipo de redirecio namento	Código do país
`/docs?id=<my-blog-id-value`	`/documents/<my-blog-post-id-value>`	`permanent redirect (301)`	

Formato JSON

```
[
  {
    "source": "/docs?id=<my-blog-id-value>",
    "status": "301",
    "target": "/documents/<my-blog-id-value>",
    "condition": null
  }
]
```

> **Note**
>
> O Amplify encaminha todos os parâmetros da sequência de caracteres de consulta para o caminho de destino para redirecionamentos 301 e 302. No entanto, se o endereço original incluir uma sequência de caracteres de consulta definida com um valor específico, conforme demonstrado neste exemplo, o Amplify não encaminhará os parâmetros de consulta. Nesse

caso, o redirecionamento se aplica somente às solicitações para o endereço de destino com o valor de consulta `id` especificado.

É possível usar o exemplo a seguir para redirecionar todos os caminhos que não podem ser encontrados em determinado nível de uma estrutura de pastas para index.html em uma pasta especificada.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`/documents/<folder>/<child-folder>/<grand-child-folder>`	`/documents/index.html`	`not found (404)`	

Formato JSON

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

Redirecionamentos baseados em região

É possível usar o exemplo a seguir para redirecionar solicitações com base na região.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`/documents`	`/documents/us/`	`temporary redirect (302)`	`<US>`

Formato JSON

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

Uso de expressões curinga em redirecionamentos e regravações

É possível usar a expressão curinga, <*>, no endereço original para redirecionar ou regravar. É necessário colocar a expressão no final do endereço original, e ela deve ser exclusiva. O Amplify ignora os endereços originais que incluam mais de uma expressão curinga ou os usa em um posicionamento diferente.

Veja a seguir um exemplo de um redirecionamento válido com uma expressão curinga.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
/docs/<*>	/documents/<*>	permanent redirect (301)	

Os dois exemplos a seguir demonstram redirecionamentos inválidos com expressões curinga.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
/docs/<*>/content	/documents/<*>/content	permanent redirect (301)	
/docs/<*>/content/<*>	/documents/<*>/content/<*>	permanent redirect (301)	

Uso de variáveis de ambiente em uma aplicação Amplify

As variáveis de ambiente são pares de valores-chave que é possível adicionar às configurações do seu aplicativo para disponibilizá-las para o Amplify Hosting. Como uma melhor prática, é possível usar variáveis de ambiente para expor dados de configuração do aplicativo. Todas as variáveis de ambiente que você adiciona são criptografadas para impedir o acesso não autorizado.

O Amplify impõe as seguintes restrições às variáveis de ambiente que você cria.

- O Amplify não permite que você crie nomes de variáveis de ambiente com um prefixo `AWS`. Esse prefixo está reservado somente para uso interno do Amplify.
- O valor de uma variável de ambiente não pode exceder 5500 caracteres.

> **Important**
>
> Não use variáveis de ambiente para armazenar segredos. Para uma aplicação Gen 2, use o atributo Gerenciamento secreto no console do Amplify. Para obter mais informações, consulte Segredos e variáveis de ambiente na documentação do Amplify. Para um aplicativo de primeira geração, armazene segredos em um segredo de ambiente criado usando o AWS Systems Manager Parameter Store. Para obter mais informações, consulte Gerenciamento de segredos de ambiente.

Referência de variáveis de ambiente do Amplify

As seguintes variáveis de ambiente podem ser acessadas por padrão no console do Amplify.

Nome da variável	Description	Valor de exemplo
_BUILD_TIMEOUT	A duração do tempo limite de compilação em minutos. O valor mínimo é 5. O valor máximo é 120.	`30`

Nome da variável	Description	Valor de exemplo
_LIVE_UPDATES	A ferramenta será atualizada para a versão mais recente.	`[{"name":"Amplify CLI","pkg":"@aws-amplify/cli","type":"npm","version":"latest"}]`
USER_DISABLE_TESTS	A etapa de teste é ignorada durante uma compilação. É possível desativar os testes para todas as ramificações ou ramificações específicas em um aplicativo. Essa variável de ambiente é usada para aplicativos que realizam testes durante a fase de compilação. Para obter mais informações sobre a configuração dessa variável, consulte Desativação de testes para uma aplicação ou ramificação do Amplify.	`true`
AWS_APP_ID	O ID da compilação atual	`abcd1234`
AWS_BRANCH	O nome da ramificação da compilação atual	`main`, `develop`, `beta`, `v2.0`
AWS_BRANCH_ARN	O nome do recurso da Amazon (ARN) da ramificação da versão atual	`aws:arn:amplify:us-west-2:123456789012:appname/branch/...`
AWS_CLONE_URL	O URL de clone usado para buscar o conteúdo do repositório Git	`git@github.com:<user-name>/<repo-name>.git`

Nome da variável	Description	Valor de exemplo
AWS_COMMIT_ID	O ID de confirmação da compilação atual “HEAD” para recompilações	`abcd1234`
AWS_JOB_ID	O ID de trabalho da compilação atual. Isso inclui algum preenchimento com "0" para que sempre tenha o mesmo comprimento.	`0000000001`
AWS_PULL_REQUEST_ID	O ID da solicitação de pull da compilação de pré-visualização na Web. Essa variável de ambiente não está disponível ao ser usada AWS CodeCommit como seu provedor de repositório.	`1`
AWS_PULL_REQUEST_SOURCE_BRANCH	O nome da ramificação de recursos de uma pré-visualização de solicitação de pull enviada para uma ramificação da aplicação no console do Amplify.	`featureA`
AWS_PULL_REQUEST_DESTINATION_BRANCH	O nome da ramificação da aplicação no console do Amplify para a qual uma solicitação de pull de ramificação da aplicação recursos está sendo enviada.	`main`
AMPLIFY_AMAZON_CLIENT_ID	O ID do cliente da Amazon	`123456`

Nome da variável	Description	Valor de exemplo
AMPLIFY_AMAZON_CLIENT_SECRET	O segredo do cliente da Amazon	`example123456`
AMPLIFY_FACEBOOK_CLIENT_ID	O ID do cliente do Facebook	`123456`
AMPLIFY_FACEBOOK_CLIENT_SECRET	O segredo do cliente do Facebook	`example123456`
AMPLIFY_GOOGLE_CLIENT_ID	O ID do cliente do Google	`123456`
AMPLIFY_GOOGLE_CLIENT_SECRET	O segredo do cliente do Google	`example123456`
AMPLIFY_DIFF_DEPLOY	Ative ou desative a implantação de frontend baseada em diff. Para obter mais informações, consulte Configuração de compilação e implantação de frontend baseado em diff.	`true`
AMPLIFY_DIFF_DEPLOY_ROOT	O caminho a ser usado para comparações de implantação de frontend baseadas em diff, em relação à raiz do seu repositório.	`dist`

Nome da variável	Description	Valor de exemplo
AMPLIFY_DIFF_BACKEND	Ative ou desative as compilações de backend baseadas em diff. Isso se aplica somente às aplicações Gen 1. Para obter mais informações, consulte Configuração de compilações de backend baseadas em diff para uma aplicação Gen 1.	`true`
AMPLIFY_BACKEND_PULL_ONLY	O Amplify gerencia essa variável de ambiente. Isso se aplica somente às aplicações Gen 1. Para obter mais informações, consulte Edite um frontend existente para apontar para um backend diferente.	`true`
AMPLIFY_BACKEND_APP_ID	O Amplify gerencia essa variável de ambiente. Isso se aplica somente às aplicações Gen 1. Para obter mais informações, consulte Edite um frontend existente para apontar para um backend diferente.	`abcd1234`

Nome da variável	Description	Valor de exemplo
AMPLIFY_SKIP_BACKEND_BUILD	Se você não tiver uma seção de backend em sua especificação de compilação e quiser desativar as compilações de backend, defina essa variável de ambiente como `true`. Isso se aplica somente às aplicações Gen 1.	`true`
AMPLIFY_ENABLE_DEBUG_OUTPUT	Defina essa variável como `true` para imprimir um rastreamento de pilha nos logs. Isso é útil para depurar erros de compilação do backend.	`true`
AMPLIFY_MONOREPO_APP_ROOT	O caminho a ser usado para especificar a raiz do aplicativo monorepo, em relação à raiz do seu repositório.	`apps/react-app`
AMPLIFY_USERPOOL_ID	O ID do grupo de usuários do Amazon Cognito importado para autenticação	`us-west-2_example`
AMPLIFY_WEBCLIENT_ID	O ID do cliente do aplicativo a ser usado por aplicativos da web O cliente do aplicativo deve ser configurado com acesso ao grupo de usuários do Amazon Cognito especificado pela variável de ambiente AMPLIFY_USERPOOL_ID.	`123456`

Nome da variável	Description	Valor de exemplo
AMPLIFY_NATIVECLIENT_ID	O ID do cliente do aplicativo a ser usado por aplicativos nativos O cliente do aplicativo deve ser configurado com acesso ao grupo de usuários do Amazon Cognito especificado pela variável de ambiente AMPLIFY_USERPOOL_ID.	`123456`
AMPLIFY_IDENTITYPOOL_ID	O ID do banco de identidades do Amazon Cognito	`example-identitypool-id`
AMPLIFY_PERMISSIONS_BOUNDARY_ARN	O ARN para uso da política do IAM como limite de permissões que se aplica a todas os perfis do IAM criados pelo Amplify.	`arn:aws:iam::123456789012:policy/example-policy`
AMPLIFY_DESTRUCTIVE_UPDATES	Defina essa variável de ambiente como verdadeiro para permitir que uma API GraphQL seja atualizada com operações de esquema que podem potencialmente causar perda de dados.	`true`

> **Note**
>
> As variáveis de `AMPLIFY_AMAZON_CLIENT_SECRET` ambiente `AMPLIFY_AMAZON_CLIENT_ID` e são OAuth tokens, não uma chave de AWS acesso e uma chave secreta.

Variáveis de ambiente da estrutura de frontend

Se você estiver desenvolvendo seu aplicativo com uma estrutura de frontend que suporta suas próprias variáveis de ambiente, é importante entender que elas não são iguais às variáveis de ambiente que você configura no console do Amplify. Por exemplo, React (prefixado REACT_APP) e Gatsby (prefixado GATSBY) permitem que você crie variáveis de ambiente de runtime que essas estruturas agrupam automaticamente em sua compilação de produção de frontend. Para entender os efeitos do uso dessas variáveis de ambiente para armazenar valores, consulte a documentação da estrutura de frontend que você está usando.

Armazenar valores confidenciais, como chaves de API, dentro dessas variáveis de ambiente prefixadas da estrutura de frontend não é uma prática recomendada e é altamente desencorajado.

Configurar variáveis de ambiente

Use as instruções a seguir para definir variáveis de ambiente para uma aplicação no console do Amplify.

> **Note**
>
> As variáveis de ambiente são visíveis no menu de configurações do aplicativo do console Amplify somente quando um aplicativo é configurado para implantação contínua e conectado a um repositório git. Para obter instruções sobre esse tipo de implantação, consulte Introdução ao código existente.

Para definir variáveis de ambiente

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. No console do Amplify, escolha Hospedagem e, em seguida, escolha Variáveis de ambiente.
3. Na página Variáveis de ambiente, selecione Gerenciar variáveis de ambiente.
4. Em Variável, insira sua chave. Em Valor, insira seu valor. Por padrão, o Amplify aplica as variáveis de ambiente em todas as ramificações para que você não precise inseri-las novamente quando conecta uma nova ramificação.
5. (Opcional) Para personalizar uma variável de ambiente especificamente para uma ramificação, adicione uma substituição de ramificação da seguinte forma:

    a. Escolha Ações e Adicionar substituição de variável.

b. Agora, você tem um conjunto de variáveis de ambiente específicas para sua ramificação.

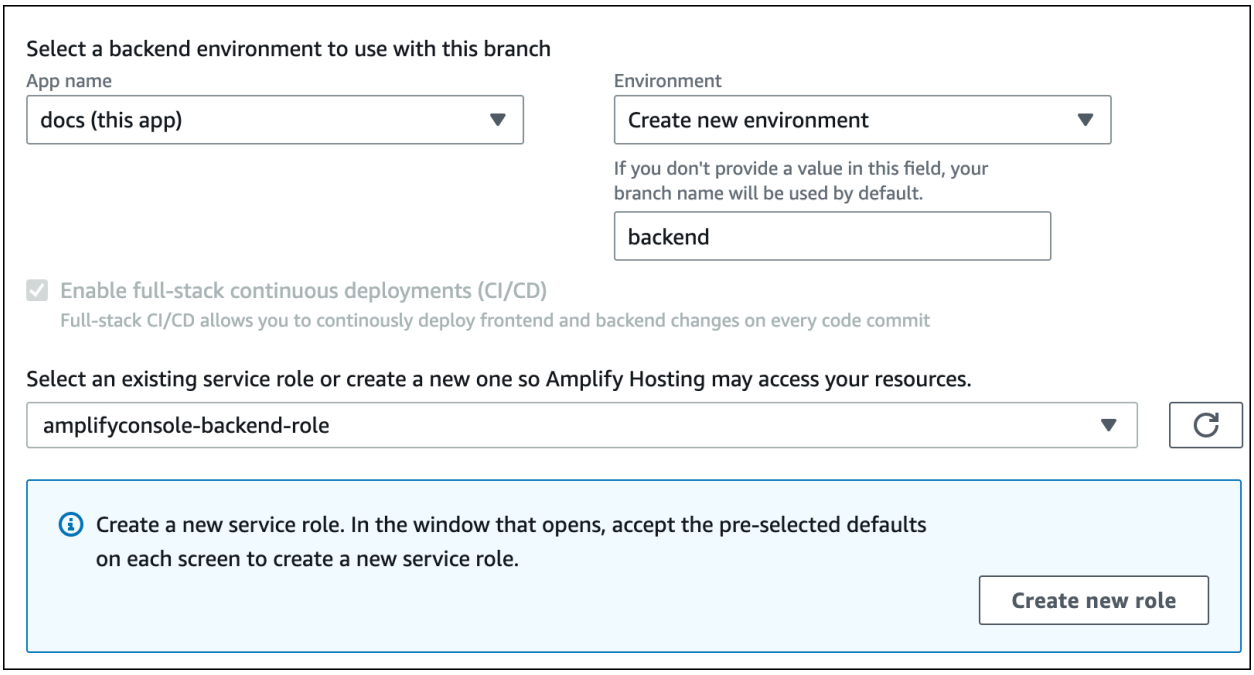
6. Escolha Salvar.

Crie um novo ambiente de backend com parâmetros de autenticação para login social

Para conectar uma ramificação a um aplicativo

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. O procedimento para conectar uma ramificação a um aplicativo varia dependendo se você está conectando uma ramificação a um novo aplicativo ou a um aplicativo existente.
   - Conectando uma ramificação a um novo aplicativo

     a. Na página Configurações de compilação, localize a seção Selecionar um ambiente de backend para usar com esta ramificação. Em Ambiente, escolha Criar novo ambiente e insira o nome do seu ambiente de backend. A captura de tela a seguir mostra a seção Selecionar um ambiente de backend para usar com esta ramificação da página Configurações de compilação com a inserção do nome **backend** do ambiente de backend.



     b. Expanda a seção Configurações avançadas na página Configurações de compilação e adicione variáveis de ambiente para chaves de login social. Por exemplo,

`AMPLIFY_FACEBOOK_CLIENT_SECRET` é uma variável de ambiente válida. Para ver a lista de variáveis de ambiente do sistema Amplify que estão disponíveis por padrão, consulte a tabela em Referência de variáveis de ambiente do Amplify.

- Conectando uma ramificação a um aplicativo existente

  a. Se você estiver conectando uma nova ramificação a um aplicativo existente, defina as variáveis do ambiente de login social antes de conectar a ramificação. No painel de navegação, escolha Configurações do aplicativo, Variáveis de ambiente.

  b. Na seção Variáveis de ambiente, escolha Gerenciar variáveis.

  c. Na seção Gerenciar variáveis, escolha Adicionar variável.

  d. Em Variável (chave), insira seu ID de cliente. Para Valor, insira o segredo do cliente.

  e. Escolha Salvar.

Gerenciamento de segredos de ambiente

Com o lançamento do Amplify Gen 2, o fluxo de trabalho de segredos de ambiente é simplificado para centralizar o gerenciamento de segredos e variáveis de ambiente no console do Amplify. Para obter instruções sobre como configurar e acessar segredos para uma aplicação Amplify Gen 2, consulte Segredos e variáveis de ambiente na Documentação do Amplify.

Os segredos de ambiente para uma aplicação Gen 1 são semelhantes às variáveis de ambiente, mas são pares de valores-chave do AWS Systems Manager Parameter Store que podem ser criptografados. Alguns valores devem ser criptografados, como a chave privada Sign in with Apple para Amplify.

Usando AWS Systems Manager para definir segredos de ambiente para um aplicativo Amplify Gen 1

Use as instruções a seguir para definir um segredo de ambiente para um aplicativo Amplify de primeira geração usando o AWS Systems Manager console.

Para definir um segredo de ambiente

1. Faça login no Console de gerenciamento da AWS e abra o AWS Systems Manager console.
2. No painel de navegação, escolha Gerenciamento de aplicativos e, em seguida, escolha Parameter Store.

3. Na página AWS Systems Manager Parameter Store, escolha Criar parâmetro.
4. Na página Criar parâmetro, na seção Detalhes de parâmetro, faça o seguinte:
   a. Para Nome, digite um parâmetro no formato **`/amplify/{your_app_id}/{your_backend_environment_name}/{your_parameter_name}`**.
   b. Em Type (Tipo), escolha SecureString.
   c. Para a fonte da chave KMS, escolha Minha conta atual para usar a chave padrão para sua conta.
   d. Em Valor, insira seu valor secreto para criptografar.
5. Escolha Criar parâmetro.

> **Note**
>
> O Amplify só tem acesso às chaves abaixo de `/amplify/{your_app_id}/{your_backend_environment_name}` para a compilação do ambiente específico. Você deve especificar o padrão AWS KMS key para permitir que o Amplify decifre o valor.

Acesso a segredos de ambiente para uma aplicação Gen 1

Os segredos do ambiente de uma aplicação Gen 1 são armazenados em `process.env.secrets` como uma string JSON.

Referência de segredos de ambiente do Amplify

Especifique um parâmetro do Systems Manager no formato `/amplify/{your_app_id}/{your_backend_environment_name}/AMPLIFY_SIWA_CLIENT_ID`.

É possível usar as seguintes segredos de ambiente que podem ser acessadas por padrão no Amplify.

Nome da variável	Description	Valor de exemplo
AMPLIFY_SIWA_CLIENT_ID	O login com o ID do cliente da Apple	`com.yourapp.auth`

Nome da variável	Description	Valor de exemplo
AMPLIFY_SIWA_TEAM_ID	O login com o ID da equipe da Apple	`ABCD123`
AMPLIFY_SIWA_KEY_ID	O login com o ID da chave da Apple	`ABCD123`
AMPLIFY_SIWA_PRIVATE_KEY	O login com a chave privada da Apple	-----BEGIN PRIVATE KEY----- ****..... -----END PRIVATE KEY-----

Configuração de cabeçalhos personalizados para uma aplicação do Amplify

Os cabeçalhos HTTP personalizados possibilitam que você especifique cabeçalhos para todas as respostas HTTP. Os cabeçalhos de resposta podem ser usados para fins de depuração, segurança e informativos. É possível especificar cabeçalhos no console do Amplify, ou baixando e editando o arquivo `customHttp.yml` de uma aplicação e salvando-o no diretório raiz do projeto. Para ver os procedimentos detalhados, consulte Configuração de cabeçalhos personalizados.

Anteriormente, cabeçalhos HTTP personalizados eram especificados para uma aplicação editando-se a especificação de compilação (buildspec) no consolde do Amplify ou baixando e atualizando o arquivo `amplify.yml` e salvando-o no diretório raiz do projeto. Recomendamos fortemente a migração dos cabeçalhos personalizados especificados dessa forma para fora do buildspec e do arquivo `amplify.yml`. Para instruções, consulte Migração de cabeçalhos personalizados para fora da especificação de compilação e amplify.yml.

Tópicos

- Referência de YAML de cabeçalho personalizado
- Configuração de cabeçalhos personalizados
- Migração de cabeçalhos personalizados para fora da especificação de compilação e amplify.yml
- Requisitos de cabeçalho personalizado monorepo

Referência de YAML de cabeçalho personalizado

Especifique cabeçalhos personalizados usando o seguinte formato YAML:

```
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'
```

Para um monorepo, use o seguinte formato YAML:

```
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'
```

Ao adicionar cabeçalhos personalizados ao seu aplicativo, você especificará seus próprios valores para o seguinte:

pattern

Os cabeçalhos personalizado são aplicados a todos os caminhos de arquivo de URL que correspondam ao padrão.

headers

Defina cabeçalhos que correspondem ao padrão de arquivo.

key

O nome do cabeçalho personalizado.

valor

O valor do cabeçalho personalizado.

Para saber mais sobre cabeçalhos HTTP, consulte a lista de cabeçalhos HTTP da Mozilla.

Configuração de cabeçalhos personalizados

Há duas formas de se especificar cabeçalhos HTTP personalizados para uma aplicação do Amplify. É possível especificar cabeçalhos no console do Amplify ou especificar cabeçalhos baixando e editando o arquivo `customHttp.yml` de uma aplicação e salvando-o no diretório raiz do seu projeto.

Para definir cabeçalhos personalizados para uma aplicação e salvá-los no console

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual definir cabeçalhos personalizados.
3. No painel de navegação, escolha Hospedagem e, em seguida, escolha Cabeçalhos personalizados.
4. Na página Cabeçalhos personalizados, escolha Editar.
5. Na janela Editar cabeçalhos personalizados, insira as informações dos seus cabeçalhos personalizados usando o formato YAML de cabeçalho personalizado.
   a. Para `pattern`, insira o padrão a ser correspondente.
   b. Para `key`, insira o nome do cabeçalho personalizado.
   c. Para `value`, insira o valor do cabeçalho personalizado.
6. Escolha Salvar.
7. Reimplante o aplicativo para aplicar os novos cabeçalhos personalizados.
   - Para um CI/CD aplicativo, navegue até a ramificação para implantar e escolha Reimplantar esta versão. É possível também realizar uma nova compilação a partir do seu repositório Git.
   - Para um aplicativo de implantação manual, implante o aplicativo novamente no console do Amplify.

Para definir cabeçalhos personalizados para uma aplicação e salvá-los na raiz do seu repositório

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual definir cabeçalhos personalizados.
3. No painel de navegação, escolha Hospedagem e, em seguida, escolha Cabeçalhos personalizados.
4. Na página Cabeçalhos personalizados, escolha Baixar YML.

5. Abra o arquivo `customHttp.yml` baixado no editor de código de sua preferência e insira as informações dos cabeçalhos personalizados usando o formato YAML do cabeçalho personalizado.

   a. Para `pattern`, insira o padrão a ser correspondente.

   b. Para `key`, insira o nome do cabeçalho personalizado.

   c. Para `value`, insira o valor do cabeçalho personalizado.

