Guia do usuário

AWS Amplify Hospedagem

AWS Amplify Hospedagem: Guia do usuário

Table of Contents

Bem-vindo à AWS Amplify hospedagem

O Amplify Hosting fornece um fluxo de trabalho baseado em Git para hospedar aplicativos web sem servidor completos 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.

Estruturas compatíveis

O Amplify Hosting suporta muitas estruturas SSR comuns, estruturas de aplicativos de página única (SPA) e geradores de sites estáticos, incluindo os seguintes.

Estruturas SSR

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

Estruturas SPA

- React
- Angular
- Vue.js
- Iônico
- Brasa

Geradores de sites estáticos

- Onze
- Gatsby
- Hugo
- Jekyll

- VuePress

Atributos do Amplify Hosting

Filiais de destaque

Gerencie ambientes de preparação e produção para o front-end e back-end conectando novas ramificações.

Domínios personalizados

Conecte seu aplicativo a um domínio personalizado.

Visualizações prévias de pull request

Visualize as alterações durante as análises de código.

nd-to-end Teste E

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

Agências protegidas por senha

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

Redireciona e reescreve

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 as janelas de manutenção, garantindo que seu aplicativo web seja atualizado somente após a conclusão de toda a implantação. Isso elimina cenários em que não é possível fazer upload dos arquivos corretamente.

Comece a usar o Amplify Hosting

Para começar a usar o Amplify Hosting, veja o Conceitos básicos do Amplify Hosting tutorial. 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.

Crie um back-end

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 criar e conectar um back-end ao seu aplicativo, consulte Criar e conectar back-end nos documentos do Amplify.

Se você estiver procurando a documentação para criar back-ends para um aplicativo de primeira geração, usando a CLI e o Amplify Studio, consulte Criar e conectar back-end nos documentos do Amplify de primeira geração.

Preços do Amplify Hosting

AWS Amplify faz parte do Nível gratuito da AWS. Você pode começar gratuitamente e pagar conforme o uso quando exceder os limites do nível gratuito. Para obter informações sobre as taxas de hospedagem do Amplify, consulte AWS Amplify Preços.

Conceitos básicos do Amplify Hosting

Para ajudar você a entender como o Amplify Hosting funciona, este tutorial explica como criar e implantar um aplicativo Next.js a partir de um repositório Git.

Tópicos

Pré-requisitos

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

Inscreva-se para um Conta da AWS

Se você ainda não é AWS cliente, precisa criar um Conta da AWS seguindo as instruções on-line. A inscrição permite que você acesse o Amplify e outros AWS serviços que você pode usar com seu aplicativo.

Cria uma aplicação

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

Crie um repositório Git

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

Etapa 1: Conectar um repositório Git

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

Para conectar um aplicativo em um repositório Git

1. Abra o console do Amplify.
2. Se você estiver implantando seu primeiro aplicativo na região atual, por padrão, você começará na página de AWS Amplifyserviço.

   Escolha Criar novo aplicativo na parte superior da página.
3. Na página Comece a criar 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, consulteConfigurar o acesso do Amplify aos repositórios do 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 para se conectar.
   c. Escolha Próximo.

Etapa 2: confirme as configurações de compilação

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

Para confirmar as configurações de criação de um aplicativo

1. Na página de configurações do aplicativo, localize a seção Configurações de compilação.

Verifique se o comando Frontend build e o diretório de saída Build estão corretos. Para este aplicativo de exemplo Next.js, o diretório de saída do Build está definido como `.next`.

2. O procedimento para adicionar uma função de serviço varia dependendo se você deseja criar uma nova função ou usar uma existente.
   - Para criar uma nova função:
     - Escolha Criar e usar um novo perfil de serviço.
   - Para usar uma função existente:
     a. Escolha Usar uma função existente.
     b. Na lista de funções de serviço, selecione a função a ser usada.
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 um aplicativo

1. Na página de revisão, confirme se os detalhes do repositório e as configurações do aplicativo estão corretos.
2. Escolha Salvar e implantar. Sua criação de front-end normalmente leva de 1 a 2 minutos, mas pode variar de acordo com o tamanho do aplicativo.
3. Quando a implantação estiver concluída, você poderá visualizar seu aplicativo usando o link para o domínio `amplifyapp.com` padrão.