6. Salve o arquivo `customHttp.yml` editado no diretório raiz do seu projeto. Se você estiver trabalhando com um monorepo, salve o arquivo `customHttp.yml` na raiz do seu repositório.

7. Reimplante o aplicativo para aplicar os novos cabeçalhos personalizados.

   - Para um CI/CD aplicativo, execute uma nova compilação do seu repositório Git que inclua o novo arquivo. `customHttp.yml`
   - Para um aplicativo de implantação manual, implante o aplicativo novamente no console do Amplify e inclua o novo arquivo `customHttp.yml` com os artefatos que você carrega.

> **Note**
>
> Os cabeçalhos personalizados definidos no arquivo `customHttp.yml` e implantados no diretório raiz da aplicação substituirão os cabeçalhos personalizados definidos na seção Cabeçalhos personalizados no console do Amplify.

Exemplo de cabeçalhos personalizados de segurança

Cabeçalhos de segurança permitem aplicar HTTPS, impedindo ataques XSS e defendendo seu navegador contra clickjacking. Use a seguinte sintaxe YAML para aplicar cabeçalhos de segurança personalizados ao seu aplicativo.

```
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'"
```

Configuração de cabeçalhos personalizados de controle de cache

As aplicações hospedadas com o Amplify respeitam os cabeçalhos `Cache-Control` enviados pela origem, a menos que você os substitua por cabeçalhos personalizados definidos por você. O Amplify só aplica cabeçalhos personalizados de controle de cache para respostas com êxito com um código de status `200 OK`. Isso evita que as respostas de erro sejam armazenadas em cache e veiculadas a outros usuários que façam a mesma solicitação.

É possível ajustar manualmente a diretiva `s-maxage` para ter mais controle sobre o desempenho e a disponibilidade de implantação do seu aplicativo. Por exemplo, para aumentar o tempo de permanência do conteúdo em cache na borda, é possível aumentar manualmente o tempo de vida útil (TTL) atualizando `s-maxage` para um valor maior que o padrão de 600 segundos (10 minutos).

Para especificar um valor para `s-maxage`, use o seguinte formato YAML. Este exemplo mantém o conteúdo associado em cache na borda por 3600 segundo (uma hora).

```
customHeaders:
  - pattern: '/img/*'
    headers:
      - key: 'Cache-Control'
        value: 's-maxage=3600'
```

Para obter mais informações sobre como controlar o desempenho do aplicativo com cabeçalhos, consulte Uso do cabeçalho Cache-Control para aumentar a performance da aplicação.

Migração de cabeçalhos personalizados para fora da especificação de compilação e amplify.yml

Anteriormente, os cabeçalhos HTTP personalizados eram especificados para uma aplicação editando-se a especificação buildspec no console do Amplify ou baixando e atualizando o arquivo `amplify.yml` e salvando-o no diretório raiz do projeto. É altamente recomendável que você migre seus cabeçalhos personalizados da especificação de compilação e do arquivo `amplify.yml`.

Especifique seus cabeçalhos personalizados na seção Cabeçalhos personalizados do console do Amplify ou baixando e editando o arquivo `customHttp.yml`.

Para migrar cabeçalhos personalizados armazenados no console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo no qual realizar a migração personalizada do cabeçalho.
3. No painel de navegação, escolha Hospedagem, Configurações de compilação. Na seção Especificação de compilação do aplicativo, é possível revisar as especificações de construção do seu aplicativo.
4. Escolha Baixar para salvar uma cópia do seu buildspec atual. É possível consultar essa cópia posteriormente, se for preciso recuperar alguma configuração.
5. Quando o download for concluído, escolha Editar.
6. Anote as informações do cabeçalho personalizado no arquivo, pois você as usará posteriormente na etapa 9. Na janela Editar, exclua todos os cabeçalhos personalizados do arquivo e escolha Salvar.
7. No painel de navegação, escolha Hospedagem, Cabeçalhos personalizados.
8. Na página Cabeçalhos personalizados, escolha Editar.
9. Na janela Editar cabeçalhos personalizados, insira as informações dos cabeçalhos personalizados que você excluiu na etapa 6.
10. Escolha Salvar.
11. Reimplante qualquer ramificação à qual você deseja que os novos cabeçalhos personalizados sejam aplicados.

Para migrar cabeçalhos personalizados de amplify.yml para CustomHttp.yml

1. Navegue até o arquivo `amplify.yml` atualmente implantado no diretório raiz do seu aplicativo.
2. Abra o arquivo `amplify.yml` com seu editor de código preferido.
3. Anote as informações do cabeçalho personalizado no arquivo, pois você as usará posteriormente na etapa 8. Exclua os cabeçalhos personalizados no arquivo. Salve e feche o arquivo.
4. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
5. Escolha o aplicativo para o qual definir cabeçalhos personalizados.
6. No painel de navegação, escolha Hospedagem, Cabeçalhos personalizados.

7. Na página Cabeçalhos personalizados, escolha Baixar.
8. Abra o arquivo `customHttp.yml` baixado no editor de código de sua escolha e insira as informações dos cabeçalhos personalizados que você excluiu de `amplify.yml` na etapa 3.
9. Salve o arquivo `customHttp.yml` editado no diretório raiz do seu projeto. Se você estiver trabalhando com um monorepo, salve o arquivo na raiz do seu repositório.
10. Reimplante o aplicativo para aplicar os novos cabeçalhos personalizados.
    - Para um CI/CD aplicativo, execute uma nova compilação do seu repositório Git que inclua o novo arquivo. `customHttp.yml`
    - Para um aplicativo de implantação manual, implante o aplicativo novamente no console do Amplify e inclua o novo arquivo `customHttp.yml` com os artefatos que você carrega.

> **Note**
>
> Os cabeçalhos personalizados definidos no arquivo `customHttp.yml` e implantados no diretório raiz da aplicação substituirão os cabeçalhos personalizados definidos na seção Cabeçalhos personalizados no console do Amplify.

Requisitos de cabeçalho personalizado monorepo

Ao especificar cabeçalhos personalizados para um aplicativo em um monorepo, esteja ciente dos seguintes requisitos de configuração:

- Há um formato YAML específico para um monorepo. Para obter a sintaxe correta, consulte Referência de YAML de cabeçalho personalizado.
- É possível especificar cabeçalhos personalizados para uma aplicação em um monorepo usando a seção Cabeçalhos personalizados no console do Amplify. Você deve reimplantar sua aplicação para aplicar os novos cabeçalhos personalizados.
- Como alternativa ao uso do console, é possível especificar cabeçalhos personalizados para um aplicativo em um monorepo em um arquivo `customHttp.yml`. Você deve salvar o arquivo `customHttp.yml` na raiz do seu repositório e, em seguida, reimplantar o aplicativo para aplicar os novos cabeçalhos personalizados. Os cabeçalhos personalizados especificados no arquivo `customHttp.yml` substituem quaisquer cabeçalhos personalizados especificados usando a seção Cabeçalhos personalizados do console do Amplify.

Gerenciar a configuração de cache de uma aplicação

O Amplify usa CloudFront a Amazon para gerenciar a configuração de cache de seus aplicativos hospedados. Uma configuração de cache é aplicada a cada aplicação para otimizar a performance.

Em 13 de agosto de 2024, o Amplify lançou aprimoramentos na eficiência do cache para aplicações. Para obter mais informações, consulte Melhorias no cache da CDN para melhorar o desempenho do aplicativo com AWS Amplify hospedagem.

A tabela a seguir resume o suporte do Amplify para comportamentos específicos de cache antes e depois do lançamento dos aprimoramentos de cache.

Comportamento de armazenamento em cache	Suporte anterior	Com aprimoramentos no armazenamento em cache
É possível adicionar cabeçalhos personalizados para uma aplicação no console do Amplify ou em um arquivo `customHeaders.yaml`. Um dos cabeçalhos que podem ser substituídos é o `Cache-Control`. Para obter mais informações, consulte Configuração de cabeçalhos personalizados para uma aplicação do Amplify.	Sim	Sim
O Amplify respeita os cabeçalhos `Cache-Control` que você define em um arquivo `customHeaders.yaml`, e eles têm precedência sobre as configurações de cache padrão do Amplify.	Sim	Sim

Comportamento de armazenamento em cache	Suporte anterior	Com aprimoramentos no armazenamento em cache
O Amplify respeita os cabeçalhos `Cache-Control` definidos na estrutura de uma aplicação para rotas dinâmicas (por exemplo, rotas SSR do Next.js). Se um cabeçalho `Cache-Control` for definido no arquivo `customHeaders.yaml` da aplicação, isso terá precedência sobre as configurações no arquivo `next.config.js` .	Sim	Sim
Cada nova implantação de CI/CD aplicativo limpa o cache.	Sim	Sim
É possível ativar o modo de performance para uma aplicação.	Sim	Não A configuração do modo de performance não está mais disponível no console do Amplify. Contudo, é possível criar um cabeçalho `Cache-Control` que defina a diretiva `s-maxage`. Para instruções, consulte Uso do cabeçalho Cache-Control para aumentar a performance da aplicação.

A tabela a seguir lista as alterações nos valores padrão de configurações de cache específicas.

Configuração de cache	Valor padrão anterior	Valor padrão com aprimoramentos no armazenamento em cache
Duração do cache para ativos estáticos	Dois segundos	Um ano
Duração do cache para respostas de proxy reverso	Dois segundos	Zero segundos (sem armazenamento em cache)
Vida útil máxima (TTL)	Dez minutos	Um ano

Para obter mais informações sobre como o Amplify determina a configuração de armazenamento em cache a ser aplicada a uma aplicação e instruções sobre como gerenciar a configuração da chave de cache, consulte os tópicos a seguir.

Tópicos

- Como o Amplify aplica a configuração de cache a uma aplicação
- Gerenciamento de cookies de chave de cache
- Uso do cabeçalho Cache-Control para aumentar a performance da aplicação

Como o Amplify aplica a configuração de cache a uma aplicação

Para gerenciar o armazenamento em cache da sua aplicação, o Amplify determina o tipo de conteúdo que está sendo veiculado examinando o tipo de plataforma da aplicação e as regras de reescrita. Para as aplicações do `Compute`, o Amplify também examina as regras de roteamento no manifesto de implantação.

> **Note**
>
> O tipo de plataforma da aplicação é definido pela Amplify Hosting durante a implantação. Uma aplicação SSG (estática) é definida para o tipo de plataforma `WEB`. Uma aplicação SSR (Next.js 12 ou posterior) é definida para o tipo de plataforma `WEB_COMPUTE`.

O Amplify identifica os quatro tipos de conteúdo a seguir e aplica a política de cache gerenciado especificada.

Estático

O conteúdo veiculado por aplicações com a plataforma `WEB`, ou as rotas estáticas em uma aplicação `WEB_COMPUTE`.

Esse conteúdo usa a política de cache Amplify-StaticContent.

Otimização de imagem

As imagens veiculadas pelas rotas `ImageOptimization` em uma aplicação `WEB_COMPUTE`.

Esse conteúdo usa a política de cache Amplify-ImageOptimization.

Computação

O conteúdo veiculado pelas rotas `Compute` em uma aplicação `WEB_COMPUTE`. Isso inclui todo conteúdo renderizado do lado do servidor (SSR).

Esse conteúdo usa a política de cache Amplify-Default ou Amplify-DefaultNoCookies, dependendo do valor de `cacheConfig.type` definido em sua `App` do Amplify.

Proxy reverso

O conteúdo veiculado por caminhos que correspondem a uma regra personalizada de reescrita por proxy reverso. Para obter mais informações sobre a criação dessa regra personalizada, consulte Regravação de proxy reverso no capítulo Uso de redirecionamentos.

Esse conteúdo usa a política de cache Amplify-Default ou Amplify-DefaultNoCookies, dependendo do valor de `cacheConfig.type` definido em sua `App` do Amplify.

Noções básicas sobre as políticas de cache gerenciado do Amplify

O Amplify usa as políticas de cache gerenciado predefinidas a seguir para otimizar a configuração de cache padrão para suas aplicações hospedadas.

- Amplify-Default
- Amplify-DefaultNoCookies
- Amplify-ImageOptimization

- Amplify-StaticContent

Configurações de política de cache gerenciado padrão do Amplify

Exibir essa política no CloudFront console

Esta política foi projetada para uso com uma origem que é uma aplicação Web do AWS Amplify.

Essa política tem as seguintes configurações:

- TTL mínimo: 0 segundo
- TTL máximo: 31.536.000 segundos (um ano)
- TTL padrão: 0 segundo
- Cabeçalhos incluídos na chave de cache:
  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`
  - `Host`
- Cookies included in cache key (Cookies incluídos na chave de cache): todos os cookies serão incluídos.
- Query strings included in cache key (Strings de consulta incluídas na chave de cache): todas as strings de consulta serão incluídas.
- Configuração de objetos compactados em cache: habilitado para Gzip e Brotli.

Amplify - configurações de política de cache DefaultNoCookies gerenciado

Exibir essa política no CloudFront console

Esta política foi projetada para uso com uma origem que é uma aplicação Web do AWS Amplify.

Essa política tem as seguintes configurações:

- TTL mínimo: 0 segundo
- TTL máximo: 31.536.000 segundos (um ano)
- TTL padrão: 0 segundo
- Cabeçalhos incluídos na chave de cache:

  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`
  - `Host`
- Cookies incluídos na chave de cache: nenhum cookie será incluído.
- Query strings included in cache key (Strings de consulta incluídas na chave de cache): todas as strings de consulta serão incluídas.
- Configuração de objetos compactados em cache: habilitado para Gzip e Brotli.

Amplify - configurações de política de cache ImageOptimization gerenciado

Exibir essa política no CloudFront console

Esta política foi projetada para uso com uma origem que é uma aplicação Web do AWS Amplify.

Essa política tem as seguintes configurações:

- TTL mínimo: 0 segundo
- TTL máximo: 31.536.000 segundos (um ano)
- TTL padrão: 0 segundo
- Cabeçalhos incluídos na chave de cache:
  - `Authorization`
  - `Accept`
  - `Host`
- Cookies incluídos na chave de cache: nenhum cookie será incluído.
- Query strings included in cache key (Strings de consulta incluídas na chave de cache): todas as strings de consulta serão incluídas.
- Configuração de objetos compactados em cache: habilitado para Gzip e Brotli.

Amplify - configurações de política de cache StaticContent gerenciado

Exibir essa política no CloudFront console

Esta política foi projetada para uso com uma origem que é uma aplicação Web do AWS Amplify.

Essa política tem as seguintes configurações:

- TTL mínimo: 0 segundo
- TTL máximo: 31.536.000 segundos (um ano)
- TTL padrão: 0 segundo
- Cabeçalhos incluídos na chave de cache:
  - `Authorization`
  - `Host`
- Cookies incluídos na chave de cache: nenhum cookie será incluído.
- Strings de consulta incluídas na chave de cache: nenhuma string de consulta será incluída.
- Configuração de objetos compactados em cache: habilitado para Gzip e Brotli.

Gerenciamento de cookies de chave de cache

Ao implantar sua aplicação no Amplify, é possível escolher se deseja incluir ou excluir cookies na chave de cache. No console do Amplify, essa configuração é especificada na página Cabeçalhos personalizados e cache usando o botão Configurações da chave de cache. Para instruções, consulte Inclusão ou exclusão de cookies da chave de cache.

Incluir cookies na chave de cache

Com essa configuração, o Amplify escolhe automaticamente uma configuração de cache ideal para sua aplicação com base no tipo de conteúdo que está sendo veiculado. É necessário escolher explicitamente esse tipo de configuração de cache.

Se você estiver usando o SDKs ou o AWS CLI, essa configuração corresponde `cacheConfig.type` à configuração `AMPLIFY_MANAGED` com o `CreateApp` ou `UpdateApp` APIs.

Excluir cookies da chave de cache

Essa é a configuração de cache padrão. Essa configuração de cache é semelhante à configuração `AMPLIFY_MANAGED`, exceto pelo fato de excluir todos os cookies da chave de cache.

Optar por excluir cookies da chave de cache pode resultar em melhor performance do cache. Entretanto, antes de escolher essa configuração de cache, é importante considerar se a sua aplicação usa cookies para veicular conteúdo dinâmico.

Se você estiver usando o SDKs ou o AWS CLI, essa configuração corresponde à configuração do `cacheConfig.type` para `AMPLIFY_MANAGED_NO_COOKIES` com o `CreateApp` ou `UpdateApp` APIs.

Para obter mais informações sobre a chave de cache, consulte Entenda a chave de cache no Amazon CloudFront Developer Guide;.

Inclusão ou exclusão de cookies da chave de cache

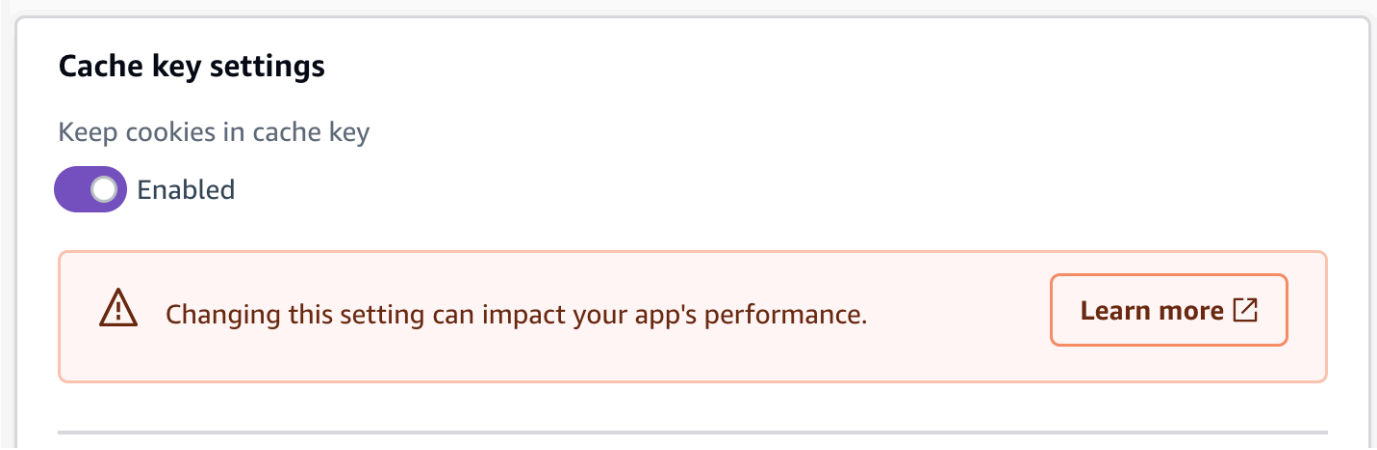
Você pode definir a configuração do cookie da chave de cache para um aplicativo no console do Amplify ou no. SDKs AWS CLI

Use o procedimento a seguir para especificar se deseja incluir ou excluir cookies da chave de cache ao implantar uma nova aplicação usando o console do Amplify.

Para definir a configuração do cookie da chave de cache ao implantar uma aplicação no Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todas as aplicações, escolha Criar nova aplicação.
3. Na página Comece a desenvolver com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
4. Na página Adicionar ramificação do repositório, faça o seguinte:
   a. Selecione o nome do repositório a ser conectado.
   b. Selecione o nome da ramificação do repositório a ser conectada.
   c. Escolha Próximo.
5. Se a aplicação exigir um perfil de serviço do IAM, é possível permitir que a computação do Amplify Hosting crie automaticamente um perfil de serviço, ou é possível especificar um perfil que tenha criado.
   - Para permitir que o Amplify crie automaticamente um perfil e o anexe à sua aplicação:
     - Escolha Criar e usar um novo perfil de serviço.
   - Para anexar um perfil de serviço que você criou anteriormente:
     a. Selecione Usar um perfil de serviço existente.
     b. Selecione o perfil a ser usado na lista.
6. Escolha Configurações avançadas e localize a seção Configurações da chave de cache.

7. Escolha Manter cookies na chave de cache ou Remover cookies de chave de cache. A captura de tela a seguir mostra o botão Configurações da chave de cache no console.



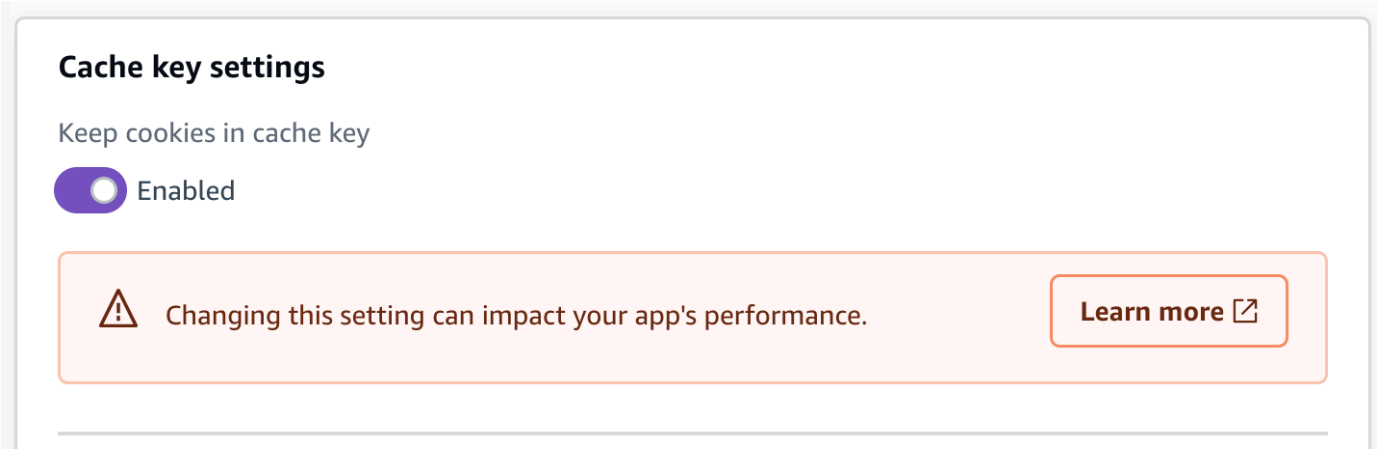
8. Escolha Próximo.
9. Na página Revisar, escolha Salvar e implantar.

Alteração da configuração do cookie de chave de cache para uma aplicação

É possível alterar a configuração de cookie de chave de cache para uma aplicação que já esteja implantada no Amplify. Use o procedimento a seguir para alterar se deseja incluir ou excluir cookies de chave de cache para uma aplicação usando o console do Amplify.

Para alterar a configuração de cookie de chave de cache para uma aplicação implantada

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todas as aplicações, escolha a aplicação que deseja atualizar.
3. No painel de navegação, escolha Hospedagem e, em seguida, escolha Cabeçalhos personalizados e cache.
4. Na página Cabeçalhos personalizados e cache, localize a seção Configurações de chave de cache e escolha Editar.
5. Escolha Manter cookies na chave de cache ou Remover cookies de chave de cache. A captura de tela a seguir mostra o botão Configurações da chave de cache no console.



6. Escolha Salvar.

Uso do cabeçalho Cache-Control para aumentar a performance da aplicação

A arquitetura de hospedagem padrão do Amplify otimiza o equilíbrio entre desempenho de hospedagem e disponibilidade de implantação. Para a maioria dos clientes, recomendamos usar a arquitetura padrão.

Se você precisar de um controle mais preciso sobre a performance de uma aplicação, é possível definir manualmente o cabeçalho `Cache-Control` de HTTP para otimizar a performance da hospedagem, mantendo o conteúdo armazenado em cache na borda da rede de entrega de conteúdo (CDN) por um intervalo mais longo.

As diretivas `max-age` e `s-maxage` do cabeçalho HTTP `Cache-Control` afetam a duração do armazenamento em cache do conteúdo da sua aplicação. A diretiva `max-age` informa o navegador de permanência (em segundos) de um conteúdo no cache antes de o obtê-lo do servidor de origem. A diretiva `s-maxage` substitui a `max-age` e permite especificar o tempo de permanência (em segundos) de um conteúdo na borda da CDN antes de o obtê-lo do servidor de origem antes de o obtê-lo do servidor de origem.

As aplicações hospedadas com o Amplify respeitam os cabeçalhos `Cache-Control` enviados pela origem, a menos que você os substitua por cabeçalhos personalizados definidos por você. O Amplify só aplica cabeçalhos `Cache-Control` personalizados para respostas com êxito com um código de status `200 OK`. Isso evita que as respostas de erro sejam armazenadas em cache e veiculadas a outros usuários que façam a mesma solicitação.

É possível ajustar manualmente a diretiva `s-maxage` para ter mais controle sobre o desempenho e a disponibilidade de implantação do seu aplicativo. Por exemplo, para alterar o intervalo de tempo que seu conteúdo permanece armazenado em cache na borda, é possível definir manualmente o tempo de vida útil (TTL) atualizando `s-maxage` para um valor diferente do padrão de 31536000 segundos (1 ano).

É possível definir cabeçalhos personalizados para um aplicativo na seção Cabeçalhos personalizados do console do Amplify. Para ver um exemplo de uso do formato YAML, consulte Configuração de cabeçalhos personalizados de controle de cache.

Use o procedimento a seguir para definir a diretiva `s-maxage` para manter o conteúdo armazenado em cache na borda da CDN por 24 horas.

Para definir um cabeçalho Cache-Control personalizado

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual definir cabeçalhos personalizados.
3. No painel de navegação, escolha Hospedagem, Cabeçalhos personalizados.
4. Na página Cabeçalhos personalizados, escolha Editar.
5. Na janela Editar cabeçalhos personalizados, insira as informações para seu cabeçalho personalizado conforme a seguir:
   a. Em `pattern`, insira **`**/*`** para todos os caminhos.
   b. Em `key`, digite **`Cache-Control`**.
   c. Em `value`, digite **`s-maxage=86400`**.
6. Escolha Salvar.
7. Reimplante a aplicação para aplicar o novo cabeçalho personalizado.

Proteção contra distorções para implantações do Amplify

A proteção contra distorção de implantação está disponível para as aplicações do Amplify para eliminar problemas de distorção de versão entre clientes e servidores em aplicações da Web. Ao aplicar proteção de distorção a uma aplicação do Amplify, é possível garantir que seus clientes sempre interajam com a versão correta dos ativos do lado do servidor, independentemente de quando a implantação ocorrer.

A distorção de versão é um desafio comum para desenvolvedores da Web. Ela ocorre quando um navegador da Web está executando uma versão desatualizada de uma aplicação e o servidor está executando uma nova. Essa discrepância pode causar comportamento imprevisível, erros e uma experiência degradada para o usuário da aplicação. O recurso de proteção de distorção de implantação do Amplify vincula os clientes em execução em navegadores da web a uma implantação específica. Isso garante que o Amplify sempre forneça os ativos dessa implantação específica, mantendo o cliente e o servidor sincronizados.

O recurso de proteção contra distorções do Amplify pode reduzir os erros dos usuários da sua aplicação à medida que você lança novas implantações. Também pode melhorar a experiência do desenvolvedor ao reduzir o tempo gasto no gerenciamento de problemas de compatibilidade com versões anteriores e futuras.

Detalhes do recurso de proteção contra distorções:

Tipos de aplicação compatíveis

- É possível adicionar a proteção contra distorção a aplicações estáticas e de SSR criadas com qualquer estrutura com suporte pelo Amplify. As aplicações podem ser implantadas a partir de um repositório Git ou de uma implantação manual.

- Você não pode adicionar proteção contra distorções a uma aplicação implantado na plataforma `WEB_DYNAMIC` (Next.js versão 11 ou anterior).

Duração

- Para aplicações estáticas, o Amplify oferece uma semana de implantações. Para aplicações de SSR, garantimos proteção contra distorções para até oito implantações anteriores.

Custo

- Não há custo adicional para adicionar a proteção contra distorções a uma aplicação.

Considerações sobre a performance

Quando a proteção de distorção está ativada para uma aplicação, o Amplify deve atualizar suas configurações de cache de CDN. Portanto, é necessário esperar que sua primeira implantação depois de ativar a proteção contra distorções leve até dez minutos.

Tópicos

Configuração da proteção contra distorções de implantação para uma aplicação do Amplify

Você pode adicionar ou remover a proteção contra distorção de implantação de um aplicativo usando o console do Amplify, AWS Command Line Interface o ou o. SDKs O recurso é aplicado no nível da ramificação. Somente novas implantações, feitas após a ativação da proteção contra distorção em uma ramificação, serão protegidas contra distorções.

Para adicionar ou remover a proteção contra distorção de implantação usando os campos AWS CLI ou SDKs, use os `UpdateBranch.enableSkewProtection` campos `CreateBranch.enableSkewProtection` e. Para obter mais informações, consulte CreateBranche UpdateBranchna documentação de referência da API Amplify.

Se você quiser remover uma implantação específica para que ela não seja mais atendida, use a API `DeleteJob`. Para obter mais informações, consulte a DeleteJobdocumentação de referência da API Amplify.

No momento, você só pode ativar a proteção contra distorções em uma aplicação que já esteja implantada no Amplify Hosting. Use as instruções a seguir para adicionar a proteção contra distorções a uma ramificação usando o console do Amplify.

Ative a proteção contra distorções para uma ramificação de uma aplicação do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/
2. Na página Todas as aplicações, escolha o nome da aplicação implantada para habilitar a proteção contra distorções.

3. No painel de navegação, escolha Configurações da aplicação, e, em seguida, escolha Configurações da ramificação.
4. Na seção Ramificações, escolha o nome da ramificação a ser atualizada.
5. No menu Ações, escolha Habilitar proteção contra distorções.
6. Na janela de confirmação, escolha Confirmar. A proteção contra distorções agora está habilitada para a ramificação.
7. Reimplante sua ramificação da aplicação. Somente implantações feitas após a ativação da proteção contra distorções serão protegidas contra distorções.

Use as instruções a seguir para remover a proteção contra distorções de uma ramificação de uma aplicação usando o console do Amplify.

Remova a proteção contra distorções de uma ramificação de uma aplicação do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/
2. Na página Todas as aplicações, escolha o nome da aplicação implantada para remover a proteção contra distorções.
3. No painel de navegação, escolha Configurações da aplicação, e, em seguida, escolha Configurações da ramificação.
4. Na seção Ramificações, escolha o nome da ramificação a ser atualizada.
5. No menu Ações, escolha Desabilitar proteção contra distorções. A proteção contra distorções agora está desabilitada para a ramificação e somente o conteúdo mais recente será veiculado.

Como funciona a proteção contra distorções

Na maioria dos casos, o comportamento padrão do cookie _dpl atenderá às suas necessidades de proteção contra distorções. No entanto, nos cenários avançados a seguir, a proteção contra distorções será melhor ativada usando o cabeçalho `X-Amplify-Dpl` e o parâmetro de consulta `dpl`.

- Carregamento do seu site em várias guias do navegador ao mesmo tempo
- Uso de trabalhadores de serviço

O Amplify avalia a solicitação recebida na ordem a seguir ao determinar o conteúdo a ser veiculado ao cliente:

1. Cabeçalho **`X-Amplify-Dpl`**: as aplicações podem usar esse cabeçalho para direcionar solicitações para uma implantação específica do Amplify. Esse cabeçalho de solicitação pode ser definido usando o valor de `process.env.AWS_AMPLIFY_DEPLOYMENT_ID`.
2. Parâmetro de consulta **`dpl`**: as aplicações Next.js definirão automaticamente o parâmetro de consulta _dpl para solicitações de ativos com impressão digital (arquivos.js e .css).
3. Cookie _dpl: esse é o padrão para todas as aplicações protegidas contra distorções. Para um navegador específico, o mesmo cookie é enviado para cada guia ou instância do navegador que interage com um domínio.

   Lembre-se de que, se diferentes guias do navegador tiverem versões diferentes de um site carregadas, o cookie _dpl será compartilhado por todas as guias. Nesse cenário, não é possível obter proteção total contra distorções com o cookie _dpl e é necessário considerar o uso do cabeçalho `X-Amplify-Dpl` para proteção contra distorções.

X-Amplify-Dpl exemplo de cabeçalho

O exemplo a seguir demonstra o código de uma página de SSR do Next.js que acessa a proteção contra distorções por meio do cabeçalho `X-Amplify-Dpl`. A página renderiza seu conteúdo com base em uma de suas rotas de API. A implantação para servir à rota da API é especificada usando o cabeçalho `X-Amplify-Dpl`, que é definido com o valor `process.env.AWS_AMPLIFY_DEPLOYMENT_ID`.

```
import { useEffect, useState } from 'react';

export default function MyPage({deploymentId}) {
    const [data, setData] = useState(null);

    useEffect(() => {
        fetch('/api/hello', {
            headers: {
                'X-Amplify-Dpl': process.env.AWS_AMPLIFY_DEPLOYMENT_ID
            },
        })
        .then(res => res.json())
        .then(data => setData(data))
        .catch(error => console.error("error", error))
```

```
    }, []);

    return <div>
        {data ? JSON.stringify(data) : "Loading ... " }
    </div>
}
```

Monitoramento de uma aplicação do Amplify

AWS Amplify fornece os seguintes recursos para monitorar seus aplicativos hospedados:

- CloudWatch métricas — O Amplify emite métricas por meio da Amazon CloudWatch que você pode usar para monitorar tráfego, erros, transferência de dados e latência de seus aplicativos.
- Logs de acesso: o Amplify fornece logs de acesso com informações detalhadas sobre solicitações feitas à sua aplicação.
- CloudTrail registro — O Amplify é integrado e AWS CloudTrail fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amplify. Você pode ver esses eventos no CloudTrail console.

Tópicos

Monitorando um aplicativo Amplify com a Amazon CloudWatch

AWS Amplify é integrado à Amazon CloudWatch, permitindo que você monitore métricas para seus aplicativos Amplify quase em tempo real e crie alarmes que enviam notificações quando uma métrica excede um limite definido por você. Para obter mais informações sobre como o CloudWatch serviço funciona, consulte o Guia CloudWatch do usuário da Amazon.

CloudWatch Métricas suportadas

O Amplify suporta sete CloudWatch métricas no `AWS/AmplifyHosting` namespace para monitorar tráfego, erros, transferência de dados, latência e tokens de solicitação para seus aplicativos. Essas métricas são agregadas em intervalos de um minuto. CloudWatch as métricas de monitoramento são gratuitas e não contam para as cotas CloudWatch de serviço.

A tabela a seguir descreve cada métrica com suporte e lista as estatísticas mais relevantes. Nem todas as estatísticas são aplicáveis a todas as métricas.

Métrica	Description
Solicitações	O número total de solicitações de visualizadores recebidas pelo seu aplicativo. A estatística mais relevante é `Sum`. Use a estatística `Sum` para obter o número total de solicitações.
BytesDownloaded	A quantidade total de dados transferidos do seu aplicativo (baixados) em bytes pelos visualizadores para as solicitações `GET`, `HEAD` e `OPTIONS`. A estatística mais relevante é `Sum`.
BytesUploaded	A quantidade total de dados transferidos para a sua aplicação (enviados) em bytes para qualquer solicitação, incluindo os cabeçalhos. O Amplify não cobra pelos dados enviados em suas aplicações. A estatística mais relevante é `Sum`.
4xxErrors	O número de solicitações que retornaram um erro no intervalo do código de status HTTP 400-499. A estatística mais relevante é `Sum`. Use a estatística `Sum` para obter o total de ocorrências desses erros.
5xxErrors	O número de solicitações que retornaram um erro no intervalo do código de status HTTP 500-599.

Métrica	Description
	A estatística mais relevante é `Sum`. Use a estatística `Sum` para obter o total de ocorrênci as desses erros.
Latência	O tempo até o primeiro byte em segundos. Este é o tempo total entre o momento em que o Amplify Hosting recebe uma solicitação e o momento em que retorna uma resposta à rede. Isso não inclui a latência da rede encontrada para que uma resposta alcance o dispositivo do visualizador. As estatísticas mais relevantes são `Average`, `Maximum`, `Minimum`, `p10`, `p50`, `p90`, `p95` e `p100`. Use a estatística `Average` para avaliar as latências esperadas.

Métrica	Description
TokensConsumed	Os tokens de solicitação consumidos pela sua aplicação. A estatística `Sum` representa o consumo total de tokens de solicitação. É possível comparar essa estatística com sua cota de serviço `Request tokens per second` atual para determinar se é necessário solicitar um aumento de cota para evitar possíveis limitações durante um futuro evento de alto tráfego. A estatística `Average` representa o consumo de tokens de solicitação nos horários normais e de pico. O maior consumo de tokens normalmente leva a um tempo maior até o primeiro byte (TTFB). Portanto, é possível usar essa estatística ao avaliar a latência da sua aplicação. Se sua latência for baixa, você pode melhorar seu downstream APIs para reduzir o consumo de tokens e evitar a limitação que pode ocorrer quando o consumo de tokens excede a cota de serviço do seu aplicativo. `Request tokens per second` Para obter mais informações sobre as cotas de serviço `Request tokens per second`, consulte Service quotas do Amplify Hosting.

O Amplify fornece as seguintes dimensões CloudWatch métricas.

Dimensão	Description
App	Os dados métricos são fornecidos pelo aplicativo.

Dimensão	Description
Conta da AWS	Os dados métricos são fornecidos em todos os aplicativos no Conta da AWS.

Acessando CloudWatch métricas

Você pode acessar CloudWatch as métricas diretamente do console do Amplify usando o procedimento a seguir.

> **Note**
>
> Você também pode acessar CloudWatch as métricas Console de gerenciamento da AWS em https://console.aws.amazon.com/cloudwatch/.

Para acessar métricas usando o console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o serviço do qual você deseja visualizar as métricas.
3. No painel de navegação, selecione Monitoramento e, em seguida, escolha Métricas.

Criação de CloudWatch alarmes

Você pode criar CloudWatch alarmes no console do Amplify que enviam notificações quando critérios específicos são atendidos. Um alarme monitora uma única CloudWatch métrica e envia uma notificação do Amazon Simple Notification Service quando a métrica ultrapassa o limite de um número específico de períodos de avaliação.

Você pode criar alarmes mais avançados que usam expressões matemáticas métricas no CloudWatch console ou usando o. CloudWatch APIs Por exemplo, é possível criar um alarme que envie uma notificação quando o percentual 4xxErrors ultrapassar 15% por três períodos consecutivos. Para obter mais informações, consulte Criação de um CloudWatch alarme com base em uma expressão matemática métrica no Guia CloudWatch do usuário da Amazon.

O CloudWatch preço padrão se aplica aos alarmes. Para obter mais informações, consulte os CloudWatchpreços da Amazon.

Use o procedimento a seguir para criar um alarme no console do Amplify.

Para criar um CloudWatch alarme para uma métrica do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo no qual deseja definir um alarme.
3. No painel de navegação, selecione Monitoramento e, em seguida, escolha Alarmes.
4. Na página Alarmes, escolha Criar alarme.
5. Na janela Criar alarme, configure seu alarme da seguinte forma:

    a. Em Métrica, escolha o nome da métrica a ser monitorada na lista.

    b. Em Nome de alarme, digite um nome para o alarme. Por exemplo, se você estiver monitorando Solicitações, poderá nomear o alarme **`HighTraffic`**. O nome deve conter somente caracteres ASCII.

    c. Para Definir notificações, utilize um dos seguintes procedimentos:

    - i. Escolha Novo para configurar um novo tópico do Amazon SNS.
        ii. Em Endereço de e-mail, insira o endereço de e-mail do destinatário das notificações.
        iii. Escolha Adicionar novo endereço de e-mail para adicionar mais destinatários.
    - i. Escolha Existente para reutilizar um tópico do Amazon SNS.
        ii. Para Tópico existente do SNS, selecione o nome do tópico Amazon SNS na lista.

    d. Para Sempre que a Estatística da Métrica, defina as condições para seu alarme da seguinte forma:

        i. Especifique se a métrica deve ser maior que, menor que ou igual ao valor limite.
        ii. Especifique o valor do limite.
        iii. Especifique o número de períodos de avaliação consecutiva que devem estar no estado do alarme para acionar o alarme.
        iv. Especifique a duração do período de avaliação.

    e. Escolha Confirmar.

> **Note**
>
> Cada destinatário do Amazon SNS que você especificar recebe um e-mail de confirmação das Notificações AWS . O e-mail contém um link que o destinatário deve seguir para confirmar sua assinatura e receber notificações.

Acessando CloudWatch registros para aplicativos SSR

Amplify envia informações sobre seu tempo de execução de SSR para o Amazon CloudWatch Logs em seu. Conta da AWS Ao implantar uma aplicação de SSR na computação do Amplify Hosting, a aplicação requer um perfil de serviço do IAM que o Amplify assume ao chamar outros serviços em seu nome. É possível permitir que a computação do Amplify Hosting crie automaticamente um perfil de serviço ou especificar um perfil que criou.

Se você optar por permitir que o Amplify crie uma função do IAM para você, a função já terá as permissões para criar CloudWatch registros. Se você criar sua própria função do IAM, precisará adicionar as seguintes permissões à sua política para permitir que o Amplify acesse o Amazon CloudWatch Logs.

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

Para obter mais informações sobre a adição de um perfil de serviço, consulte Adição de um perfil de serviço com permissões para implantar recursos de backend. Para mais informações sobre como implantar aplicativos renderizados no lado do servidor, consulte Implantação de aplicações renderizadas do lado do servidor com o Amplify Hosting.

Você pode visualizar os registros computacionais do Amplify Hosting para um aplicativo SSR no console ou no CloudWatch console do Amplify. Use as instruções a seguir para visualizar os logs no console do Amplify.

Para visualizar CloudWatch os registros de um aplicativo SSR no console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo SSR para ver os CloudWatch registros.

3. No painel de navegação, escolha Monitoramento e, em seguida, escolha Logs de computação do Hosting.
4. Na página Hosting compute logs, pesquise e selecione um grupo de CloudWatch registros para uma ramificação específica.

Recuperação de análise de logs de acesso de uma aplicação do Amplify

O Amplify armazena logs de acesso de todos os aplicativos que você hospeda no Amplify. Os loga de acesso contêm informações sobre todas as solicitações feitas aos seus aplicativos hospedados. O Amplify retém todos os logs de acesso de uma aplicação até que você exclua a aplicação. Todos os logs de acesso de uma aplicação estão disponíveis no console do Amplify. Contudo, cada solicitação individual por logs de acesso é limitada a um período especificado por você de duas semanas.

> **Warning**
>
> Não inclua segredos, credenciais ou dados confidenciais URLs como caminho ou parâmetros de consulta. Esses valores podem ser visualizados em texto simples nos logs de acesso da sua aplicação do Amplify.

O Amplify nunca reutiliza CloudFront distribuições entre clientes. O Amplify cria CloudFront distribuições com antecedência para que você não precise esperar pela criação de uma CloudFront distribuição ao implantar um novo aplicativo. Antes que essas distribuições sejam atribuídas a um aplicativo Amplify, elas podem receber tráfego de bots. No entanto, eles estão configurados para sempre responder como Não encontrados antes de serem atribuídos. Se os logs de acesso do seu aplicativo contiverem entradas de um período antes de você criar seu aplicativo, essas entradas estão relacionadas a essa atividade.

> **Important**
>
> Recomendamos que você use os logs para compreender a natureza das solicitações do seu conteúdo, não como uma contabilidade completa de todas as solicitações. O Amplify entrega logs de acesso com base no melhor esforço. A entrada do log de uma solicitação específica pode ser entregue muito depois do processamento da solicitação e, raramente, nunca ser

entregue. Quando uma entrada de registro é omitida dos registros de acesso, o número de entradas nos registros de acesso não corresponde ao uso que aparece nos relatórios de AWS faturamento e uso.

Recuperação dos logs de acesso de uma aplicação

Siga o procedimento a seguir para recuperar os logs de acesso de uma aplicação do Amplify.

Para visualizar logs de acesso

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo para o qual deseja acessar os logs.
3. No painel de navegação, selecione Monitoramento e, em seguida, escolha Logs de acesso.
4. Escolha Editar intervalo de tempo.
5. Na janela Editar intervalo de tempo, faça o seguinte:
   a. Em Data de início, especifique o primeiro dia do intervalo de duas semanas para recuperar os logs.
   b. Em Hora de início, escolha a hora do primeiro dia para iniciar a recuperação do log.
   c. Escolha Confirmar.
6. O console do Amplify exibe os logs do intervalo de tempo especificado na seção Logs de acesso. Escolha Baixar para salvar os logs em formato CSV.

Como analisar os logs de acesso

Para analisar logs de acesso, é possível armazenar os arquivos CSV em um bucket do Amazon S3. Uma forma de analisar seus logs de acesso é usar o Athena. O Athena é um serviço de consulta interativo que pode ajudar você a analisar dados para AWS serviços. Você pode seguir as step-by-step instruções aqui para criar uma tabela. Quando sua tabela tiver sido criada, é possível consultar dados da seguinte maneira.

```
SELECT SUM(bytes) AS total_bytes
FROM logs
WHERE "date" BETWEEN DATE '2018-06-09' AND DATE '2018-06-11'
LIMIT 100;
```

Registrando chamadas da API Amplify usando AWS CloudTrail

AWS Amplify é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amplify. CloudTrail captura todas as chamadas de API para o Amplify como eventos. As chamadas capturadas incluem as chamadas do console do Amplify e as chamadas de código para as operações da API do Amplify. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Amplify. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações CloudTrail coletadas, você pode determinar a solicitação que foi feita ao Amplify, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

Amplifique as informações em CloudTrail

CloudTrail está ativado em sua AWS conta por padrão. Quando a atividade ocorre no Amplify, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. É possível visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de CloudTrail eventos com histórico de eventos no Guia AWS CloudTrail do usuário.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos para o Amplify, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando uma trilha é criada no console, a mesma é aplicada a todas as regiões da AWS . A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para mais informações, consulte o seguinte no Guia do usuário do AWS CloudTrail :

- Criando uma trilha para sua AWS conta
- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e Recebendo arquivos de CloudTrail log de várias contas

Todas as operações do Amplify são registradas CloudTrail e documentadas na Referência da API do AWS Amplify Console, na Referência da API AWSAmplify Admin UI e na Referência da API do Amplify UI Builder. Por exemplo, chamadas para as `DeleteBackendEnvironment` operações`CreateApp`, `DeleteApp` e geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- A solicitação foi feita com credenciais de usuário root ou AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Foi a solicitação feita por outro AWS serviço.