> **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 página Set-Cookie na Mozilla Developer Network.

Etapa 4: (opcional) limpar os recursos

Se você não precisar mais do aplicativo que implantou para o tutorial, poderá excluí-lo. Esta etapa ajuda a garantir que você não será cobrado pelos recursos que não está utilizando.

Para excluir um aplicativo

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

Adicione recursos ao seu aplicativo

Agora que você tem um aplicativo implantado no Amplify, você pode explorar alguns dos seguintes recursos que estão disponíveis para seu aplicativo hospedado.

Variáveis de ambiente

Os aplicativos geralmente precisam de informações de configuração em tempo de execução. Essas configurações podem ser detalhes da conexão do banco de dados, chaves de API ou parâmetros. As variáveis de ambiente fornecem uma maneira de expor essas configurações no momento da construção. Para obter mais informações, consulte Variáveis de ambiente.

Domínios personalizados

Neste tutorial, o Amplify hospeda seu aplicativo para você no `amplifyapp.com` domínio padrão 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 Configurar domínios personalizados.

Visualizações de solicitação pull

As visualizações prévias de pull request na Web oferecem às equipes uma maneira de visualizar as alterações das pull requests (PRs) antes de mesclar o código em uma ramificação de produção ou integração. Para obter mais informações, consulte Pré-visualizações da Web para pull requests.

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.

Implante aplicativos renderizados 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 aplicações criadas usando o framework Next.js, e você não precisa realizar nenhuma configuração manual no AWS Management Console. O Amplify também é compatível com qualquer framework de SSR baseado em Javascript com um adaptador de compilação de código aberto que transforma a saída de compilação de uma aplicação na estrutura de diretórios esperada pelo Amplify Hosting.

Para saber como o Amplify oferece suporte ao SSR, consulte os seguintes tópicos.

Tópicos

O que é renderização do lado do servidor

O Amplify é compatível com a implantação e a hospedagem de aplicações Web estáticas criadas com frameworks de aplicação de página única (SPA), como React, e aplicações criadas com um gerador de site estático (SSG), como o Gatsby. Os aplicativos web estáticos consistem em uma combinação de arquivos, como HTML, CSS e JavaScript arquivos, que são armazenados em uma rede de distribuição de conteúdo (CDN). Quando um navegador cliente faz uma solicitação ao site, o servidor retorna uma página para o cliente com uma resposta HTTP e o navegador do cliente interpreta o conteúdo e o exibe para o usuário.

O Amplify também é compatível com aplicações Web com renderização no lado do servidor (SSR). Quando um cliente envia uma solicitação para uma página SSR, o HTML da página é criado no servidor em cada solicitação. O SSR permite que um desenvolvedor personalize um site por

solicitação e por usuário. Além disso, o SSR pode melhorar o desempenho e a otimização de mecanismos de pesquisa (SEO) de um site.

Compatibilidade do Amplify com frameworks de SSR

O Amplify Hosting suporta qualquer estrutura SSR JavaScript baseada em um pacote de implantação que está em conformidade com a saída de compilação que o Amplify espera. O Amplify Hosting fornece uma especificação de implantação que padroniza a estrutura de arquivos e diretórios para a saída da compilação de uma aplicação para qualquer framework de SSR.

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 suas estruturas específicas. 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 ou um adaptador de framework, poderá desenvolver sua própria solução para gerar um pacote de implantação que esteja em conformidade com a estrutura de diretórios esperada pelo Amplify Hosting.

O Amplify Hosting suporta os seguintes tipos primitivos: ativos estáticos, computação, otimização de imagem e regras de roteamento. Você pode aproveitar esses tipos primitivos para implantar aplicativos com funcionalidades mais avançadas. Para obter informações detalhadas sobre cada tipo primitivo, consulteAmplifique o suporte ao tipo primitivo SSR.