Para obter mais informações, consulte o elemento CloudTrail userIdentity no Guia do AWS CloudTrail usuário.

Noções básicas sobre as entradas de arquivos de log do Amplify

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ListApps`operação AWS Amplify Console API Reference.


```
{
    "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"
}
```

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a [ListBackendJobs](#)operação de referência da API AWS Amplify Admin UI.

```
{
    "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"
}
```

Uso de perfis do IAM com aplicações do Amplify

Um perfil do IAM é uma identidade do IAM com permissões específicas. As permissões do perfil determinam o que a identidade pode e não pode fazer na AWS. É possível criar perfis do IAM na sua Conta da AWS e usá-los para delegar permissões ao Amplify Hosting. Para saber mais sobre perfis do IAM, consulte Perfis do IAM no Guia do usuário do IAM.

É possível usar os tipos de perfis do IAM a seguir para conceder ao Amplify Hosting as permissões necessárias para realizar ações em seu nome ou executar código de computação que acesse outros recursos da AWS .

Perfil de serviço do IAM

O Amplify assume esse perfil para executar ações em seu nome. Esse perfil é necessário para aplicações com recursos de backend.

Perfil do SSR Compute do IAM

Permite que uma aplicação renderizada do lado do servidor (SSR) acesse com segurança recursos específicos da AWS .

Função IAM SSR CloudWatch Logs

Quando você implanta um aplicativo SSR, o aplicativo exige uma função de serviço do IAM que o Amplify assume para permitir que o Amplify acesse o Amazon Logs. CloudWatch

Tópicos

- Adição de um perfil de serviço com permissões para implantar recursos de backend
- Adicionar uma função SSR Compute para permitir o acesso aos recursos AWS
- Adicionar uma função de serviço com permissões para acessar o CloudWatch Logs

Adição de um perfil de serviço com permissões para implantar recursos de backend

O Amplify exige permissões para implantar recursos do backend com o frontend. Você usa um perfil de serviço para fazer isso. Uma função de serviço é a função AWS Identity and Access Management

(IAM) que fornece ao Amplify Hosting permissões para implantar, criar e gerenciar back-ends em seu nome.

Quando você cria uma nova aplicação que exige um perfil de serviço do IAM, é possível permitir que o Amplify Hosting crie automaticamente um perfil de serviço para você, ou é possível selecionar um perfil do IAM que já tenha criado. Nesta seção, você criará um perfil de serviço do Amplify que tenha permissões administrativas da conta e permita explicitamente acesso direto aos recursos que as aplicações do Amplify exigem para implantar, criar e gerenciar backends.

Para criar um perfil de serviço do Amplify no console do IAM

Para criar um perfil de serviço

1. Faça login no console do IAM e escolha Perfis na barra de navegação à esquerda e escolha Criar perfil.
2. Na página Selecionar tipo de entidade confiável, escolha Serviço da AWS . Em Caso de uso, selecione Amplify - Implantação de backend e, em seguida, Próximo.
3. Na página Adicionar permissões, escolha Próximo.
4. Na página Nomear, visualizar e criar, em Nome do perfil, insira um nome significativo, como **`AmplifyConsoleServiceRole-AmplifyRole`**.
5. Aceite todos os padrões e escolha Criar perfil.
6. Volte ao console do Amplify para anexar o perfil à sua aplicação.
   - Se você estiver no processo de implantação de uma nova aplicação, faça o seguinte:
     a. Atualize a lista de perfis de serviço.
     b. Escolha o perfil que você acabou de criar. Para este exemplo, deve ser semelhante a AmplifyConsoleServiceRole- AmplifyRole.
     c. Escolha Próximo e siga as etapas para concluir a implantação da sua aplicação.
   - Se você já tiver uma aplicação existente, faça o seguinte:
     a. No painel de navegação, escolha Configurações da aplicação, e, a seguir, escolha Perfis do IAM.
     b. Na página Perfis do IAM, na seção Perfil de serviço, escolha Editar.
     c. Na página Perfil de serviço, selecione o perfil que você acabou de criar na lista Perfil de serviço.
     d. Escolha Salvar.

7. Agora o Amplify tem permissões para implantar recursos do backend na sua aplicação.

Edição da política de confiança de um perfil de serviço para evitar o representante confuso

O problema do “confused deputy” é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executar a ação. Para obter mais informações, consulte Cross-service prevenção delegada confusa.

Atualmente, a política de confiança padrão para o `Amplify-Backend Deployment` perfil de serviço impõe `aws:SourceArn` e `aws:SourceAccount` chaves de condição de contexto global para prevenir o problema do “confused deputy”. No entanto, se você já criou um perfil `Amplify-Backend Deployment` em sua conta, pode atualizar a política de confiança do perfil para adicionar essas condições para se proteger contra o "confused deputy".

Use o exemplo a seguir para restringir o acesso aos aplicativos em sua conta. Substitua a região e o ID da aplicação no exemplo por suas próprias informações.

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
```

Para obter instruções sobre como editar a política de confiança de uma função usando o Console de gerenciamento da AWS, consulte Modificar uma função (console) no Guia do usuário do IAM.

Adicionar uma função SSR Compute para permitir o acesso aos recursos AWS

Essa integração permite atribuir um perfil do IAM ao serviço SSR Compute do Amplify para permitir que sua aplicação renderizada do lado do servidor (SSR) acesse com segurança recursos específicos da AWS com as permissões do perfil Por exemplo, você pode permitir que as funções de computação SSR do seu aplicativo acessem com segurança outros AWS serviços ou recursos, como

um bucket do Amazon Bedrock Amazon S3, com base nas permissões definidas na função do IAM atribuída.

O perfil do IAM SSR Compute fornece credenciais temporárias, eliminando a necessidade de codificar credenciais de segurança de longa duração em variáveis de ambiente. O uso da função IAM SSR Compute se alinha às melhores práticas de AWS segurança de conceder permissões de privilégio mínimo e usar credenciais de curto prazo quando possível.

As instruções mais adiante nesta seção descrevem como criar uma política com permissões personalizadas e anexar a política a um perfil. Ao criar o perfil, será necessário anexar uma política de confiança personalizada que dê permissão ao Amplify para assumir o perfil. Se a relação de confiança não estiver definida corretamente, você receberá um erro ao tentar adicionar o perfil. A política de confiança personalizada a seguir concede ao Amplify a permissão de assumir o perfil.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Você pode associar uma função do IAM em sua Conta da AWS a um aplicativo SSR existente usando o console AWS SDKs do Amplify ou o. AWS CLI A função que você anexa é associada automaticamente ao serviço de computação Amplify SSR, concedendo a ele as permissões que você especifica para acessar outros recursos. AWS Como as necessidades da sua aplicação mudam com o tempo, é possível modificar o perfil do IAM anexado sem reimplantar sua aplicação. Isso proporciona flexibilidade e reduz o tempo de inatividade da aplicação.

> **Important**
>
> Você é responsável por configurar a aplicação para atender aos seus objetivos de segurança e conformidade. Isso inclui gerenciar seu perfil SSR Compute, que deve ser configurado para ter o conjunto mínimo de permissões necessárias para dar suporte ao seu caso de uso. Para obter mais informações, consulte Gerenciamento da segurança do perfil SSR Compute do IAM.

Criação de um perfil SSR Compute no console do IAM

Antes de poder anexar um perfil do IAM SSR Compute a uma aplicação do Amplify, o perfil já deve existir na sua Conta da AWS. Nesta seção, você aprenderá como criar uma política do IAM e vinculá-la a um perfil que o Amplify pode assumir para acessar recursos específicos da AWS .

Recomendamos que você siga as AWS melhores práticas de conceder permissões de privilégio mínimo ao criar uma função do IAM. O perfil do IAM SSR Compute é chamado somente a partir das funções de computação de SSR e, portanto, só deve conceder as permissões necessárias para a execução do código.

Você pode usar o Console de gerenciamento da AWS AWS CLI, ou SDKs para criar políticas no IAM. Para obter mais informações, consulte Definição de permissões personalizadas do IAM com políticas gerenciadas pelo cliente no Guia do usuário do IAM.

As instruções a seguir demonstram como usar o console do IAM para criar uma política do IAM que define as permissões a serem concedidas ao serviço Amplify Compute.

Para usar o editor de políticas JSON do console do IAM para criar uma política

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Políticas.
3. Selecione Criar política.
4. Na seção Editor de políticas, escolha a opção JSON.
5. Digite ou cole um documento de política JSON.
6. Quando terminar de adicionar as permissões à política, escolha Avançar.

7. Na página Revisar e criar, digite um nome de política e uma descrição (opcional) para a política que você está criando. Revise Permissões definidas nessa política para ver as permissões que são concedidas pela política.
8. Escolha Criar política para salvar sua nova política.

Depois de criar uma política, use as instruções a seguir para anexar a política a um perfil do IAM.

Para criar uma função que conceda permissões do Amplify a recursos específicos AWS

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação do console, escolha Roles (Perfis) e, em seguida, clique em Create role (Criar perfil).
3. Selecione o tipo de função Custom trust policy (Política de confiança personalizada).
4. Na seção Política de confiança personalizada, insira a política de confiança personalizada para o perfil. Uma política de confiança de perfil é obrigatória, e define as entidades principais em que você confia para assumir o perfil.

   Copie e cole a política de confiança a seguir para conceder permissão ao serviço do Amplify para assumir esse perfil.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

5. Resolva os avisos de segurança, erros ou avisos gerais gerados durante a validação da política e depois escolha Avançar.
6. Na página Adicionar permissões, pesquise pelo nome da política criada no procedimento anterior e selecione-a. Escolha Próximo.
7. Em Role name (Nome da função), insira um nome. Os nomes das funções devem ser exclusivos em seu Conta da AWS. Eles não são diferenciados por maiúsculas e minúsculas. Por exemplo, não é possível criar perfis denominados **`PRODROLE`** e **`prodrole`**. Como outros AWS recursos podem fazer referência à função, você não pode editar o nome da função após sua criação.
8. (Opcional) Em Descrição da função, insira uma descrição para a nova função.
9. (Opcional) Escolha Editar nas seções Etapa 1: selecionar entidades confiáveis ou Etapa 2: adicionar permissões para editar a política personalizada e as permissões do perfil.
10. Revise a função e escolha Criar função.

Adição de um perfil do IAM SSR Compute a uma aplicação do Amplify

Depois de criar uma função do IAM no seu Conta da AWS, você pode associá-la a um aplicativo no console do Amplify.

Para adicionar um perfil SSR Compute a uma aplicação no console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/
2. Na página Todas as aplicações, escolha o nome da aplicação à qual adicionar um perfil Compute.
3. No painel de navegação, escolha Configurações da aplicação, e, em seguida, escolha Perfis do IAM.
4. Na seção Perfil Compute, escolha Editar.
5. Na lista Perfil padrão, pesquise o nome do perfil que você deseja anexar e selecione-o. Neste exemplo, é possível escolher o nome do perfil que você criou no procedimento anterior. Por padrão, o perfil que você selecionar será associado a todas as ramificações da sua aplicação.

   Se a relação de confiança do perfil não estiver definida corretamente, você receberá um erro e não será capaz de adicionar o perfil.
6. (opcional) se a sua aplicação estiver em um repositório público e usar a criação automática de ramificações ou tiver pré-visualizações na Web para solicitações de pull ativadas, não

recomendamos o uso de um perfil no nível da aplicação. Em vez disso, atribua o perfil de computação somente às ramificações que exijam acesso a recursos específicos. Para substituir o comportamento padrão no nível da aplicação e anexar um perfil a uma ramificação específica, faça o seguinte:

a. Em Ramificação, selecione o nome da ramificação a ser usada.
b. Em Perfil Compute, selecione o nome do perfil a ser associado à ramificação.

7. Escolha Salvar.

Gerenciamento da segurança do perfil SSR Compute do IAM

A segurança é uma responsabilidade compartilhada entre você AWS e você. Você é responsável por configurar a aplicação para atender aos seus objetivos de segurança e conformidade. Isso inclui gerenciar seu perfil SSR Compute, que deve ser configurado para ter o conjunto mínimo de permissões necessárias para dar suporte ao seu caso de uso. As credenciais para o perfil SSR Compute que você especifica estão imediatamente disponíveis no runtime da sua função de SSR. Se seu código de SSR expuser essas credenciais, intencionalmente, devido a um bug ou ao permitir a execução remota de código (RCE), um usuário não autorizado poderá obter acesso ao perfil de SSR e suas permissões.

Quando uma aplicação em um repositório público usa um perfil SSR Compute e a criação automática de ramificações ou pré-visualizações na Web para solicitações de pull, é necessário cuidadosamente quais ramificações podem acessar o perfil. Recomendamos que você não use um perfil no nível da aplicação. Em vez disso, é necessário anexar um perfil de computação no nível da ramificação. Isso permite que você conceda permissões somente às ramificações que exijam acesso a recursos específicos.

Se as credenciais do seu perfil estiverem expostas, execute as ações a seguir para remover todo o acesso às credenciais do perfil.

1. Revogar todas as sessões

   Para obter instruções sobre como revogar imediatamente todas as permissões para as credenciais do perfil, consulte Revogação de credenciais de segurança temporárias de perfil do IAM.

2. Exclua o perfil do console do Amplify

   Essa ação tem efeito imediato. Não é necessário reimplantar sua aplicação.

Para excluir um perfil do Compute no console do Amplify

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify em. https://console.aws.amazon.com/amplify/
2. Na página Todas as aplicações, escolha o nome da aplicação da qual remover um perfil Compute.
3. No painel de navegação, escolha Configurações da aplicação, e, em seguida, escolha Perfis do IAM.
4. Na seção Perfil Compute, escolha Editar.
5. Para excluir o Perfil padrão, escolha o X à direita do nome do perfil.
6. Escolha Salvar.

Adicionar uma função de serviço com permissões para acessar o CloudWatch Logs

Amplify envia informações sobre seu tempo de execução de SSR para o Amazon CloudWatch Logs em seu. Conta da AWS Ao implantar um aplicativo SSR, o aplicativo requer um perfil de serviço IAM que o Amplify assume ao chamar outros serviços em seu nome. É possível permitir que a computação do Amplify Hosting crie automaticamente um perfil de serviço ou especificar um perfil que criou.

Se você optar por permitir que o Amplify crie uma função do IAM para você, a função já terá as permissões para criar CloudWatch registros. Se você criar sua própria função do IAM, precisará adicionar as seguintes permissões à sua política para permitir que o Amplify acesse o Amazon CloudWatch Logs.

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

Webhooks unificados para repositórios Git

O Amplify Hosting usa webhooks para iniciar automaticamente uma compilação após uma nova confirmação no seu repositório Git. O recurso unificado de webhooks melhora as integrações do Amplify com os provedores de Git e permite que você conecte mais aplicações do Amplify a um único repositório. Com webhooks unificados, o Amplify agora usa um único webhook por região para todas as aplicações associadas em seu repositório. Por exemplo, se seu repositório estiver conectado a aplicações nas regiões Leste dos EUA (Norte da Virgínia) e Oeste dos EUA (Oregon), você terá dois webhooks unificados.

Antes desse lançamento, o Amplify criara um novo webhook para cada aplicação associada a um repositório. Se você tivesse várias aplicações em um único repositório, poderia atingir os limites de webhook impostos por provedores individuais do Git e ser impedido de adicionar mais aplicações. Isso foi especialmente desafiador para equipes que trabalhavam em monorepos, onde existem vários projetos em um único repositório.

Os webhooks unificados oferecem os benefícios a seguir:

- Supere os limites de webhook do provedor de Git: é possível conectar quantas aplicações do Amplify precisar a um único repositório.
- Suporte aprimorado a monorepo: você tem mais flexibilidade e eficiência ao trabalhar com monorepos, onde vários projetos compartilham um único repositório.
- Gerenciamento simplificado: gerenciar várias aplicações do Amplify com um único webhook de repositório reduz a complexidade e os possíveis pontos de falha.
- Integração aprimorada do fluxo de trabalho: é possível usar os webhooks alocados pelo seu provedor de Git para outros fluxos de trabalho essenciais em seu processo de desenvolvimento.

Conceitos básicos dos webhooks unificados

Criação de uma nova aplicação

Quando você implanta uma nova aplicação no Amplify Hosting a partir de um repositório de Git, o recurso unificado de webhooks é implementado automaticamente em seu repositório. Para obter instruções sobre como criar uma nova aplicação, consulte Noções básicas da implantação de uma aplicação no Amplify Hosting.

Atualização de uma aplicação existente

Para aplicações do Amplify existentes, é necessário reconectar seu repositório de Git à sua aplicação para substituir os webhooks existentes por um webhook unificado. Se você já atingiu o número máximo de webhooks permitido pelo seu provedor de Git, a migração para o webhook unificado pode não obter êxito. Nesse caso, remova manualmente pelo menos um webhook existente antes de se reconectar.

É possível ter várias aplicações em um repositório que sejam implantadas em diferentes regiões da AWS . Como as operações do Amplify são baseadas na região, a migração para um webhook unificado ocorre apenas para os webhooks na região em que você reconectou sua aplicação do Amplify. Como resultado, é possível ver webhooks baseados em ID da aplicação e webhooks unificados baseados em região em seu repositório.

Use as instruções a seguir para migrar uma aplicação existente do Amplify para um webhook unificado.

Para migrar uma aplicação existente do Amplify para um webhook unificado

1. Faça login Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha a aplicação que você deseja migrar para um webhook unificado.
3. No painel de navegação, escolha Configurações da aplicação, e, em seguida, escolha Configurações da ramificação.
4. Na página Configurações da ramificação, escolha Reconectar repositório.
5. Para verificar se a migração obteve êxito para o webhook unificado, navegue até as configurações do webhook no seu repositório de Git. É necessário ver um único URL de webhook no formato `https://amplify-webhooks.`*`Region`*`.amazonaws.com/`*`git-provider`*.

Segurança no Amplify

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Third-party auditores testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade aplicáveis AWS Amplify, consulte AWS Serviços no escopo do programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amplify. Os tópicos a seguir mostram como configurar o Amplify para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do Amplify.

Tópicos

Gerenciamento de identidade e acesso para o Amplify

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para utilizar os recursos do Amplify. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas de identidade e acesso do Amplify).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como o Amplify funciona com o IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Identity-based exemplos de políticas para Amplify)

Autenticação com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login

único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário-raiz

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte O que é o IAM Identity Center? no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Identity-based políticas

Identity-based políticas são documentos de políticas de permissões JSON que você anexa a uma identidade (usuário, grupo ou função). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Identity-based as políticas podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Resource-based políticas

Resource-based políticas são documentos de política JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

Resource-based políticas são políticas embutidas que estão localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de Controle de Serviços (SCPs): as SCPs especificam o número máximo de permissões para uma organização ou uma unidade organizacional no AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs): definem o número máximo de permissões disponíveis para recursos em suas contas. Consulte mais informações em Resource control policies (RCPs) no Guia do usuário do AWS Organizations .
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o Amplify funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amplify, saiba quais recursos do IAM estão disponíveis para uso com o Amplify.

Atributos do IAM que é possível usar com o Amplify

Recurso do IAM	Suporte do Amplify
Identity-based políticas	Sim
Resource-based políticas	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Sessões de acesso direto (FAS)	Sim
Perfis de serviço	Sim
Service-linked funções	Não

Para ter uma visão de alto nível de como o Amplify e AWS outros serviços funcionam com a maioria dos recursos do IAM, AWS consulte os serviços que funcionam com o IAM no Guia do usuário do IAM.

Identity-based políticas para Amplify

Compatível com políticas baseadas em identidade: sim

Identity-based políticas são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como um usuário do IAM, um grupo de usuários ou uma função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais atributos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Identity-based exemplos de políticas para Amplify

Para visualizar exemplos de políticas baseadas em identidade do Amplify, consulte Identity-based exemplos de políticas para Amplify.

Resource-based políticas dentro do Amplify

Compatibilidade com políticas baseadas em recursos: não

Resource-based políticas são documentos de política JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações políticas para Amplify

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para obter uma lista das ações do Amplify, consulte Ações definidas pelo AWS Amplify na Referência de autorização do serviço.

As ações de política no Amplify usam o seguinte prefixo antes da ação:

```
amplify
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

```
"Action": [
      "amplify:action1",
      "amplify:action2"
         ]
```

Para visualizar exemplos de políticas baseadas em identidade do Amplify, consulte Identity-based exemplos de políticas para Amplify.

Recursos de políticas para Amplify

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para obter uma lista dos tipos de recursos do Amplify e seus ARNs, consulte Tipos de recursos definidos pelo AWS Amplify na Referência de autorização do serviço. Para saber com quais ações é possível especificar o ARN de cada atributo, consulte Ações definidas pelo AWS Amplify.

Para visualizar exemplos de políticas baseadas em identidade do Amplify, consulte Identity-based exemplos de políticas para Amplify.

Chaves de condição de política para o Amplify

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para obter uma lista de chaves de condição do Amplify, consulte Chaves de condição do AWS Amplify na Referência de autorização do serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Ações definidas por AWS Amplify.

Para visualizar exemplos de políticas baseadas em identidade do Amplify, consulte Identity-based exemplos de políticas para Amplify.

Listas de controle de acesso (ACLs) no Amplify

Compatível com ACLs: não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Attribute-based controle de acesso (ABAC) com Amplify

Compatível com ABAC (tags em políticas): parcial

Attribute-based controle de acesso (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Amplify

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Sessões de acesso direto para o Amplify

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Perfis de serviço do Amplify

Compatível com perfis de serviço: sim

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amplify. Edite perfis de serviço somente quando o Amplify fornecer orientação para isso.

Service-linked funções para Amplify

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir a função de realizar uma ação em seu nome. Service-linked as funções aparecem no seu Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculadas a serviços, consulte AWS Serviços que funcionam com o IAM no Guia do usuário do IAM.. Encontre um serviço na tabela que inclua um `Yes` na coluna de Service-linked função. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Identity-based exemplos de políticas para Amplify

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do Amplify. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para saber como criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criação de políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recurso definidos pelo Amplify, incluindo o formato dos ARNs para cada tipo de recurso, consulte Ações, recursos e chaves de condição do AWS Amplify na Referência de autorização do serviço.

Tópicos

- Práticas recomendadas de política
- Usando o console do Amplify

- Permitir que os usuários visualizem suas próprias permissões

Práticas recomendadas de política

Identity-based as políticas determinam se alguém pode criar, acessar ou excluir recursos do Amplify em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e passe para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando

as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o console do Amplify

Para acessar o AWS Amplify console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amplify em seu. Conta da AWS Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Com o lançamento do Amplify Studio, a exclusão de um aplicativo ou backend requer ambas as permissões `amplify` e `amplifybackend`. Se uma política do IAM fornecer somente permissões `amplify`, o usuário receberá um erro de permissões ao tentar excluir um aplicativo. Se você for um administrador que estiver escrevendo políticas, determine as permissões corretas para dar aos usuários que precisam realizar ações de exclusão.

Para garantir que usuários e funções ainda possam usar o console do Amplify, anexe também o Amplify `ConsoleAccess` ou a política `ReadOnly` AWS gerenciada às entidades. Para obter informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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 políticas gerenciadas para AWS Amplify

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) for lançada ou novas operações de API forem disponibilizadas para serviços existentes.

Para saber mais, consulte AWS Políticas gerenciadas pela no Guia do usuário do IAM.

Política gerenciada pela AWS: AdministratorAccess-Amplify

É possível anexar a política `AdministratorAccess-Amplify` às suas identidades do IAM. Essa política é anexada a um perfil de serviço que permite que o Amplify realize ações em seu nome.

Ao implantar um back-end no console do Amplify, você deve criar `Amplify-Backend Deployment` uma função de serviço que o Amplify usa para criar e gerenciar recursos. AWS O IAM anexa a política gerenciada `AdministratorAccess-Amplify` ao perfil de serviço `Amplify-Backend Deployment`.

Essa política concede permissões administrativas à conta e, ao mesmo tempo, permite explicitamente o acesso direto aos recursos que os aplicativos do Amplify exigem para criar e gerenciar backends.

Detalhes das permissões

Essa política fornece acesso a vários AWS serviços, incluindo ações do IAM. Essas ações permitem que identidades com essa política sejam usadas AWS Identity and Access Management para criar outras identidades com qualquer permissão. Isso permite o escalonamento de permissões e essa política deve ser considerada tão poderosa quanto a política `AdministratorAccess`.

Essa política concede permissões de ação `iam:PassRole` para todos os recursos. Isso é necessário para oferecer suporte à configuração de grupos de usuários do Amazon Cognito.

Para visualizar as permissões para esta política, consulte AdministratorAccess-Amplify na Referência de políticas gerenciadas pela AWS .

AWS política gerenciada: AmplifyBackendDeployFullAccess

É possível anexar a política `AmplifyBackendDeployFullAccess` às suas identidades do IAM.

Esta política concede ao Amplify permissões de acesso total para implantar recursos de back-end do Amplify usando o. AWS Cloud Development Kit (AWS CDK) As permissões são transferidas para as AWS CDK funções que têm as permissões de `AdministratorAccess` política necessárias.

Detalhes das permissões

Esta política inclui permissões para fazer as ações a seguir.

- `Amplify`: recuperar metadados sobre aplicações implantadas.
- `CloudFormation`: criar, atualizar e excluir pilhas gerenciadas do Amplify.
- `SSM`: criar, atualizar e excluir os parâmetros `String` e `SecureString` do SSM Parameter Store gerenciados pelo Amplify.
- `AWS AppSync`— Atualize e recupere recursos de AWS AppSync esquema, resolvedor e função. O objetivo é oferecer suporte à funcionalidade de hotswapping do sandbox Gen 2.
- `Lambda`: atualizar e recuperar a configuração das funções gerenciadas do Amplify. O objetivo é oferecer suporte à funcionalidade de hotswapping do sandbox Gen 2.

  Recupere as tags de uma função do Lambda. O objetivo é oferecer suporte às funções do Lambda definidas pelos clientes.
- `Amazon S3`: recuperar os ativos de implantação do Amplify.
- `AWS Security Token Service`— Permite que a AWS Cloud Development Kit (AWS CDK) CLI assuma a função de implantação.
- `Amazon RDS`: ler metadados de instâncias de banco de dados, clusters e proxies.
- `Amazon EC2`: ler as informações da zona de disponibilidade de uma sub-rede.
- `CloudWatch Logs`: recuperar os logs da função do Lambda de um cliente. O objetivo é permitir que um ambiente sandbox de desenvolvimento em nuvem da Amplify transmita os logs de uma função do Lambda para o terminal do cliente.

Para visualizar as permissões para esta política, consulte AmplifyBackendDeployFullAccess na Referência de políticas gerenciadas pela AWS .

Amplifique as atualizações para AWS políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Amplify desde que esse serviço começou a rastrear essas mudanças. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página Histórico do documento para AWS Amplify.

Alteração	Descrição	Data
AmplifyBackendDeployFullAccess: atualizar para uma política existente	Adicione acesso de leitura ao recurso `logs:FilterLogEvents` para permitir que o Amplify transmita logs de funções em que um grupo de logs personalizado tenha sido criado. Essa é uma extensão da capacidade existente de transmitir os logs de uma função do Lambda.	14 de novembro de 2024
AmplifyBackendDeployFullAccess – atualização para uma política existente	Adicione acesso de leitura aos recursos `lambda:ListTags` e `logs:FilterLogEvents` para oferecer suporte às funções do Lambda definidas pelos clientes. Essas permissões permitem que um ambiente sandbox de desenvolvimento em nuvem da Amplify transmita os logs de uma função do Lambda para o terminal do cliente.	18 de julho de 2024
AmplifyBackendDeployFullAccess – atualização para uma política existente	Adicione acesso de leitura ao `arn:aws:ssm:*:*:parameter/cdk-bootstrap/*` recurso para permitir que o Amplify detecte a versão bootstrap do CDK na conta de um cliente.	31 de maio de 2024

Alteração	Descrição	Data
AmplifyBackendDeployFullAccess – atualização para uma política existente	Adicione uma nova declaração de política `AmplifyDiscoverRDSVpcConfig` com permissões somente de leitura do Amazon RDS e do Amazon EC2, definidas de acordo com as condições do recurso e da conta. Essas permissões oferecem suporte ao comando `npx amplify generate schema-from-database` do Amplify Gen 2o , que permite aos clientes gerar um esquema de dados Typescript a partir de um banco de dados SQL existente . Adicione as permissões `rds:DescribeDBProxies` , `rds:DescribeDBInstances` , `rds:DescribeDBClusters` , `rds:DescribeDBSubnetGroups` e `ec2:DescribeSubnets` . O `npx amplify generate schema-from-database` comando exige essas permissões para verificar se um host de banco de dados especificado está hospedado no Amazon RDS e gerar automaticamente a configuração da Amazon VPC	17 de abril de 2024

Alteração	Descrição	Data
	necessária para provisionar os outros recursos necessários para configurar uma AWS AppSync API apoiada por um banco de dados SQL.	
AmplifyBackendDeployFullAccess – atualização para uma política existente	Adicione a ação de política `cloudformation:DeleteStack` para oferecer suporte à exclusão da pilha quando a API `DeleteBranch` for chamada. Adicione a ação de política `lambda:GetFunction` para oferecer suporte às funções de hotswapping. Adicione a ação de política `lambda:UpdateFunctionConfiguration` para apoiar atualizações na função do Lambda.	5 de abril de 2024
AdministratorAccess-Amplify – atualização para uma política existente	Adicione as `cloudformation:UnTagResource` permissões `cloudformation:TagResource` e para suportar chamadas para CloudFormation APIs.	4 de abril de 2024

Alteração	Descrição	Data
AmplifyBackendDeployFullAccess – atualização para uma política existente	Adicione a ação `lambda:InvokeFunction` política para oferecer suporte ao AWS Cloud Development Kit (AWS CDK) hotswapping. Ele AWS CDK faz chamadas diretas para uma função Lambda para realizar a troca de ativos do Amazon S3. Adicione a ação de política `lambda:UpdateFunctionCode` para oferecer suporte às funções de hotswapping.	02 de janeiro de 2024
AmplifyBackendDeployFullAccess – atualização para uma política existente	Adicione ações de política para viabilizar a operação `UpdateApiKey` . Isso é necessário para permitir uma implantação bem-sucedida da aplicação após sair e reiniciar o sandbox sem excluir recursos.	17 de novembro de 2023
AmplifyBackendDeployFullAccess – atualização para uma política existente	Adicione a permissão `amplify:GetBackendEnvironment` para apoiar a implantação do aplicativo Amplify.	6 de novembro de 2023

Alteração	Descrição	Data
AmplifyBackendDeployFullAccess – Nova política	O Amplify adicionou uma nova política com as permissões mínimas necessárias para implantar os recursos de backend do Amplify.	8 de outubro de 2023
AdministratorAccess-Amplify – atualização para uma política existente	Adicione a permissão `ecr:DescribeRepositories` exigida pela CLI (Interface de Linha de Comando) do Amplify.	1º de junho de 2023

Alteração	Descrição	Data
AdministratorAccess-Amplify – atualização para uma política existente	Adicione uma ação de política para apoiar a remoção de tags de um recurso AWS AppSync . Adicione uma ação política para apoiar o recurso Amazon Polly. Adicione uma ação de política para oferecer suporte à atualização da configuração do OpenSearch domínio. Adicione uma ação de política para apoiar a remoção de tags de um perfil AWS Identity and Access Management . Adicione uma ação de política para apoiar a remoção de tags de um recurso Amazon DynamoDB. Adicione as permissões `cloudfront:GetCloudFrontOriginAccessIdentity` e `cloudfront:GetCloudFrontOriginAccessIdentityConfig` ao bloco de instruções para apoiar os fluxos de trabalho `CLISDKCalls` de publicação e hospedagem do Amplify.	24 de fevereiro de 2023

Alteração	Descrição	Data
	Adicione a permissão `s3:PutBucketPublicAccessBlock` ao bloco de instruções para permitir que o `CLIManageviaCFNPolicy` apoie a melhor prática de segurança do Amazon S3 AWS CLI de habilitar o recurso Bloqueio de Acesso Público do Amazon S3 em buckets internos. Adicione a `cloudformation:DescribeStacks` permissão ao bloco de instruções `CLISDKCalls` para permitir a recuperação das CloudFormation pilhas dos clientes em novas tentativas no processador de back-end do Amplify para evitar a duplicação de execuções se uma pilha estiver sendo atualizada. Adicione a permissão `cloudformation:ListStacks` ao bloco de instruções `CLICloudformationPolicy` . Essa permissão é necessária para apoiar totalmente a CloudFormation DescribeStacks ação.	

Alteração	Descrição	Data
AdministratorAccess-Amplify – atualização para uma política existente	Adicione ações políticas para permitir que o recurso de renderização do lado do servidor do Amplify envie as métricas do aplicativo para as de um cliente. CloudWatch Conta da AWS	30 de agosto de 2022
AdministratorAccess-Amplify – atualização para uma política existente	Adicione ações de política para bloquear o acesso público ao bucket do Amazon S3 de implantação do Amplify.	27 de abril de 2022
AdministratorAccess-Amplify – atualização para uma política existente	Adicione uma ação para permitir que os clientes excluam seus aplicativos renderizados do lado do servidor (SSR). Isso também permite que a CloudFront distribuição correspondente seja excluída com sucesso. Adicione uma ação para permitir que os clientes especifiquem uma função do Lambda diferente para lidar com eventos de uma fonte de eventos existente usando a CLI do Amplify. Com essas mudanças, AWS Lambda será capaz de realizar a UpdateEventSourceMappingação.	17 de abril de 2022

Alteração	Descrição	Data
AdministratorAccess-Amplify – atualização para uma política existente	Adicione uma ação política para ativar as ações do Amplify UI Builder em todos os recursos.	2 de dezembro de 2021
AdministratorAccess-Amplify – atualização para uma política existente	Adicione ações de política para apoiar o atributo de autenticação do Amazon Cognito que usa provedores de identidade social. Adicione uma ação política para oferecer suporte às camadas Lambda. Adicione uma ação política para apoiar a categoria Amplify Storage.	8 de novembro de 2021

Alteração	Descrição	Data
AdministratorAccess-Amplify – atualização para uma política existente	Adicione ações do Amazon Lex para apoiar a categoria Amplify Interactions. Adicione ações do Amazon Rekognition para apoiar a categoria Amplify Predictions. Adicione uma ação do Amazon Cognito para oferecer suporte à configuração de MFA nos grupos de usuários do Amazon Cognito. Adicione CloudFormation ações ao suporte CloudForm ation StackSets. Adicione ações do Amazon Location Service para apoiar a categoria Amplify Geo. Adicione uma ação do Lambda para oferecer suporte às camadas do Lambda no Amplify. Adicione ações de CloudWatc h registros para apoiar CloudWatch eventos. Adicione ações do Amazon S3 para oferecer suporte à categoria Amplify Storage. Adicione ações de política para oferecer suporte a	27 de setembro de 2021

Alteração	Descrição	Data
	aplicativos renderizados do lado do servidor (SSR).	

Alteração	Descrição	Data
AdministratorAccess-Amplify – atualização para uma política existente	Consolide todas as ações do Amplify em uma única ação `amplify:*` . Adicione uma ação do Amazon S3 para oferecer suporte à criptografia de buckets Amazon S3 do cliente. Adicione ações de limite de permissão do IAM para oferecer suporte aos aplicativos do Amplify que têm limites de permissão ativados. Adicione ações do Amazon SNS para oferecer suporte à visualização de números de telefone de origem e à visualização, criação, verificação e exclusão de números de telefone de destino. Amplify Studio: adicione ações do Amazon Cognito AWS Lambda, IAM CloudFormation e políticas para permitir o gerenciamento de back-ends no console do Amplify e no Amplify Studio. Adicione uma declaração de política AWS Systems Manager (SSM) para	28 de julho de 2021

Alteração	Descrição	Data
	gerenciar os segredos do ambiente do Amplify. Adicione uma CloudFormation `ListResources` ação para suportar camadas Lambda para aplicativos Amplify.	
O Amplify iniciou o rastreamento das alterações	A Amplify começou a monitorar as mudanças em suas políticas AWS gerenciadas.	28 de julho de 2021

Solução de problemas de identidade e acesso do Amplify

Use as seguintes informações para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o Amplify e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no Amplify
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha AWS conta para acessar meus recursos do Amplify

Não tenho autorização para executar uma ação no Amplify

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `amplify:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplify:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `amplify:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Com o lançamento do Amplify Studio, a exclusão de um aplicativo ou backend requer ambas as permissões `amplify` e `amplifybackend`. Se um administrador tiver escrito uma política do IAM que fornece somente permissões `amplify`, você receberá um erro de permissões ao tentar excluir um aplicativo.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para excluir um recurso do *`example-amplify-app`* fictício, mas não tem as permissões do `amplifybackend:`*`RemoveAllBackends`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplifybackend;:RemoveAllBackends on resource: example-amplify-app
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *`example-amplify-app`* usando a ação `amplifybackend:`*`RemoveAllBackends`*.

Não estou autorizado a realizar iam: PassRole

Se receber uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, suas políticas devem ser atualizadas para permitir a transmissão de um perfil ao Amplify.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro exemplificado a seguir ocorre quando um usuário do IAM chamada `marymajor` tenta usar o console para executar uma ação no Amplify. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha AWS conta para acessar meus recursos do Amplify

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), é possível usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amplify é compatível com esses recursos, consulte Como o Amplify funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Proteção de dados no Amplify

AWS Amplify está em conformidade com o modelo de responsabilidade AWS compartilhada de , que inclui regulamentos e diretrizes para proteção de dados. AWS é responsável por proteger a infraestrutura global que executa todos os AWS serviços. AWS mantém o controle sobre os dados hospedados nessa infraestrutura, incluindo os controles de configuração de segurança para lidar com o conteúdo do cliente e os dados pessoais. AWS clientes e parceiros da APN, atuando como

controladores ou processadores de dados, são responsáveis por todos os dados pessoais que colocam na AWS nuvem.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções AWS de criptografia, juntamente com todos os controles de segurança padrão nos AWS serviços.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados pessoais armazenados no Amazon S3.

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como números de conta dos seus clientes, em campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Amplify ou outros AWS serviços usando o console, a API ou AWS os AWS CLI SDKs. Todos os dados que você insere no Amplify ou em outros serviços podem ser separados para inclusão em logs de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar a solicitação a esse servidor.

Para mais informações sobre proteção de dados, consulte a publicação Modelo de responsabilidade compartilhada da AWS e do GDPR no Blog de segurança da AWS .

Criptografia inativa

A criptografia em repouso refere-se à proteção de dados contra acesso não autorizado criptografando dados enquanto estão armazenados. O Amplify criptografa os artefatos de construção de um aplicativo por padrão usando o Amazon AWS KMS keys S3 que são gerenciados pelo. AWS Key Management Service

Amplify usa CloudFront a Amazon para servir seu aplicativo aos seus clientes. CloudFront usa SSDs que são criptografados para pontos de presença (PoPs) de localização periférica e volumes criptografados do EBS para caches regionais de borda (RECs). O código e a configuração da função no CloudFront Functions são sempre armazenados em um formato criptografado nos SSDs

criptografados, nos PoPs do ponto de borda e em outros locais de armazenamento usados pelo. CloudFront

Criptografia em trânsito

Criptografia em trânsito refere-se a impedir os dados de serem interceptados enquanto eles se movem entre endpoints de comunicação. O Amplify Hosting fornece criptografia para dados em trânsito por padrão. Todas as comunicações entre clientes e o Amplify e entre o Amplify e suas dependências downstream são protegidas por meio de conexões TLS assinadas usando o processo de assinatura do Signature versão 4. Todos os endpoints do Amplify Hosting usam SHA-256 certificados gerenciados por. Autoridade de Certificação Privada da AWS Para obter mais informações, consulte Processo de assinatura do Signature versão 4 e O que é o Autoridade de Certificação Privada da AWS.

Gerenciamento das chaves de criptografia

AWS Key Management Service (KMS) é um serviço gerenciado para criar e controlar AWS KMS keys as chaves de criptografia usadas para criptografar dados do cliente. AWS Amplify gera e gerencia chaves criptográficas para criptografar dados em nome dos clientes. Não há chaves de criptografia para você gerenciar.

Validação de conformidade para AWS Amplify

Third-party os auditores avaliam a segurança e a conformidade AWS Amplify como parte de vários programas de AWS conformidade. Isso inclui SOC, PCI, ISO, HIPAA, MTCS, C5, ENS High, OSPAR K-ISMS, HITRUST CSF e FINMA.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentos aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a documentação AWS de segurança.

Segurança de infraestrutura em AWS Amplify

Como serviço gerenciado, AWS Amplify é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o Amplify pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Suítes de criptografia com sigilo direto perfeito (PFS), como DHE (efêmero) ou ECDHE (curva elíptica efêmera Diffie-Hellman). Diffie-Hellman A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Registro e monitoramento de eventos de segurança no Amplify

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e o desempenho do Amplify e de suas outras AWS soluções. AWS fornece as seguintes ferramentas de monitoramento para observar o Amplify, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora em tempo real seus AWS recursos e os aplicativos nos quais você executa AWS. É possível coletar e rastrear métricas, criar painéis personalizados e definir alarmes que notificam você ou que realizam ações quando uma certa métrica atinge um limite especificado. Por exemplo, você pode CloudWatch monitorar o uso da CPU ou outras métricas de suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações sobre o uso de CloudWatch métricas e alarmes com o Amplify, consulte. Monitoramento de uma aplicação do Amplify
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log das instâncias do Amazon EC2 e de outras fontes. AWS CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.
- AWS CloudTrailcaptura chamadas de API e eventos relacionados feitos por ou em nome de sua AWS conta e entrega os arquivos de log para um bucket do Amazon Simple Storage Service

(Amazon S3) especificado por você. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte Registrando chamadas da API Amplify usando AWS CloudTrail.

- EventBridgeA Amazon é um serviço de ônibus de eventos sem servidor que facilita a conexão de seus aplicativos com dados de várias fontes. EventBridge fornece um fluxo de dados em tempo real de seus próprios aplicativos, aplicativos Software-as-a-Service (SaaS) e AWS serviços e encaminha esses dados para destinos como. AWS Lambda Isso permite monitorar eventos que ocorrem em serviços e crie arquiteturas orientadas a eventos. Para obter mais informações, consulte o Guia EventBridge do usuário da Amazon.

Cross-service prevenção delegada confusa

O problema "confused deputy" é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executar a ação. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. Cross-service a representação pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar `aws:SourceArn`as chaves de contexto de condição `aws:SourceAccount`global nas políticas de recursos para limitar as permissões que AWS Amplify concedem outro serviço ao recurso. Se você utilizar ambas as chaves de contexto de condição global, o valor `aws:SourceAccount` e a conta `aws:SourceArn` no valor deverão utilizar o mesmo ID de conta quando utilizados na mesma instrução de política.

O valor de `aws:SourceArn` deve ser o ARN da filial do aplicativo Amplify. Especifique esse valor no formato `arn:`*`Partition`*`:amplify:`*`Region`*`:`*`Account`*`:apps/`*`AppId`*`/branches/`*`BranchName`*.

A maneira mais eficaz de se proteger do problema ‘confused deputy’ é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou se especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, .`arn:aws:`*`servicename`*`::`*`123456789012`*`:*`

O exemplo a seguir mostra uma política de confiança de perfis que é possível aplicar para limitar o acesso a qualquer aplicativo Amplify em sua conta e evitar o problema de “confused deputy”. Para usar essa política, substitua o texto vermelho em itálico na política de exemplo por suas próprias informações.

JSON

```
{
  "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"
      }
    }
  }
}
```

O exemplo a seguir mostra uma política de confiança de perfis que é possível aplicar para limitar o acesso a um aplicativo Amplify específico em sua conta e evitar o problema de “confused deputy”. Para usar essa política, substitua o texto vermelho em itálico na política de exemplo por suas próprias informações.

JSON

```
{
  "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"
      }
    }
  }
}
```

Práticas recomendadas de segurança para o Amplify

O Amplify oferece uma série de recursos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como recomendações úteis em vez de requisitos.

Usando cookies com o domínio padrão do Amplify

Quando você usa o Amplify para implantar um aplicativo web, o Amplify o hospeda para você no domínio padrão `amplifyapp.com`. É possível visualizar seu aplicativo em um URL formatado como `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`.

Para aumentar a segurança de seus aplicativos do Amplify, o domínio amplifyapp.com é registrado na Lista Pública de Sufixos (PSL). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para seus aplicativos do Amplify. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a Set-Cookiepágina na Rede de Desenvolvedores da Mozilla.

Service quotas do Amplify Hosting

A seguir estão as cotas de serviço para AWS Amplify hospedagem. As service quotas, também chamadas de limites, correspondem ao número máximo de recursos ou operações de serviço para sua conta da Conta da AWS.

Contas da AWS Os novos reduziram os aplicativos e as cotas de trabalhos simultâneos. AWS aumenta essas cotas automaticamente com base no seu uso. Também é possível solicitar um aumento da cota.