É possível escolher entre os seguintes cenários para começar a implantar uma aplicação de SSR no Amplify.

Implantar uma aplicação Next.js

> O Amplify é compatível com aplicações criadas usando Next.js sem exigir um adaptador ou configuração manual no console. Para ter mais informações, consulte Amplifique o suporte para Next.js.

Implantar uma aplicação que usa um adaptador de framework

> Você pode consultar qualquer adaptador de framework de código aberto disponível para implantar sua aplicação de SSR no Amplify Hosting. Para ter mais informações, consulte Como usar um adaptador de framework.

Há um adaptador disponível para o framework Nuxt. Para obter mais informações sobre como usar esse adaptador, consulte a Documentação do Nuxt.

Compilar 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 ter mais informações, consulte Como implantar um servidor Express usando o manifesto de implantação.

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 ter mais informações, consulte Como usar a 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.

Implantar uma aplicação de SSR no Amplify

É possível aplicar as instruções neste tópico 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. Não será necessário usar nenhum adaptador se você estiver implantando uma aplicação Next.js.

Se estiver implantando uma aplicação de SSR que use um adaptador de framework, primeiro será necessário instalar e configurar o adaptador. Para obter instruções, consulte Como usar um adaptador de framework.

Para implantar uma aplicação de SSR no Amplify Hosting

1. Faça login AWS Management Console e abra o console do Amplify.
2. Na página Todos os aplicativos, escolha Criar novo aplicativo.
3. Na página Comece a criar 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 para se conectar.

c. Escolha Próximo.

5. Na página de configurações do aplicativo, o Amplify detecta automaticamente os aplicativos SSR Next.js.

   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 Habilitar registros do aplicativo SSR na seção Implantação de renderização no lado do servidor (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 uma função de serviço varia dependendo se você deseja criar uma nova função ou usar uma existente.

   - Para criar uma nova função:
     - Escolha Criar e usar um novo perfil de serviço.
   - Para usar uma função existente:
     a. Escolha Usar uma função existente.
     b. Na lista de funções de serviço, selecione a função a ser usada.

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

Como usar um adaptador de framework

Você pode instalar e 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, você poderá implantá-la no Amplify Hosting. Para obter instruções detalhadas, consulte Implantar uma aplicação de SSR no Amplify

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

Use a especificação de implantação do Amplify para configurar a saída da compilação para um framework de SSR que você deseja integrar ao Amplify Hosting. Se você for o criador de um framework, poderá usar a especificação de implantação para entender como estruturar a saída da compilação que o Amplify espera. Se você não estiver usando um framework, poderá desenvolver sua própria solução para gerar a saída da compilação que o Amplify espera.

Especificação de implantação do Amplify Hosting

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. Uma estrutura pode gerar essa estrutura de diretórios esperada como saída de seu comando de construção, permitindo que a estrutura aproveite os tipos primitivos de serviço do Amplify Hosting. A Amplify Hosting entende a estrutura do pacote de implantação e o implanta adequadamente.