O console do Service Quotas fornece informações sobre as cotas para sua conta. É possível usar o console do Service Quotas para visualizar cotas padrão e solicitar aumentos de cota para cotas ajustáveis. Para obter mais informações, consulte Solicitar um aumento da cota no Guia do usuário do Service Quotas.

Nome	Padrão	Ajustá	Description
Apps	Cada região com suporte: 25	Sim	O número máximo de aplicativos que você pode criar no AWS Amplify Console nessa conta na região atual.
Ramificações por aplicativo	Cada região compatível: 50	Não	O número máximo de ramificações por aplicativo que podem ser criadas nessa conta na região atual.
Tamanho do artefato de compilação	Cada região compatível: 5 gigabytes	Não	O tamanho máximo (em GB) de um artefato de compilação de aplicativo. Um artefato de construção é implantado pelo AWS Amplify Console após uma construção.

Nome	Padrão	Ajustá	Description
Tamanho do artefato de cache	Cada região compatível: 5 gigabytes	Não	O tamanho máximo (em GB) de um artefato de cache.
Tarefas simultâneas	Cada região compatível: 5	Sim	O número máximo de tarefas simultâneas que podem ser criados nesta conta na região atual.
Domínios por aplicativo	Cada região compatível: 5	Sim	O número máximo de domínios que podem ser criados nessa conta na região atual.
Tamanho do artefato do cache do ambiente	Cada região compatível: 5 gigabytes	Não	O tamanho máximo (em GB) do artefato de cache do ambiente.
Tamanho do arquivo ZIP de implantaç ão manual	Cada região compatível: 5 gigabytes	Não	O tamanho máximo (em GB) de um arquivo ZIP de implantação manual.
Máximo de criações de aplicativos por hora	Cada região compatível: 25	Não	O número máximo de aplicativos que você pode criar no AWS Amplify Console por hora nessa conta na região atual.

Nome	Padrão	Ajustá	Description
Tokens de solicitações por segundo	Cada região com suporte: 20.000	Sim	O número máximo de tokens de solicitação por segundo para uma aplicação. O Amplify Hosting aloca tokens para solicitações com base na quantidade de recursos (tempo de processam ento e transferência de dados) que elas consomem.
Subdomínios por domínio	Cada região compatível: 50	Não	O número máximo de subdomínios por domínio que podem ser criados nessa conta na região atual.
Webhooks por aplicativo	Cada região compatível: 50	Sim	O número máximo de webhooks que podem ser criados nessa conta na região atual.

Para obter mais informações sobre as cotas de serviço do Amplify, consulte Endpoints e cotas do AWS Amplify na Referência geral da AWS.

Solução de problemas do Amplify Hosting

Se você encontrar erros ou problemas de implantação ao trabalhar com o Amplify Hosting, consulte os tópicos desta seção.

Tópicos

Solução de problemas gerais do Amplify

As informações a seguir podem ajudá-lo a solucionar problemas gerais com o Amplify Hosting.

Tópicos

Código de status 429 de HTTP (excesso de solicitações)

O Amplify controla o número de solicitações por segundo (RPS) para seu site com base no tempo de processamento e na transferência de dados que as solicitações recebidas consomem. Se a sua aplicação retornar um código de status 429 de HTTP, as solicitações recebidas excederão a

quantidade de tempo de processamento e transferência de dados atribuída à sua aplicação. Esse limite de aplicação é gerenciado pela cota de serviço `REQUEST_TOKENS_PER_SECOND` do Amplify. Para obter mais informações sobre cotas, consulte Service quotas do Amplify Hosting.

Para corrigir esse problema, recomendamos otimizar sua aplicação para reduzir a duração da solicitação e a transferência de dados para aumentar o RPS da aplicação. Por exemplo, com os mesmos 20.000 tokens, uma página SSR altamente otimizada que responda em 100 milissegundos pode suportar um RPS mais alto em comparação com uma página com latência superior a 200 milissegundos.

Da mesma forma, uma aplicação que retorne um tamanho de resposta de 1 MB consumirá mais tokens do que uma aplicação que retorne um tamanho de resposta de 250 KB.

Também recomendamos que você aproveite o CloudFront cache da Amazon configurando Cache-Control cabeçalhos que maximizem o tempo em que uma determinada resposta é mantida no cache. As solicitações atendidas pelo CloudFront cache não contam para o limite de taxa. Cada CloudFront distribuição pode lidar com até 250.000 solicitações por segundo, permitindo que você escale muito seu aplicativo usando o cache. Para obter mais informações sobre o CloudFront cache, consulte Otimizando o armazenamento em cache e a disponibilidade no Amazon CloudFront Developer Guide.

O console do Amplify não exibe o status de compilação e a hora da última atualização da minha aplicação

Quando você navega até a página Todas as aplicações no console do Amplify, um quadro é exibido para cada uma das suas aplicações na região atual. Se você não vê o status de compilação, como Implantado, e a hora da Ultima atualização exibida para uma aplicação, a aplicação não tem uma ramificação de estágio de `Production` associada a ele.

Para listar as aplicações no console, o Amplify usa a API `ListApps`. O Amplify usa o atributo `ProductionBranch.status` para exibir o status da compilação e o atributo `ProductionBranch.lastDeployTime` para exibir a hora da última atualização. Para obter mais informações sobre essa API, consulte a ProductionBranchdocumentação da API Amplify Hosting.

Use as instruções a seguir para associar um estágio de `Production` à ramificação da aplicação.

1. Faça login no console do Amplify.
2. Na página Todas as aplicações, escolha a aplicação que deseja atualizar.

3. No painel de navegação, escolha Configurações da aplicação, e, em seguida, Configurações da ramificação.
4. Na seção Configurações da ramificação, escolha Editar.
5. Em Ramificação de produção, selecione a ramificação que você deseja usar.
6. Escolha Salvar.
7. Volte para a página Todas as aplicações. O status da compilação e a hora da última atualização agora devem ser exibidos para sua aplicação.

As visualizações na Web não estão sendo criadas para novas solicitações de pull

O recurso de visualizações na Web permite que você visualize as alterações das solicitações de pull antes de mesclá-las em uma ramificação de integração. Uma visualização prévia da Web implanta cada solicitação de pull feita em seu repositório em um URL de visualização exclusiva que é diferente do URL que seu site principal usa.

Se você ativou as visualizações na Web para sua aplicação, mas elas não estão sendo criadas para novos PRs, investigue se uma das situações a seguir é a causa do problema.

1. Verifique se a sua aplicação atingiu a cota máxima de serviço `Branches per app`. Para obter mais informações sobre cotas, consulte Service quotas do Amplify Hosting.

   Para permanecer dentro da cota padrão de 50 ramificações por aplicação, considere ativar a exclusão automática de ramificações em sua aplicação. Isso evitará que você acumule ramificações na sua conta que não existem mais no seu repositório.
2. Se você estiver usando um GitHub repositório público e seu aplicativo Amplify tiver uma função de serviço do IAM associada a ele, o Amplify não cria visualizações por motivos de segurança. Por exemplo, aplicativos com backends e aplicativos implantados na plataforma de `WEB_COMPUTE` hospedagem exigem um perfil de serviço do IAM. Portanto, não é possível habilitar visualizações na web para esses tipos de aplicativos se o repositório deles for público.

   Para permitir que as visualizações na web funcionem para seu aplicativo, você pode desassociar a função de serviço (se o aplicativo não tiver um back-end ou não for um `WEB_COMPUTE` aplicativo) ou tornar o GitHub repositório privado.

Minha implantação manual está bloqueada com um status pendente no console do Amplify

As implantações manuais permitem que você publique sua aplicação da Web com o Amplify Hosting sem conectar um provedor do Git. É possível usar uma das quatro opções de implantação a seguir.

1. Arraste e solte a pasta da aplicação no console do Amplify.
2. Arraste e solte um arquivo .zip (que contenha os artefatos de construção do seu site) no console do Amplify.
3. Faça o upload de um arquivo .zip (que contenha os artefatos de construção do seu site) em um bucket do Amazon S3 e conecte o bucket a uma aplicação no console do Amplify.
4. Use um URL público que aponte para um arquivo .zip (que contenha os artefatos de construção do seu site) no console do Amplify.

Estamos cientes dos problemas com a funcionalidade de arrastar e soltar ao usar uma pasta de aplicação para uma implantação manual no console do Amplify. Essas implantações podem apresentar falha pelos motivos a seguir.

- Ocorrem problemas transitórios de rede.
- Há uma alteração local nos arquivos durante o upload.
- A sessão do navegador tenta fazer o upload de uma grande quantidade de ativos estáticos simultaneamente.

Enquanto trabalhamos para melhorar a confiabilidade de nossos uploads de arrastar e soltar, recomendamos que você use um arquivo .zip em vez de arrastar e soltar as pastas da aplicação.

É altamente recomendável fazer o upload de um arquivo .zip em um bucket do Amazon S3, pois isso evita o upload de arquivos do console do Amplify e fornece maior confiabilidade para implantações manuais. A integração do Amplify com o Amazon S3 simplifica esse processo. Para obter mais informações, consulte Implantar um site estático para o Amplify a partir de um bucket do Amazon S3.

Preciso atualizar a Node.js versão do meu aplicativo

O Amplify não oferece mais suporte a aplicativos que usam Node.js as versões 14, 16 e 18. O comportamento depende do seu tipo de aplicativo:

- Aplicativos SSR: ocorrerão falhas de compilação ao usar versões obsoletas Node.js . Você não poderá implantar atualizações até fazer o upgrade para Node.js 20 ou mais tarde. Os tempos de execução suportados incluem Node.js 20, 22 e 24.
- Non-SSR Aplicativos: podem continuar usando Node.js versões obsoletas se você as instalar manualmente por meio de buildspec ou atualizações de pacotes ao vivo.

Os aplicativos que já estão implantados continuarão em execução, independentemente da Node.js versão.

Se você estiver usando a imagem de compilação do Amazon Linux 2023, a Node.js versão 20 é suportada por padrão. A imagem AL2023 suporta Node.js 22 e 24, com uma Node.js versão padrão de 22.

O Amazon Linux 2 (AL2) não oferece suporte automático à Node.js versão 20 ou posterior. Se, atualmente, você estiver usando o AL2, recomendamos mudar para o AL2023. É possível alterar a imagem de compilação no console do Amplify. Você também pode usar uma imagem de compilação personalizada que ofereça suporte à Node.js versão especificada.

Antes de atualizar, recomendamos que você teste a aplicação em uma nova ramificação para verificar se ela funciona corretamente.

Opções de atualização

Console do Amplify

Você pode usar o recurso de atualizações de pacotes ao vivo no console do Amplify para especificar a versão do Node.js a ser usada. Para instruções, consulte Uso de versões específicas de pacotes e dependências na imagem de compilação.

Imagem de compilação personalizada

Se você estiver usando uma imagem de compilação personalizada e o NVM estiver instalado em sua imagem, será possível adicionar o `nvm install 20` ao seu Dockerfile. Para saber mais sobre os requisitos e as instruções de configuração de uma imagem de compilação personalizada, consulte Personalização da imagem de compilação.

Configurações de compilação

Você pode especificar a Node.js versão a ser usada nas configurações de `amplify.yml` compilação do seu aplicativo adicionando o `nvm use` comando à seção de comandos do

PreBuild. Para obter instruções sobre como atualizar as configurações de compilação de uma aplicação, consulte Definição das configurações de compilação de uma aplicação do Amplify.

O exemplo a seguir demonstra como personalizar as configurações de compilação para definir a Node.js versão padrão como Node.js 20 e atualizar para a Node.js versão 24 em uma ramificação de teste chamada`node-24`.

```
frontend:
  phases:
    preBuild:
      commands:
        - nvm use 20
        - if [ "${AWS_BRANCH}" = "node-24" ]; then nvm use 24; fi
```

> **Warning**
>
> Esteja ciente de que os comandos do `preBuild` são executados após as atualizações ativas do pacote. A Node.js versão especificada pelo `nvm use` comando substituirá a Node.js versão definida pelas atualizações dinâmicas do pacote.

Solução de problemas da imagem de compilação do Amazon Linux 2023

As informações a seguir podem ajudá-lo a solucionar problemas com a imagem de compilação do Amazon Linux 2023 (AL2023).

Tópicos

- Quero executar as funções do Amplify com o runtime do Python
- Quero executar comandos que exijam privilégios de superusuário ou root

Quero executar as funções do Amplify com o runtime do Python

O Amplify Hosting agora usa a imagem de compilação do Amazon Linux 2023 por padrão quando você implanta uma nova aplicação. O AL2023 vem pré-instalado com as versões 3.8, 3.9, 3.10 e 3.11 do Python.

Para compatibilidade com versões anteriores da imagem do Amazon Linux 2, a imagem de compilação do AL2023 tem links simbólicos para versões mais antigas do Python pré-instaladas.

Por padrão, o Python versão 3.10 é usado globalmente. Para criar suas funções usando uma versão específica do Python, execute os comandos a seguir no arquivo de especificação de compilação da sua aplicação.

```
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
```

Quero executar comandos que exijam privilégios de superusuário ou root

Se você estiver usando a imagem de compilação do Amazon Linux 2023 e receber um erro ao executar comandos do sistema que exijam privilégios de superusuário ou root, será necessário executar esses comandos usando o comando `sudo` do Linux. Por exemplo, se você receber um erro ao executar `yum install -y gcc`, use `sudo yum install -y gcc`.

A imagem de compilação do Amazon Linux 2 usou o usuário root, mas a imagem AL2023 do Amplify executa seu código com um usuário personalizado `amplify`. O Amplify concede a esse usuário privilégios para executar comandos usando o comando `sudo` do Linux. É uma prática recomendada usar `sudo` para comandos que exigja privilégios de superusuário.

Solução de problemas de compilação

Se você encontrar problemas ao criar ou compilar uma aplicação do Amplify, consulte os tópicos desta seção para obter ajuda.

Tópicos

As novas confirmações no meu repositório não estão acionando as compilações do Amplify

Se novas confirmações no seu repositório Git não estiverem acionando as compilações do Amplify, verifique se o webhook ainda está presente no seu repositório. Se estiver presente, verifique o histórico das solicitações de webhook para ver se há alguma falha. O Amplify tem um limite de carga útil de 256 KB para webhooks de entrada. Se você enviar uma confirmação para o seu repositório com um grande número de arquivos alterados, poderá exceder esse limite e fazer com que as compilações não sejam acionadas.

O nome do meu repositório não está listado no console do Amplify ao criar uma nova aplicação

Ao criar uma nova aplicação no console do Amplify, é possível escolher entre os repositórios disponíveis da sua organização na página Adicionar repositório e ramificação. Seu repositório de destino pode não ser exibido na lista se não tiver sido atualizado recentemente. Isso pode ocorrer se sua organização tiver um grande número de repositórios. Para resolver esse problema, envie uma confirmação para o repositório e atualize a lista de repositórios no console. Isso deve fazer com que o repositório seja exibido.

Minha compilação falha com o erro **`Não é possível encontrar o módulo aws-exports`** (somente aplicativos de primeira geração)

Se a sua aplicação não conseguir encontrar o arquivo `aws-exports.js` durante uma compilação, o erro a seguir será retornado.

```
TS2307: Cannot find module 'aws-exports'
```

A interface de linha de comandos (CLI) do Amplify gera o arquivo `aws-exports.js` durante a compilação de backend. Para resolver esse erro, é necessário criar um arquivo `aws-exports.js` para uso na compilação. Adicione o código a seguir à especificação de compilação para criar o arquivo:

```
backend:
  phases:
    build:
      commands:
        - "# Execute Amplify CLI with the helper script"
        - amplifyPush --simple
```

Para ver um exemplo completo das configurações de especificação de compilação de uma aplicação do Amplify, consulte Sintaxe de referência do YAML de especificação de compilação.

Quero ignorar um tempo limite de compilação

O tempo limite de compilação padrão é de 30 minutos. É possível substituir o tempo limite de compilação padrão usando a variável de ambiente `_BUILD_TIMEOUT`. O tempo limite mínimo de compilação é de 5 minutos O tempo limite máximo de compilação é de 120 minutos

Para obter instruções sobre como definir uma variável de ambiente para uma aplicação no console do Amplify, consulte Configurar variáveis de ambiente.

Solucionar problemas de domínios personalizados

Se você encontrar problemas ao conectar um domínio personalizado a uma aplicação do Amplify, consulte os tópicos a seguir nesta seção para obter ajuda.

Se você não encontrar uma solução para seu problema aqui, entre em contato com o Suporte. Para obter mais informações, consulte Criação de um caso de suporte no Guia do usuário do AWS Support .

Tópicos

- Preciso verificar se meu CNAME é resolvido
- Meu domínio hospedado com terceiros está paralisado no estado de Verificação pendente
- Meu domínio hospedado com o Amazon Route 53 está paralisado no estado de verificação pendente

- Minha aplicação com subdomínios de vários níveis está presa no estado Verificação pendente
- Meu provedor de DNS não oferece suporte a registros A com nomes de domínio totalmente qualificados
- Eu recebo um CNAMEAlreadyExistsException erro
- Eu recebo um erro de verificação adicional necessária
- Eu recebo um erro 404 na URL CloudFront
- Recebo erros de certificado SSL ou HTTPS ao visitar meu domínio
- Componentes de caminho sem suporte em redirecionamentos de domínio
- Eu recebo um erro 400 por associação de domínio entre contas

Preciso verificar se meu CNAME é resolvido

1. Depois de atualizar seus registros DNS com seu provedor de domínio terceirizado, você pode usar uma ferramenta como dig ou um site gratuito https://www.whatsmydns.net/para verificar se o registro CNAME está sendo resolvido corretamente. A captura de tela a seguir demonstra como usar whatsmydns.net para verificar seu registro CNAME para o domínio www.example.com.

2. Escolha Pesquisar e whatsmydns.net exibirá os resultados do seu CNAME. A captura de tela a seguir é um exemplo de uma lista de resultados que verifica se o CNAME foi resolvido corretamente para uma URL cloudfront.net.

Meu domínio hospedado com terceiros está paralisado no estado de Verificação pendente

1. Se seu domínio personalizado estiver preso no estado Verificação pendente, verifique se seus registros CNAME estão sendo resolvidos. Consulte o tópico anterior de solução de problemas, Como faço para verificar se meu CNAME está resolvido, para obter instruções sobre como realizar essa tarefa.
2. Se seus CNAME registros não estiverem sendo resolvidos, confirme se a entrada CNAME existe nas configurações de DNS com seu provedor de domínio.

   > **Important**
   >
   > É importante atualizar seus registros CNAME assim que você criar seu domínio personalizado. Depois que o aplicativo é criado no console do Amplify, o registro CNAME é verificado em intervalos de alguns minutos para determinar se ocorre solução. Se não ocorrer depois de uma hora, a verificação será feita em intervalos de algumas horas, o que poderá causar um atraso na disponibilidade do domínio. Se você adicionou ou atualizou seus registros CNAME algumas horas depois de criar sua aplicação, essa é a causa mais provável da aplicação ficar presa no estado de verificação pendente.

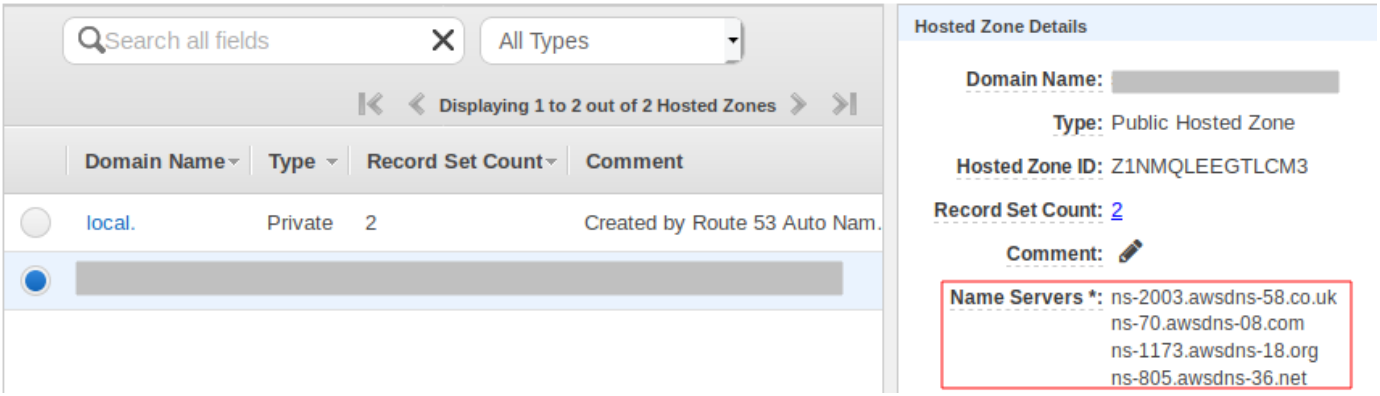
3. Se você verificou que o registro CNAME existe, pode haver um problema com seu provedor de DNS. É possível entrar em contato com o provedor de DNS para diagnosticar por que o de verificação de DNS CNAME não está sendo resolvido ou é possível migrar o DNS para Route53. Para obter mais informações, consulte Tornar o Amazon Route 53 o serviço de DNS para um domínio existente.

Meu domínio hospedado com o Amazon Route 53 está paralisado no estado de verificação pendente

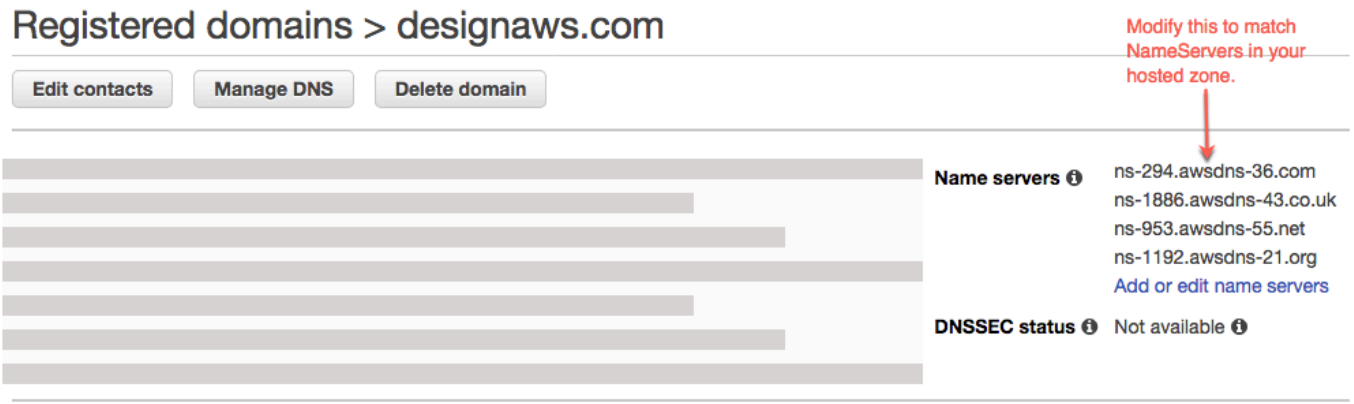
Se você transferiu seu domínio para Amazon Route 53, é possível que o domínio tenha nomes de servidores diferentes dos emitidos pelo Amplify quando o aplicativo foi criado. Execute as seguintes etapas para diagnosticar a causa do erro.