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
```

Amplifique o suporte ao tipo primitivo SSR

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

Ativos estáticos

Fornece aos frameworks a capacidade de hospedar arquivos estáticos.

Computação

Fornece aos framework a capacidade de executar um servidor HTTP Node.js 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.

O diretório .amplify-hosting/static

Você deve 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 são servidos por meio do tipo 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`.

O diretório .amplify-hosting/compute

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 tipo primitivo de computação 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 módulo Node.js e iniciar um servidor HTTP que escute na porta 3000.
- Você pode 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 compactado de cada pacote de recursos computacionais não pode ultrapassar 220 MB. Por exemplo, o subdiretório `.amplify/compute/default` não pode ultrapassar 220 MB quando compactado.

O arquivo .amplify-hosting/deploy-manifest.json

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 `routes` atributo permite que as estruturas aproveitem o tipo primitivo das 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 Usar redirecionamentos.

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	Descrição
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. 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:

Chave	Tipo	Obrigatório	Descrição
			• `*` (corresponde a 0 ou mais caracteres) • O padrão `/*` é chamado de padrão abrangente e corresponderá a todas as solicitaç ões recebidas.
target	Destino	Sim	Um objeto que define o destino para o qual rotear a solicitação correspondente. Se houver a especific ação de uma rota `Compute`, deverá haver um `ComputeRe source` correspon dente. Se houver a especific ação de uma rota `ImageOpti mization` , também deverá haver um `imageSettings` especificado.

Chave	Tipo	Obrigatório	Descrição
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 específica. 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	Descrição
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 tipos primitivos	Uma string que especifica o nome do subdiretório no pacote de implantaç ão que contém o código executável do tipo primitivo. Válido e necessário somente para o tipo 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 cabeçalho Cache-Control a ser aplicado à resposta. Válido somente para os tipos

Chave	Tipo	Obrigatório	Descrição
			estático e ImageOpti mization primitivo. O valor especificado é substituído por cabeçalhos personali zados. Para obter mais informações sobre os cabeçalho s personalizados do Amplify Hosting, consulte Cabeçalhos personalizados. **Note** Esse cabeçalho Cache-Con trol é aplicado somente a respostas bem-suced idas 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

Encaminha as solicitações para o tipo primitivo de ativos estáticos.

Computação

Encaminha as solicitações para o tipo primitivo de computação.

ImageOptimization

Encaminha as solicitações para o tipo 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 = 'nodejs16.x' | 'nodejs18.x' | 'nodejs20.x';
```

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

Chave	Tipo	Obrigatório	Descrição
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 `nodejs16.x` , `nodejs18.x` e `nodejs20.x` .
entrypoint	String	Sim	Especifica o nome do arquivo inicial com base no qual o código será executado para o recurso computacional especificado. O arquivo deve existir dentro do subdiretório que representa um

Chave	Tipo	Obrigatório	Descrição
			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": "nodejs16.x",
      "entrypoint": "index.js",
    }
  ]
```

Como usar o atributo imageSettings

O `imageSettings` atributo permite que as estruturas personalizem o comportamento do tipo primitivo de otimização de imagem, que fornece otimização sob demanda de imagens em tempo de execução.

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	Descrição
sizes	Number[]	Sim	Uma matriz de larguras de imagem compatíveis.
domains	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.
minimumCacheTTL	Número	Sim	A duração do cache em segundos para as imagens otimizadas.

Chave	Tipo	Obrigatório	Descrição
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?: 'http' | 'https';
  hostname: string;
  port?: string;
  pathname?: string;
}
```

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

Chave	Tipo	Obrigatório	Descrição
protocolo	String	Não	O protocolo do padrão remoto permitido. Os valores válidos são `http` ou `https`.
hostname	String	Sim	O nome de host do padrão remoto permitido. Você pode especificar um caractere literal ou curinga. Um “*” único corresponde a um único subdomínio. Um “**” duplo correspon

Chave	Tipo	Obrigatório	Descrição
			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	Descrição
name	String	Sim	O nome do framework .
versão	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:

- Tipo primitivo de ativos estáticos — Os arquivos estão contidos no `.amplify-hosting/static` diretório.
- Tipo primitivo de computação — Os arquivos estão contidos no `.amplify-hosting/compute/default` diretório.

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 estruturas SSR comoNuxt.js, o alvo da regra abrangente deve ser o tipo primitivo de computação. 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 `/_nuxt/*` caminho pode ser direcionado por uma regra de roteamento específica que roteia solicitações para o tipo 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 da rota abrangente deve ser do tipo 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, você pode 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, você pode adicionar um destino alternativo a essa regra para garantir que, quando houver um erro 404 do tipo primitivo estático, a solicitação retorne ao tipo primitivo de computação.

```
{
    "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 Roteamento abrangente de fallback.

Roteamento abrangente de fallback

Para estruturas SSRNuxt.js, como, por exemplo, em que uma rota abrangente é especificada para o destino do tipo primitivo de computação, os autores da estrutura podem considerar a especificação de um substituto estático para a rota abrangente para resolver o problema de roteamento de pastas. `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 volta para o tipo primitivo estático e ainda termina com um código de status 404, mas a página 404 não é 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"
    }
}
```

Exemplos de rotas do Nuxt.js

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": "nodejs18.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": "nodejs18.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. Você pode 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": "nodejs18.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. Você pode 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 18
        - 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.

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, você pode 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.

No momento, você pode usar o componente Next.js Image para otimização sob demanda das imagens ou implementar um carregador personalizado de imagens. Se você estiver usando o 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 Como usar um carregador personalizado de imagens.

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 compatíveis com o Next.js, consulte a documentação de imagens do Next.js.

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 Especificação de implantação do Amplify Hosting.

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 compatíveis com 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	Descrição	Exemplo
url	String	Sim	Um caminho relativo ou URL absoluto para a imagem de origem. Para um URL remoto, há compatibilidade com os protocolos HTTP e HTTPS. O valor deve estar	`?url=https%3A%2F%2Fwww.example.com%2Fbuffalo.png`

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
			codificado no URL.	
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 especific ada, a imagem passará por ajuste de escala automático para corresponder à largura.	`?height=600`
fit	Valores de enumeraçã o: `cover`, `contain`, `fill`, `inside`, `outside`	Não	Como a imagem é redimensi onada para se ajustar à largura e à altura especificadas.	`?width=80 0&height= 600&fit=c over`
position	Valores de enumeração: `center`, `top`, `right`, `bottom`, `left`	Não	Uma posição a ser usada quando o ajuste for `cover` ou `contain`.	`?fit=cont ain&posit ion=centre`

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
trim	Número	Não	Apara pixels de todas as bordas que contenham valores semelhantes à cor de fundo especificada do pixel superior esquerdo.	`?trim=50`
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`

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
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`
format	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`

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
flip	Booleano	Não	Espelha a imagem verticalmente (de cima para baixo) no eixo x. Isso sempre ocorre antes da rotação, se houver.	`?flip`
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`

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
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 especific ado. Os valores válidos vão de 0,3 a 1.000.	`?blur=20`
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`

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
normalize	Booleano	Não	Melhora o contraste da imagem ampliando sua iluminação para englobar uma faixa dinâmica completa.	`?normalize`
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`

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
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.

Armazenamento em cache

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.

Compatibilidade com versão Node.js para aplicações Next.js

Quando o Amplify compila e implanta uma aplicação de computação Next.js, ele usa a versão de runtime do Node.js correspondente à versão principal usada para criar a aplicação Node.js.

Você pode 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 Atualizações de pacote ao vivo. 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.

Solução de problemas de implantações SSR

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

- Você está usando um adaptador de framework
- As rotas da API Edge fazem com que sua compilação do Next.js falhe
- A regeneração estática incremental sob demanda não está funcionando para seu aplicativo
- A saída de compilação do seu aplicativo excede o tamanho máximo permitido
- Sua compilação falha com um erro de falta de memória
- O tamanho da resposta HTTP é muito grande

Você está usando um adaptador de framework

Se você estiver tendo problemas para implantar uma aplicação de SSR que use um adaptador de framework, consulte Compatibilidade do Amplify com frameworks de SSR.

As rotas da API Edge fazem com que sua compilação do Next.js falhe

Atualmente, o Amplify não oferece suporte às rotas da API Edge do Next.js. Você deve usar APIs e middleware não periféricos ao hospedar seu aplicativo com o Amplify.

A regeneração estática incremental sob demanda não está funcionando para seu aplicativo

A partir da versão 12.2.0, o Next.js oferece suporte à Regeneração Estática Incremental (ISR) para limpar manualmente o cache do Next.js de uma página específica. No entanto, o Amplify atualmente não oferece suporte à ISR sob demanda. Se seu aplicativo estiver usando a revalidação sob demanda do Next.js, esse atributo não funcionará quando você implantar seu aplicativo no Amplify.

A saída de compilação do seu aplicativo excede o tamanho máximo permitido

Atualmente, o tamanho máximo de saída de compilação compatível com o Amplify para aplicativos SSR é 220 MB. Se você receber uma mensagem de erro informando que o tamanho da saída de compilação do seu aplicativo excede o tamanho máximo permitido, tome medidas para reduzi-lo.