1. Faça login no console do Amazon Route 53
2. No painel de navegação, escolha Zonas hospedadas e escolha o nome do domínio ao qual você está conectando.

3. Registre os valores do servidor de nomes na seção Detalhes da zona hospedada. Você precisa destes dois para concluir a próxima etapa. A captura de tela a seguir do console do Route 53 exibe a localização dos valores do servidor de nomes no canto inferior direito.



4. No painel de navegação, escolha Domínios registrados. Verifique se os servidores de nomes exibidos na seção Domínios registrados correspondem aos valores do servidor de nomes que você registrou na etapa anterior na seção Detalhes da Zona Hospedada. Se eles não corresponderem, edite os valores do servidor de nomes para que correspondam aos valores em sua Zona Hospedada. A captura de tela a seguir do console do Route 53 exibe a localização dos valores do servidor de nomes no lado direito.



5. Se isso não resolver o problema, entre em contato com o Suporte. Para obter mais informações, consulte Criação de um caso de suporte no Guia do usuário do AWS Support .

Minha aplicação com subdomínios de vários níveis está presa no estado Verificação pendente

Se uma aplicação com subdomínios de vários níveis ficar presa no estado Verificação pendente ao se conectar a um provedor de DNS terceirizado, pode haver um problema com o formato dos seus registros de DNS. Alguns provedores de DNS adicionam automaticamente os sufixos de domínio de

segundo nível (SLD) e domínio de primeiro nível (TLD) aos seus registros. Se você também estiver especificando o domínio no formato que inclui o SLD e o TLD, isso pode causar um problema na verificação do domínio.

Ao conectar um domínio, primeiro tente especificar o nome do domínio usando o formato completo fornecido pelo Amplify, por exemplo, `_hash.docs.backend.example.com`. Se a configuração de SSL ficar presa no estado de Verificação pendente, tente remover o TLD e o SLD dos registros. Por exemplo, se o formato completo for `_hash.docs.backend.example.com`, especifique `_hash.docs.backend`. Aguarde de 15 a 30 minutos para permitir que os registros se propaguem. Em seguida, use uma ferramenta como o MX Toolbox para verificar se o processo de verificação está funcionando.

Meu provedor de DNS não oferece suporte a registros A com nomes de domínio totalmente qualificados

Alguns provedores de DNS não oferecem apoio a registros A com um nome de domínio totalmente qualificado (FQDN), como `example.cloudfront.net`. Por exemplo, os A records do Cloudflare só podem escrever endereços IPv4 e não oferecem suporte a FQDNs. Para contornar essa limitação, recomendamos usar registros CNAME em vez de A records em sua configuração de DNS.

Como exemplo, a configuração de DNS a seguir usa A record.

```
A     | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Altere-a para a configuração de DNS a seguir para usar somente registros CNAME.

```
CNAME | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Essa solução alternativa permite que você direcione adequadamente seu domínio apex (registro @) para serviços como CloudFront, evitando a IPv4-only limitação do sistema da A records Cloudflare.

Eu recebo um CNAMEAlreadyExistsException erro

Se você receber um CNAMEAlreadyExistsExceptionerro, isso significa que um dos nomes de host que você tentou conectar (um subdomínio ou domínio apex) já está implantado em outra distribuição da Amazon. CloudFront A origem do seu erro depende dos seus provedores atuais de hospedagem e DNS.

Um CNAME alias, como `example.com` ou só `sub.example.com` pode ser associado a uma única CloudFront distribuição por vez. Isso CNAMEAlreadyExistsExceptionindica que seu domínio já está associado a outra CloudFront distribuição, dentro da mesma Conta da AWS conta ou potencialmente em uma conta diferente. O domínio deve ser desassociado da CloudFront distribuição anterior antes que a nova distribuição criada pela Amplify Hosting funcione. Talvez seja necessário verificar mais de uma conta se você ou sua organização possuírem vários Conta da AWS s.

Execute as etapas a seguir para diagnosticar a causa do CNAMEAlreadyExistsExceptionerro.

1. Faça login no CloudFront console da Amazon e verifique se você não tem esse domínio implantado em outra distribuição. Um único CNAME registro pode ser anexado a uma CloudFront distribuição por vez.
2. Se você implantou anteriormente o domínio em uma CloudFront distribuição, deverá removê-lo.
   a. No painel de navegação esquerdo, escolha Distribuições.
   b. Selecione o nome da distribuição a ser editada.
   c. Escolha a guia Geral. Na seção Configurações, escolha Editar.
   d. Remova o nome de domínio do Nome de domínio alternativo (CNAME). Depois, escolha Salvar alterações.
3. Confirme se não existe nenhuma outra CloudFront distribuição que esteja usando esse domínio no atual Conta da AWS ou em outro Contas da AWS. Se isso não interromper nenhum serviço atualmente em execução, tente excluir e recriar a zona hospedada.
4. Verifique se este domínio está conectado a um aplicativo Amplify diferente de sua propriedade. Nesse caso, verifique se você não está tentando reutilizar um dos nomes de host. Se você estiver usando `www.example.com` para uma outra aplicação, não poderá usar `www.example.com` com a aplicação que você está conectando no momento. É possível usar outros subdomínios, como `blog.example.com`.
5. Se esse domínio foi conectado com sucesso a outro aplicativo e depois excluído na última hora, tente novamente após pelo menos uma hora. Se você ainda ver essa exceção após 6 horas, entre em contato Suporte. Para obter mais informações, consulte Criação de um caso de suporte no Guia do usuário do AWS Support .
6. Se você gerencia seu domínio por meio do Route 53, certifique-se de limpar qualquer zona hospedada CNAME ou ALIAS registros que apontem para a CloudFront distribuição antiga.
7. Depois de concluir as etapas anteriores, remova o domínio personalizado do Amplify Hosting e reinicie o fluxo de trabalho para conectar um domínio personalizado no console do Amplify.

Eu recebo um erro de verificação adicional necessária

Se você receber um erro de verificação adicional necessária, isso significa que o AWS Certificate Manager (ACM) precisa de informações adicionais para processar essa solicitação de certificado. Isso pode acontecer como uma medida de proteção contra fraudes, como quando o domínio se classifica dentro dos 1000 principais sites da Alexa. Para fornecer as informações necessárias, use a Central de suporte para entrar em contato com o Suporte. Se você não tem um plano de suporte, publique um novo thread no Fórum de discussão do ACM.

> **Note**
>
> Você não pode solicitar um certificado para nomes de Amazon-owned domínio como aqueles que terminam em amazonaws.com, cloudfront.net ou elasticbeanstalk.com.

Eu recebo um erro 404 na URL CloudFront

Para veicular tráfego, o Amplify Hosting aponta para um CloudFront URL por meio de um registro CNAME. No processo de conectar um aplicativo a um domínio personalizado, o console do Amplify exibe a CloudFront URL do aplicativo. No entanto, você não pode acessar seu aplicativo diretamente usando esse CloudFront URL. Retorna um erro 404. Seu aplicativo resolve apenas usando o URL do aplicativo Amplify (por exemplo, `https://main.d5udybEXAMPLE.amplifyapp.com`) ou seu domínio personalizado (por exemplo, `www.example.com`).

O Amplify precisa rotear as solicitações para a ramificação implantada correta e usar o nome do host para fazer isso. Por exemplo, é possível configurar o domínio `www.example.com` que aponta para a ramificação principal de um aplicativo, mas também configurar `dev.example.com` que aponta para a ramificação dev do mesmo aplicativo. Portanto, é necessário visitar seu aplicativo com base nos subdomínios configurados para que o Amplify possa rotear as solicitações adequadamente.

Recebo erros de certificado SSL ou HTTPS ao visitar meu domínio

Se você tiver registros DNS de Autorização de Autoridade Certificadora (CAA) configurados com seu provedor de DNS terceirizado, o AWS Certificate Manager (ACM) talvez não consiga atualizar ou reemitir certificados intermediários para seu certificado SSL de domínio personalizado. Para resolver isso, você precisa adicionar um registro CAA para confiar em pelo menos um dos domínios da autoridade de certificação da Amazon. O procedimento a seguir descreve as etapas que você precisa desempenhar.

Para adicionar um registro CAA para confiar em uma autoridade de certificação da Amazon

1. Configure um registro CAA com seu provedor de domínio para confiar em pelo menos um dos domínios da autoridade de certificação da Amazon. Para obter mais informações sobre como configurar o registro CAA, consulte Problemas de Autorização da Autoridade de Certificação (CAA) no Guia do Usuário do AWS Certificate Manager .
2. Use um dos métodos a seguir para atualizar seu certificado SSL:
   - Atualize manualmente usando o console do Amplify.

     > **Note**
     >
     > Esse método causará tempo de inatividade para seu domínio personalizado.

     a. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
     b. Escolha o aplicativo ao qual você deseja adicionar um registro CAA.
     c. No painel de navegação, escolha Configurações do aplicativo, Gerenciamento de domínio.
     d. Na página Gerenciamento de domínio, exclua o domínio personalizado.
     e. Conecte seu aplicativo ao domínio personalizado novamente. Esse processo emite um novo certificado SSL e seus certificados intermediários agora podem ser gerenciados pelo ACM.

        Para reconectar seu aplicativo ao seu domínio personalizado, use um dos procedimentos a seguir que corresponda ao provedor de domínio que você está usando.

        - Adição de um domínio personalizado gerenciado pelo Amazon Route 53.
        - Adição de um domínio personalizado gerenciado por um provedor de DNS terceirizado.
        - Atualizando registros DNS para um domínio gerenciado pelo GoDaddy.
   - Entre em contato Suporte para que seu certificado SSL seja reemitido.

Componentes de caminho sem suporte em redirecionamentos de domínio

Os redirecionamentos de domínio correspondem apenas à parte do nome do host. Não há suporte para os componentes de caminho em regras de origem baseadas em domínio (por exemplo, `"https://domain.com/path"`), e eles farão com que a regra seja ignorada sem erros. Para obter mais informações, consulte Referência de exemplo de redirecionamentos e regravações.

Eu recebo um erro 400 por associação de domínio entre contas

Ao iniciar uma DomainAssociation solicitação para um aplicativo Amplify com um domínio que já está ou foi anteriormente associado a diferentes aplicativos do Amplify em outras contas da AWS na mesma região, isso é considerado uma associação de domínio entre contas. Se você receber esse erro, isso significa que você está tentando uma associação de domínio entre contas, o que requer verificação manual. Se você quiser continuar com uma associação de domínio entre contas, entre em contato com o suporte da AWS para obter ajuda.

Solução de problemas de aplicações renderizadas do lado do servidor

Se você tiver problemas inesperados ao implantar um aplicativo SSR com a computação do Amplify Hosting, consulte os tópicos de solução de problemas a seguir. Se você não encontrar uma solução para seu problema aqui, consulte o guia de solução de problemas de computação na web do SSR no repositório Amplify Hosting Issues. GitHub

Tópicos

- Preciso de ajuda para usar um adaptador de framework
- As rotas da API Edge fazem com que minha Next.js compilação falhe
- On-Demand A regeneração estática incremental não está funcionando para meu aplicativo
- A saída de compilação da minha aplicação excede o tamanho máximo permitido
- Minha compilação falha com um erro de falta de memória
- O tamanho da resposta de HTTP da minha aplicação é muito grande
- Como faço para medir o tempo de inicialização da minha aplicação de computação localmente?
- Minha compilação falha com um erro de versão obsoleta Node.js

Preciso de ajuda para usar um adaptador de framework

Se você estiver tendo problemas para implantar uma aplicação de SSR que use um adaptador de framework, consulte Uso de adaptadores de código aberto para qualquer estrutura SSR.

As rotas da API Edge fazem com que minha Next.js compilação falhe

Atualmente, o Amplify não oferece suporte às rotas Next.js da API Edge. É necessário usar APIs e middleware de não borda ao hospedar sua aplicação com o Amplify.

On-Demand A regeneração estática incremental não está funcionando para meu aplicativo

A partir da versão 12.2.0, Next.js oferece suporte à Regeneração Estática Incremental (ISR) para limpar manualmente o Next.js cache de uma página específica. No entanto, o Amplify atualmente não oferece suporte On-Demand a ISR. Se seu aplicativo estiver usando a revalidação Next.js sob demanda, esse recurso não funcionará quando você implantar seu aplicativo no Amplify.

A saída de compilação da minha aplicação excede o tamanho máximo permitido

Atualmente, o tamanho máximo de saída de compilação com suporte no Amplify para aplicações SSR é de 220 MB. Se você receber uma mensagem de erro informando que o tamanho da saída de compilação da sua aplicação excede o tamanho máximo permitido, tome medidas para reduzi-la.

Para reduzir o tamanho da saída de compilação de uma aplicação, é possível inspecionar os artefatos de criação da aplicação e identificar quaisquer dependências grandes a serem atualizadas ou removidas. Primeiro, baixe os artefatos de compilação para o computador local. Em seguida, verifique o tamanho dos diretórios. Por exemplo, o `node_modules` diretório pode conter binários como `@swc` e `@esbuild` que são referenciados pelos arquivos de tempo de execução Next.js do servidor. Como esses binários não são necessários no runtime, é possível excluí-los após a compilação.

Use as instruções a seguir para baixar a saída de compilação de um aplicativo e inspecionar o tamanho dos diretórios usando a (AWS Command Line Interface CLI).

Para baixar e inspecionar a saída de compilação de um Next.js aplicativo

1. Abra uma janela de terminal e execute o comando a seguir. Altere o ID da aplicação, o nome da ramificação e o ID do trabalho para suas próprias informações. Para a ID do trabalho, use o número da compilação com falha que você está investigando.

```
aws amplify get-job --app-id abcd1234 --branch-name main --job-id 2
```

2. Na saída do terminal, localize o URL dos artefatos pré-assinados na seção `job`, `steps`, `stepName: "BUILD"`. O URL é destacado em vermelho no exemplo a seguir.

```
"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. Copie e cole o URL na janela do navegador. Um arquivo `artifacts.zip` é baixado no computador. Esse é o resultado da compilação.
4. Execute o comando de uso de disco du para inspecionar o tamanho dos diretórios. O comando de exemplo a seguir retorna o tamanho dos diretórios `compute` e `static`.

```
du -csh compute static
```

Veja a seguir um exemplo da saída com informações de tamanho para os diretórios `compute` e `static`.

```
 29M    compute
3.8M    static
 33M    total
```

5. Abra o diretório `compute` e localize a pasta `node_modules`. Revise suas dependências em busca de arquivos que você possa atualizar ou remover para diminuir o tamanho da pasta.
6. Se a sua aplicação incluir binários que não sejam necessários no runtime, exclua-os após a compilação adicionando os comandos a seguir à seção de compilação do arquivo `amplify.yml` da sua aplicação.

```
- 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
```

Veja a seguir um exemplo da seção de comandos de compilação de um arquivo `amplify.yml` com esses comandos adicionados após a execução de uma compilação de produção.

```
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
```

Minha compilação falha com um erro de falta de memória

Next.js permite armazenar artefatos de compilação em cache para melhorar o desempenho em compilações subsequentes. Além disso, o AWS CodeBuild contêiner do Amplify compacta e carrega esse cache no Amazon S3, em seu nome, para melhorar o desempenho da compilação subsequente. Isso pode fazer com que sua compilação falhe com um erro de falta de memória.

Execute as ações a seguir para evitar que seu aplicativo exceda o limite de memória durante a fase de compilação. Primeiro, remova `.next/cache/**/*` da seção cache.paths das suas configurações da compilação. Em seguida, remova a variável de ambiente `NODE_OPTIONS` do seu arquivo de configurações da compilação. Em vez disso, defina a variável de ambiente `NODE_OPTIONS` no console do Amplify para definir o limite máximo de memória do nó. Para mais

informações sobre como configurar as variáveis de ambiente usando o console do Amplify, consulte Configurar variáveis de ambiente.

Depois de fazer essas alterações, tente sua versão novamente. Se for bem-sucedido, adicione `.next/cache/**/*` novamente à seção cache.paths do seu arquivo de configurações da compilação.

Para obter mais informações sobre a configuração do Next.js cache para melhorar o desempenho da compilação, consulte a AWS CodeBuild no Next.js site.

O tamanho da resposta de HTTP da minha aplicação é muito grande

Atualmente, o tamanho máximo de resposta que o Amplify suporta para aplicativos Next.js 12 e posteriores usando a plataforma Web Compute é de 5,72 MB. As respostas acima desse limite retornam erros 504 sem conteúdo para os clientes.

Como faço para medir o tempo de inicialização da minha aplicação de computação localmente?

Use as instruções a seguir para determinar o tempo de initialization/start atividade local do seu aplicativo de computação de Next.js 12 anos ou posterior. É possível comparar o desempenho da sua aplicação localmente em relação a no Amplify Hosting e usar os resultados para melhorar o desempenho da sua aplicação.

Para medir o tempo de inicialização de um aplicativo de Next.js computação localmente

1. Abra o arquivo `next.config.js` da aplicação e defina a opção `output` como `standalone` da seguinte forma.

```
** @type {import('next').NextConfig} */
const nextConfig = {
  // Other options
  output: "standalone",
};

module.exports = nextConfig;
```

2. Abra uma janela de terminal e execute o comando a seguir para compilar a aplicação.

```
next build
```

3. Execute o comando a seguir para copiar a pasta `.next/static` para `.next/standalone/.next/static`.

```
cp -r .next/static .next/standalone/.next/static
```

4. Execute o comando a seguir para copiar a pasta `public` para `.next/standalone/public`.

```
cp -r public .next/standalone/public
```

5. Execute o comando a seguir para iniciar o Next.js servidor.

```
node .next/standalone/server.js
```

6. Observe quanto tempo leva entre a execução do comando na etapa 5 e a inicialização do servidor. Quando o servidor estiver escutando em uma porta, ele deverá imprimir a mensagem a seguir.

```
Listening on port 3000
```

7. Observe quanto tempo leva para qualquer outro módulo ser carregado após a inicialização do servidor na etapa 6. Por exemplo, bibliotecas como `bugsnag` levam de 10 a 12 segundos para serem carregadas. Depois de carregado, ele exibirá a mensagem de confirmação `[bugsnag] loaded`.
8. Adicione as durações da etapa 6 e da etapa 7 juntas. Esse resultado é o tempo de atividade local initialization/start do seu aplicativo Compute.

Minha compilação falha com um erro de versão obsoleta Node.js

Problema: A compilação do seu aplicativo SSR falha com um erro de Node.js versão não suportada.

```
# NODE.JS VERSION NOT SUPPORTED
================================================================================
Your application uses Node.js v18.x.x, which is no longer supported.
AWS Amplify Console has ended support for Node.js 14, Node.js 16 and Node.js 18.

To deploy your application, please upgrade to Node.js 20 or later.

For detailed migration guidelines, visit: https://docs.aws.amazon.com/amplify/latest/
userguide/troubleshooting-general.html#update-node-version
```

```
================================================================================
```

Causa: Seu aplicativo SSR foi criado usando uma Node.js versão obsoleta (14.x, 16.x ou 18.x). Desde 15 de setembro de 2025, o Amplify bloqueou a implantação de aplicação de SSR que usem essas versões obsoletas durante o processo de compilação.

Atualize seu ambiente de compilação para usar Node.js 20 ou posterior. Para obter instruções detalhadas, consulte Preciso atualizar a Node.js versão do meu aplicativo.

Solução de problemas de redirecionamentos e reescritas

Se você encontrar problemas ao configurar redirecionamentos e reescritas para uma aplicação do Amplify, consulte os tópicos desta seção para obter ajuda.

Tópicos

- O acesso é negado para determinadas rotas, mesmo com a regra de redirecionamento do SPA.
- Quero configurar um proxy reverso para uma API

O acesso é negado para determinadas rotas, mesmo com a regra de redirecionamento do SPA.

Se você estiver recebendo um erro de acesso negado para determinadas rotas com uma regra de redirecionamento de SPA, o `baseDirectory` pode não estar definido corretamente nas configurações de compilação da aplicação. Por exemplo, se o frontend da sua aplicação for criado no diretório `build`, suas configurações de compilação também deverão apontar para o diretório `build`. O exemplo de especificação de compilação a seguir demonstra essa configuração.

```
frontend:
  artifacts:
    baseDirectory: build
    files:
        - "**/*"
```

Para ver um exemplo completo das configurações de especificação de compilação de uma aplicação do Amplify, consulte Sintaxe de referência do YAML de especificação de compilação

Quero configurar um proxy reverso para uma API

É possível usar o JSON a seguir para configurar um proxy reverso para um endpoint dinâmico.