Para reduzir o tamanho da saída de compilação de um aplicativo, você pode inspecionar os artefatos de criação do aplicativo e identificar quaisquer dependências grandes a serem atualizadas ou removidas. Primeiro, baixe os artefatos de construção para o seu 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 do servidor Next.js. Como esses binários não são necessários no tempo de execução, você pode 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 aplicativo Next.js

1. Abra uma janela de terminal e execute o comando a seguir. Altere o ID do aplicativo, o nome da filial 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 de saída 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 em uma janela do navegador. Um `artifacts.zip` arquivo é baixado para o seu computador local. Essa é a saída da sua compilação.

4. Execute o comando de uso do du disco para inspecionar o tamanho dos diretórios. O comando de exemplo a seguir retorna o tamanho dos `compute` `static` diretórios e.

```
du -csh compute static
```

Veja a seguir um exemplo da saída com informações de tamanho para os `static` diretórios `compute` e.

```
 29M    compute
3.8M    static
 33M    total
```

5. Abra o `compute` diretório e localize a `node_modules` pasta. Revise suas dependências em busca de arquivos que você possa atualizar ou remover para diminuir o tamanho da pasta.
6. Se seu aplicativo incluir binários que não são necessários no tempo de execução, exclua-os após a compilação adicionando os comandos a seguir à seção de compilação do `amplify.yml` arquivo do seu aplicativo.

```
- 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 `amplify.yml` arquivo 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
```

Sua compilação falha com um erro de falta de memória

O Next.js permite que você armazene 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 Definição de 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 cache do Next.js para melhorar o desempenho da compilação, consulte AWS CodeBuild no site do Next.js.

O tamanho da resposta HTTP é muito grande

Atualmente, o tamanho máximo de resposta que o Amplify suporta para aplicativos Next.js 12 ou posteriores usando a plataforma Web Compute é de 5,72 MB. As respostas acima desse limite retornam erros 504 sem conteúdo para os clientes.

Amplifique o suporte para Next.js

O Amplify oferece suporte à implantação e hospedagem de aplicativos web renderizados no lado do servidor (SSR) criados usando o Next.js. O Next.js é uma estrutura React para criar aplicativos web de pilha completa. Você pode implantar aplicativos criados com o Next.js 14 com recursos como otimização de imagem e middleware.

Os desenvolvedores podem usar o 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 o 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.

O Next.js fornece suporte analítico integrado para medir várias métricas de desempenho, como Tempo até o primeiro byte (TTFB) e Primeira pintura com conteúdo (FCP). Para obter mais informações sobre o Next.js, consulte Introdução no site do Next.js.

Suporte ao atributo Next.js

A computação do Amplify Hosting gerencia totalmente a renderização do lado do servidor (SSR) para aplicativos criados com as versões 12, 13 e 14 do Next.js. Se você implantou um aplicativo Next.js no Amplify antes do lançamento da computação do Amplify Hosting, seu aplicativo está usando o provedor SSR anterior do Amplify, o Classic (somente Next.js 11). A computação do Amplify Hosting não é compatível com aplicativos criados usando o Next.js versão 11 ou anterior. É altamente recomendável que você migre seus aplicativos Next.js 11 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.

Atributos compatíveis

- Páginas renderizadas do lado do servidor (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)
- Middleware
- Variáveis de ambiente
- Otimização de imagem
- Diretório de aplicativos Next.js 13

Atributos não compatíveis

- Rotas de API do Edge (o middleware Edge não é suportado)

- Regeneração estática incremental (ISR) sob demanda
- Detecção automática de localidade internacionalizada (i18n)
- Transmissão de Next.js
- Executando middleware em ativos estáticos e imagens otimizadas

Imagens do Next.js

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 de imagem do Next.js para redimensioná-lo e otimizá-lo em um formato Webp ou AVIF e, em seguida, fornecê-lo como um tamanho menor.