```
[
  {
    "source": "/documents/<*>",
    "target": "https://otherdomain/resource/<*>",
    "status": "200",
    "condition": null
  }
]
```

Para ver um exemplo básico de criação de um proxy reverso da sua aplicação do Amplify para uma API de terceiros, consulte Regravação de proxy reverso.

Solução de problemas de armazenamento em cache

Se você encontrar problemas de armazenamento em cache para uma aplicação do Amplify, consulte os tópicos desta seção para obter ajuda.

Tópicos

Quero reduzir o tamanho do cache de uma aplicação

Se você estiver usando o cache, talvez esteja armazenando em cache arquivos intermediários que não sejam limpos entre as compilações. O armazenamento em cache desses arquivos usados com pouca frequência aumentará o tamanho do cache. Para evitar isso, é possível excluir pastas específicas de armazenamento em cache usando a diretiva `!` na seção `cache` da especificação de compilação da sua aplicação.

O exemplo de configurações de compilação a seguir demonstra como usar a diretiva `!` para especificar uma pasta que você não deseja armazenar em cache.

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

```
    - "!node_modules/path/not/to/cache"
```

Quando você armazena em cache a pasta `node_modules`, `node_modules/.cache` é omitida por padrão.

Para ver um exemplo completo das configurações de especificação de compilação de uma aplicação do Amplify, consulte Sintaxe de referência do YAML de especificação de compilação

Quero desativar a leitura do cache de uma aplicação

Se você quiser desativar a leitura do cache de uma aplicação, remova a seção de cache da especificação de compilação da sua aplicação.

Configurando o acesso do Amplify aos repositórios GitHub

O Amplify agora usa o recurso GitHub Apps para autorizar o acesso somente de leitura do Amplify aos repositórios. GitHub Com o GitHub aplicativo Amplify, as permissões são mais refinadas, permitindo que você conceda ao Amplify acesso somente aos repositórios que você especificar. Para saber mais sobre GitHub aplicativos, consulte Sobre GitHub aplicativos no GitHub site.

Quando você conecta um novo aplicativo armazenado em um GitHub repositório, por padrão, o Amplify usa GitHub o aplicativo para acessar o repositório. No entanto, os aplicativos existentes do Amplify que você conectou anteriormente a partir de GitHub repositórios usam para OAuth acesso. CI/CD continuarão funcionando para esses aplicativos, mas é altamente recomendável que você os migre para usar o novo aplicativo GitHub Amplify.

Ao implantar um novo aplicativo ou migrar um aplicativo existente usando o console do Amplify, você é automaticamente direcionado para o local de instalação do aplicativo GitHub Amplify. Para acessar manualmente a página inicial de instalação do aplicativo, abra um navegador da Web e navegue até o aplicativo por região. Use o formato`https://github.com/apps/aws-amplify-`*`REGION`*, *`REGION`* substituindo-o pela região em que você implantará seu aplicativo Amplify. Por exemplo, para instalar o GitHub aplicativo Amplify na região Oeste dos EUA (Oregon), navegue até -2. https://github.com/apps/ aws-amplify-us-west

Tópicos

- Instalando e autorizando o aplicativo GitHub Amplify para uma nova implantação
- Migração de um OAuth aplicativo existente para o aplicativo Amplify GitHub
- Configurando o GitHub aplicativo Amplify para implantações de CloudFormation CLI e SDK

- Configurando visualizações na web com o aplicativo Amplify GitHub

Instalando e autorizando o aplicativo GitHub Amplify para uma nova implantação

Ao implantar um novo aplicativo no Amplify a partir do código existente em um GitHub repositório, use as instruções a seguir para instalar e autorizar o aplicativo. GitHub

Para instalar e autorizar o aplicativo Amplify GitHub

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Na página Todos os aplicativos, escolha Novo aplicativo e, em seguida, Hospedar aplicativo web.
3. Na página Começar com o Amplify Hosting, escolha e, em seguida GitHub, escolha Continuar.
4. Se for a primeira vez que você conecta um GitHub repositório, uma nova página é aberta em seu navegador GitHub em.com, solicitando permissão para autorizar AWS Amplify em sua conta. GitHub Escolha Authorize.
5. Em seguida, você deve instalar o GitHub aplicativo Amplify em sua GitHub conta. Uma página é aberta no GitHub.com solicitando permissão para instalar e autorizar AWS Amplify na sua conta. GitHub
6. Selecione a GitHub conta na qual você deseja instalar o aplicativo Amplify GitHub .
7. Execute um destes procedimentos:
   - Para aplicar a instalação a todos os repositórios, escolha Todos os repositórios.
   - Para limitar a instalação aos repositórios específicos que você selecionar, escolha Somente selecionar repositórios. Certifique-se de incluir o repositório do aplicativo que você está migrando nos repositórios selecionados.
8. Escolha Instalar e autorizar.
9. Você é redirecionado para a página Adicionar ramificação do repositório do seu aplicativo no console do Amplify.
10. Na lista Repositórios atualizados recentemente, selecione o nome do repositório a ser conectado.
11. Na lista Ramificação, selecione o nome da ramificação do repositório a ser conectada.
12. Escolha Próximo.
13. Na página Definir configurações de segurança, escolha Próximo.

14. Na página Revisar, escolha Salvar e implantar.

Migração de um OAuth aplicativo existente para o aplicativo Amplify GitHub

Os aplicativos existentes do Amplify que você conectou anteriormente a partir de GitHub repositórios usam OAuth para acesso ao repositório. É altamente recomendável que você migre esses aplicativos para usar o aplicativo GitHub Amplify.

Use as instruções a seguir para migrar um aplicativo e excluir o OAuth webhook correspondente em sua GitHub conta. Observe que o procedimento de migração varia dependendo se o aplicativo GitHub Amplify já está instalado. Depois de migrar seu primeiro aplicativo, instalar e autorizar o GitHub aplicativo, você só precisa atualizar as permissões do repositório para migrações subsequentes do aplicativo.

Para migrar um aplicativo OAuth para o GitHub aplicativo

1. Faça login no Console de gerenciamento da AWS e abra o console do Amplify.
2. Escolha o aplicativo que você deseja migrar.
3. Na página de informações do aplicativo, localize a mensagem azul Migrar para nosso GitHub aplicativo e escolha Iniciar migração.
4. Na página Instalar e autorizar GitHub aplicativo, escolha Configurar GitHub aplicativo.
5. Uma nova página é aberta em seu navegador GitHub em.com, solicitando permissão para autorizar AWS Amplify em sua GitHub conta. Escolha Authorize.
6. Selecione a GitHub conta na qual você deseja instalar o aplicativo Amplify GitHub .
7. Execute um destes procedimentos:
   - Para aplicar a instalação a todos os repositórios, escolha Todos os repositórios.
   - Para limitar a instalação aos repositórios específicos que você selecionar, escolha Somente selecionar repositórios. Certifique-se de incluir o repositório do aplicativo que você está migrando nos repositórios selecionados.
8. Escolha Instalar e autorizar.
9. Você é redirecionado para a página Instalar e autorizar o GitHub aplicativo para seu aplicativo no console do Amplify. Se a GitHub autorização for bem-sucedida, você verá uma mensagem de sucesso. Escolha Próximo.
10. Na página Instalação completa, escolha Instalação completa. Essa etapa exclui seu webhook existente, cria um novo e conclui a migração.

Configurando o GitHub aplicativo Amplify para implantações de CloudFormation CLI e SDK

Os aplicativos existentes do Amplify que você conectou anteriormente a partir de GitHub repositórios usam OAuth para acesso ao repositório. Isso pode incluir aplicativos que você implantou usando a Interface de Linha de Comando (CLI) CloudFormation do Amplify ou o. SDKs É altamente recomendável que você migre esses aplicativos para usar o novo aplicativo GitHub Amplify. A migração deve ser realizada no console do Amplify no Console de gerenciamento da AWS. Para instruções, consulte Migração de um OAuth aplicativo existente para o aplicativo Amplify GitHub.

Você pode usar CloudFormation a CLI do Amplify e a para SDKs implantar um novo aplicativo Amplify que usa o aplicativo para acesso ao repositório. GitHub Esse processo requer que você primeiro instale o GitHub aplicativo Amplify em sua GitHub conta. Em seguida, você precisará gerar um token de acesso pessoal em sua GitHub conta. Por fim, implante o aplicativo e especifique o token de acesso pessoal.

Instale o GitHub aplicativo Amplify em sua conta

1. Abra um navegador da web e navegue até o local de instalação do GitHub aplicativo Amplify na AWS região em que você implantará seu aplicativo.

   Use o formato`https://github.com/apps/aws-amplify-REGION/installations/new`, *REGION* substituindo-o por sua própria entrada. Por exemplo, se você estiver instalando seu aplicativo na região Oeste dos EUA (Oregon), especifique`https://github.com/apps/aws-amplify-us-west-2/installations/new`.
2. Selecione a GitHub conta na qual você deseja instalar o aplicativo Amplify GitHub .
3. Execute um destes procedimentos:
   - Para aplicar a instalação a todos os repositórios, escolha Todos os repositórios.
   - Para limitar a instalação aos repositórios específicos que você selecionar, escolha Somente selecionar repositórios. Certifique-se de incluir o repositório do aplicativo que você está migrando nos repositórios selecionados.
4. Escolha Instalar.

Gere um token de acesso pessoal em sua GitHub conta

1. Faça login na sua GitHub conta.
2. No canto superior direito, localize sua foto do perfil e escolha Configurações no menu.

3. No menu de navegação à esquerda, escolha Configurações do desenvolvedor.
4. Na página GitHub Aplicativos, no menu de navegação à esquerda, escolha Tokens de acesso pessoal.
5. Na página Tokens de acesso pessoal, escolha Gerar novo token.
6. Na página Novo token de acesso pessoal, em Nota, insira um nome descritivo para o token.
7. Na seção Selecionar escopos, selecione admin:repo_hook.
8. Escolha Gerar token.
9. Copie e salve o token de acesso pessoal. Você precisará fornecê-lo ao implantar um aplicativo Amplify com a CLI ou o. CloudFormation SDKs

Depois que o GitHub aplicativo Amplify for instalado em sua GitHub conta e você tiver gerado um token de acesso pessoal, você poderá implantar um novo aplicativo com a CLI do Amplify ou a. CloudFormation SDKs Use o campo `accessToken` para especificar o token de acesso pessoal criado no procedimento anterior. Para obter mais informações, consulte CreateAppa referência da API Amplify e AWS::Amplify::Appo Guia do AWS CloudFormation usuário.

O comando CLI a seguir implanta um novo aplicativo Amplify que usa o aplicativo para acesso ao GitHub repositório. Substitua *myapp-using-githubapphttps://github.com/Myaccount/react-app*,, e *MY_TOKEN* por suas próprias informações.

```
aws amplify create-app --name myapp-using-githubapp --repository https://github.com/
Myaccount/react-app --access-token MY_TOKEN
```

Configurando visualizações na web com o aplicativo Amplify GitHub

Uma visualização prévia na web implanta cada pull request (PR) feita em seu GitHub repositório em uma URL de visualização exclusiva. As visualizações prévias agora usam o aplicativo GitHub Amplify para GitHub acessar seu repositório. Para obter instruções sobre como instalar e autorizar o GitHub aplicativo para visualizações na web, consulte. Habilitação de pré-visualizações na Web para solicitações de pull

AWS Amplify Referência de hospedagem

Use os tópicos desta seção para encontrar material de referência detalhado para AWS Amplify.

Tópicos

AWS CloudFormation apoio

Use AWS CloudFormation modelos para provisionar recursos do Amplify, permitindo implantações de aplicativos web reproduzíveis e confiáveis. AWS CloudFormation fornece uma linguagem comum para você descrever e provisionar todos os recursos de infraestrutura em seu ambiente de nuvem e simplifica a implantação em várias and/or regiões de AWS contas com apenas alguns cliques.

Para o Amplify Hosting, consulte a documentação do Amplify. CloudFormation Para o Amplify Studio, consulte a documentação do Amplify UI Builder. CloudFormation

AWS Command Line Interface apoio

Use o AWS Command Line Interface para criar aplicativos Amplify programaticamente a partir da linha de comando. Para obter informações, consulte a documentação do AWS CLI .

Suporte para marcação de recursos

Você pode usar o AWS Command Line Interface para marcar os recursos do Amplify. Para obter mais informações, consulte a documentação Etiquetas de recurso do AWS CLI .

Amplify Hosting API

Essa referência do fornece as descrições das ações e dos tipos de dados para a Amplify Hosting API. Para obter mais informações, consulte a documentação de referência da API do Amplify.

Histórico do documento para AWS Amplify

A tabela a seguir descreve as mudanças importantes na documentação desde a última versão do AWS Amplify.

- Última atualização da documentação: 4 de maio de 2026

Alteração	Descrição	Data
Suporte adicional para Node.js 24	Foram adicionados Node.js 24 à lista de tempos de execução compatíveis com aplicativos SSR e atualizamos a especificação de implantação do SSR.	4 de maio de 2026
Atualizações para versões Node.js compatíveis	Informações atualizadas sobre as versões Node.js suportadas em Implantação de aplicativos renderizados do lado do servidor com o Amplify Hosting.	8 de setembro de 2025
Atualização do capítulo Definições de configurações de compilação	O capítulo Gerenciamento da configuração de compilação de uma aplicação do Amplify foi atualizado para descrever o novo recurso de tipo de instância de compilação configurável, que permite escolher um tipo de instância que forneça à sua aplicação os recursos de CPU, memória e espaço em disco necessários.	28 de maio de 2025

Alteração	Descrição	Data
Atualização do capítulo Firewall	O Suporte de firewall para sites hospedados pelo Amplify capítulo foi atualizado para descrever a disponibilidade geral (GA) da integração do Amplify com AWS WAF, incluindo a funcionalidade do GA e a estrutura de preços.	26 de março de 2025
Novo capítulo Proteção contra distorções	Foi adicionado o capítulo Proteção contra distorções para implantações do Amplify para descrever o recurso de proteção contra distorções que elimina problemas de distorção de versão entre clientes e servidores nas aplicações da Web do Amplify.	10 de março de 2025
Atualização do capítulo Webhooks	Foi adicionado o tópico Webhooks unificados para repositórios Git para descrever o recurso de webhooks unificados que usa um webhook abrangente para todas as aplicações do Amplify associadas a um único repositório Git.	10 de março de 2025

Alteração	Descrição	Data
Novo tópico Adicionando uma função SSR Compute para permitir o acesso aos AWS recursos	Foi adicionado o tópico Adicionar uma função SSR Compute para permitir o acesso aos recursos AWS para descrever como criar e associar um perfil do SSR Compute do Amplify a uma aplicação para dar ao serviço Amplify Compute acesso a recursos da AWS .	17 de fevereiro de 2025
Novo capítulo Usando AWS WAF para proteger seus aplicativos do Amplify	Foi adicionado o Suporte de firewall para sites hospedados pelo Amplify capítulo para descrever a integração do Amplify com AWS WAF (em versão prévia), que permite proteger seus aplicativos da web com uma lista de controle de acesso à web (ACL da web).	18 de dezembro de 2024
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	14 de novembro de 2024
Suporte atualizado do Amplify para o tópico Next.js	O Amplifique o suporte para Next.js tópico foi atualizado para descrever o suporte do Amplify para a Next.js versão 15.	6 de novembro de 2024

Alteração	Descrição	Data
Nova implantação de um site estático para o Amplify a partir de um bucket do Amazon S3	Foi adicionado o capítulo Implantar um site estático para o Amplify a partir de um bucket do Amazon S3 para descrever a nova integração do Amplify Hosting com o Amazon S3, que permite hospedar conteúdo estático de sites armazenados no S3 com apenas alguns cliques.	16 de outubro de 2024
Novo capítulo sobre gerenciamento de configuração de cache	Foi adicionado o capítulo Gerenciar a configuração de cache de uma aplicação para descrever o comportamento padrão do armazenamento em cache do Amplify e como ele aplica políticas de cache gerenciado ao conteúdo.	13 de agosto de 2024
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	18 de julho de 2024
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	31 de maio de 2024

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	17 de abril de 2024
O capítulo de Noções básicas foi atualizado	Atualizou o Noções básicas da implantação de uma aplicação no Amplify Hosting capítulo para usar um aplicativo de Next.js exemplo no tutorial.	12 de abril de 2024
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	5 de abril de 2024
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	4 de abril de 2024
Novo capítulo de solução de problemas	Foi adicionado o capítulo Solução de problemas do Amplify Hosting para descrever como corrigir problemas que você encontra com aplicações implantadas no Amplify Hosting.	2 de abril de 2024

Alteração	Descrição	Data
Novo suporte para SSL/TLS certificados personalizados	O Usando SSL/TLS certificados tópico foi adicionado ao Conexão de um domínio personalizado capítulo para descrever o suporte do Amplify para SSL/TLS certificados personalizados ao conectar um aplicativo a um domínio personalizado.	20 de fevereiro de 2024
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	2 de janeiro de 2024
Nova compatibilidade com frameworks de SSR	O Implantação de aplicações renderizadas do lado do servidor com o Amplify Hosting tópico foi atualizado para descrever o suporte do Amplify para qualquer estrutura Javascript-based SSR com um adaptador de código aberto.	19 de novembro de 2023
Nova compatibilidade com o lançamento do recurso de otimização de imagem	Adição do tópico Otimização de imagem para aplicações de SSR para descrever a compatibilidade integrada com otimização de imagem para aplicações renderizadas no lado do servidor.	19 de novembro de 2023

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	17 de novembro de 2023
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	6 de novembro de 2023
Novo tópico sobre subdomínios curinga	Foi adicionado o tópico Configuração de subdomínios curinga para descrever o suporte para subdomínios curinga em domínios personalizados.	6 de novembro de 2023
Novas políticas gerenciadas	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever a nova política AmplifyBackendDeployFullAccess AWS gerenciada do Amplify.	8 de outubro de 2023

Alteração	Descrição	Data
Novo suporte para lançamento de atributos de estruturas monorepo	O tópico Definição de configurações de compilação monorepo foi atualizado para descrever o suporte à implantação de aplicativos em monorepos criados usando npm workspace, pnpm workspace, Yarn workspace, Nx e Turborepo.	19 de junho de 2023
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	1º de junho de 2023
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	24 de fevereiro de 2023
Capítulo de renderização no lado do servidor atualizado	O Implantação de aplicações renderizadas do lado do servidor com o Amplify Hosting capítulo foi atualizado para descrever as mudanças recentes no suporte do Amplify para Next.js as versões 12 e 13.	17 de novembro de 2022

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	30 de agosto de 2022
Tópico de políticas gerenciadas atualizado	O tópico Compilação de um backend para uma aplicação foi atualizado para descrever como implantar um backend usando o Amplify Studio.	23 de agosto de 2022
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	27 de abril de 2022
Tópico de políticas gerenciadas atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	17 de abril de 2022
Lançamento de novo recurso do GitHub aplicativo	Foi adicionado o Configurando o acesso do Amplify aos repositórios GitHub tópico para descrever o novo GitHub aplicativo para autorizar o acesso do Amplify ao GitHub seu repositório.	5 de abril de 2022

Alteração	Descrição	Data
Lançamento do novo atributo Amplify Studio	O tópico Bem-vindo à AWS Amplify hospedagem foi atualizado para descrever as atualizações do Amplify Studio que fornecem um designer visual para criar component es de interface do usuário que podem ser conectados aos seus dados de backend.	2 de dezembro de 2021
Tópico de políticas gerenciad as atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever as mudanças recentes nas políticas gerenciadas da AWS para o Amplify para oferecer suporte ao Amplify Studio.	2 de dezembro de 2021
Tópico de políticas gerenciad as atualizado	O tópico AWS políticas gerenciadas para AWS Amplify foi atualizado para descrever alterações recentes nas políticas gerenciadas pela AWS para o Amplify.	8 de novembro de 2021
Tópico de políticas gerenciad as atualizado	O AWS políticas gerenciad as para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	27 de setembro de 2021

Alteração	Descrição	Data
Novo tópico de políticas gerenciadas	Foi adicionado o AWS políticas gerenciadas para AWS Amplify tópico para descrever as políticas AWS gerenciadas do Amplify e as mudanças recentes nessas políticas.	28 de julho de 2021
Capítulo de renderização do lado do servidor atualizado	O Implantação de aplicações renderizadas do lado do servidor com o Amplify Hosting capítulo foi atualizado para descrever o novo suporte para a Next.js versão 10. x. x e Next.js versão 11.	22 de julho de 2021
Capítulo de configuração de configurações da compilação atualizado	Foi adicionado o tópico Definição de configurações de compilação monorepo para descrever como definir as configurações da compilação e a nova variável de ambiente `AMPLIFY_MONOREPO_APP_ROOT` ao implantar um aplicativo monorepo com o Amplify.	20 de julho de 2021

Alteração	Descrição	Data
Capítulo atualizado sobre implantações de ramificação de atributo	Foi adicionado o tópico Geração automática em tempo de compilação da configuração do Amplify (somente aplicações Gen 1) para descrever como gerar automaticamente o arquivo `aws-exports.js` no momento da compilação. Foi adicionado o tópico Compilações condicionais de backend (somente aplicações Gen 1) para descrever como habilitar compilações condicionais de backend. Foi adicionado o tópico Use backends do Amplify em todas as aplicações (somente aplicações Gen 1) para descrever como reutilizar backends existentes ao criar um novo aplicativo, conectar uma nova ramificação a um aplicativo existente ou atualizar um frontend existente para apontar para um ambiente de backend diferente.	30 de junho de 2021

Alteração	Descrição	Data
Capítulo Segurança atualizado	Foi adicionado o tópico Proteção de dados no Amplify para descrever como aplicar o modelo de responsabilidade compartilhada e como o Amplify usa criptografia para proteger seus dados em repouso e em trânsito.	3 de junho de 2021
Novo suporte para o lançamento do atributo SSR	Foi adicionado o Implantação de aplicações renderizadas do lado do servidor com o Amplify Hosting capítulo para descrever o suporte do Amplify para aplicativos web que usam renderização do lado do servidor (SSR) e são criados com. Next.js	18 de maio de 2021
Novo capítulo de segurança	Foi adicionado o capítulo Segurança no Amplify para descrever como aplicar o modelo de responsabilidade compartilhada ao usar o Amplify e como configurar o Amplify para atender aos seus objetivos de segurança e conformidade.	26 de março de 2021

Alteração	Descrição	Data
Tópico de compilações personalizadas atualizado	Atualizou o tópico Imagens de compilação personalizadas e atualizações de pacotes ao vivo para descrever como configurar uma imagem de compilação personalizada hospedada no Amazon Elastic Container Registry Public.	12 de março de 2021
Tópico de monitoramento atualizado	Atualizou o tópico Monitoramento para descrever como acessar dados de CloudWatch métricas da Amazon e definir alarmes.	2 de fevereiro de 2021
Novo tópico de CloudTrail registro	Foi adicionado o tópico Logging Amplify API usando o AWS CloudTrail tópico para descrever como AWS CloudTrail captura e registra todas as ações da API para a AWS Amplify Console API Reference e a AWS Amplify Admin UI API Reference.	2 de fevereiro de 2021

Alteração	Descrição	Data
Lançamento do novo atributo Admin UI	O tópico Bem-vindo à AWS Amplify hospedagem foi atualizado para descrever a nova interface de usuário do administrador, que fornece uma interface visual para desenvolvedores frontend web e móveis criarem e gerenciar em backends de aplicativos fora do Console de gerenciamento da AWS.	1º de dezembro de 2020
Lançamento do novo atributo do modo de desempenho	O tópico Gerenciando o desempenho do aplicativo foi atualizado para descrever como ativar o modo de desempenho para otimizar o desempenho da hospedagem.	4 de novembro de 2020
Atualizado o tópico de cabeçalhos personalizados	O tópico Cabeçalhos personalizados foi atualizado para descrever como definir cabeçalhos personalizados para um aplicativo Amplify usando o console ou editando um arquivo YML.	28 de outubro de 2020

Alteração	Descrição	Data
Lançamento do novo atributo de subdomínios automáticos	Foi adicionado o tópico Configurar subdomínios automáticos para um domínio personalizado do Route 53 para descrever como usar implantações de ramificações de atributos baseadas em padrões para um aplicativo conectado a um domínio personalizado do Amazon Route 53. Foi adicionado o tópico Acesso à visualização prévia da Web com subdomínios para descrever como configurar visualizações da Web a partir de solicitações pull para serem acessíveis com subdomínios.	20 de junho de 2020
Novo tópico de notificações	Foi adicionado o tópico Notificações para descrever como configurar notificações por e-mail para um aplicativo Amplify para alertar as partes interessadas ou membros da equipe quando uma compilação for bem-sucedida ou falhar.	20 de junho de 2020

Alteração	Descrição	Data
Atualizou o tópico de domínios personalizados	Atualizou o Conexão de um domínio personalizado tópico para melhorar os procedimentos de adição de domínios personalizados no Amazon Route 53 e no Google Domains. GoDaddy Essa atualização também inclui novas informações de solução de problemas para configurar domínios personalizados.	12 de maio de 2020
AWS Amplify soltar	Esta versão apresenta o Amplify.	26 de novembro de 2018

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.