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

Preços dos aplicativos Next.js

Ao implantar seu aplicativo SSR Next.js 12 ou posterior, a computação do Amplify Hosting gerencia os recursos necessários para implantar o aplicativo SSR para você. Para obter informações sobre as cobranças de computação do Amplify Hosting, consulte Preços do AWS Amplify.

Implantando um aplicativo Next.js com o Amplify

Por padrão, o Amplify implanta novos aplicativos SSR usando o serviço de computação da Amplify Hosting com suporte para Next.js 12, 13 e 14. 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 (somente Next.js 11).

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

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

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

1. Faça login AWS Management Console e abra o console do Amplify.
2. Na página Todos os aplicativos, escolha Criar novo aplicativo.
3. Na página Comece a criar 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 para se conectar.
   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 uma função e a anexe ao seu aplicativo:
     - Escolha Criar e usar um novo perfil de serviço.
   - Para anexar uma função de serviço que você criou anteriormente:
     a. Escolha Usar uma função de serviço existente.
     b. Selecione a função a ser usada na lista.
6. Escolha Próximo.
7. Na página Revisar, escolha Salvar e implantar.

Configurações do arquivo Package.json

Quando você implanta um aplicativo Next.js, o Amplify inspeciona o script de compilação do aplicativo no `package.json` arquivo para detectar se o aplicativo é SSR ou SSG.

Veja a seguir um exemplo do script de compilação de um aplicativo SSR do Next.js. O script de compilação `"next build"` indica que o aplicativo é compatível com páginas SSG e SSR.

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

```
},
```

Veja a seguir um exemplo do script de compilação de um aplicativo SSG do Next.js. 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"
},
```

Configurações de compilação do Amplify

Depois de inspecionar o `package.json` arquivo do seu aplicativo para determinar se você está implantando um aplicativo SSG ou SSR, o Amplify verifica as configurações da compilação do aplicativo. É possível salvar as configurações da compilação no console do Amplify ou em um arquivo `amplify.yml` na raiz do seu repositório. Para ter mais informações, consulte Configurar as definições de compilação.

Se o Amplify detectar que você está implantando um aplicativo SSR Next.js e nenhum arquivo `amplify.yml` estiver presente, ele gerará uma especificação de compilação para o aplicativo e definirá `baseDirectory` como `.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, você deve 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 compilação são de um aplicativo Next.js 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/**/*
```

Se o Amplify detectar que você está implantando um aplicativo SSG, ele gera uma especificação de compilação para o aplicativo e define `baseDirectory` como `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.

Veja a seguir um exemplo das configurações da compilação de um aplicativo em que `baseDirectory` está definido como `out`. Isso indica que os artefatos de compilação são para um aplicativo Next.js 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/**/*
```

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

Quando você implanta um novo aplicativo Next.js, por padrão, o Amplify usa a versão mais recente compatível do Next.js. Atualmente, o provedor SSR de computação Amplify Hosting oferece suporte ao Next.js versão 14.

O console do Amplify detecta aplicativos em sua conta que foram implantados antes do lançamento do serviço de computação Amplify Hosting com suporte total para as versões 12, 13 e 14 do Next.js.

O console exibe um banner de informações identificando aplicativos com ramificações que são implantadas usando o provedor SSR anterior do 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.

Você deve migrar manualmente o aplicativo e todas as suas ramificações de produção ao mesmo tempo. Um aplicativo não pode conter as ramificações Classic (somente Next.js 11) e Next.js 12, 13 ou 14.

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 AWS Management Console e abra o console do Amplify.
2. Escolha o aplicativo Next.js 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 as versões 12, 13 ou 14 do 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 aplicativo Next.js, 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 o Next.js versão 11 é definido com o valor `WEB_DYNAMIC` da plataforma. Um aplicativo SSR 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, você deve 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 Next.js estático

É possível adicionar a funcionalidade SSR a um aplicativo Next.js estático (SSG) existente implantado com o Amplify. Antes de iniciar o processo de conversão do aplicativo SSG em SSR, atualize o aplicativo para usar as versões 12, 13 ou 14 do Next.js 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.

Atualize a plataforma

Há três valores válidos de plataforma. Um aplicativo SSG está configurado para o tipo de plataforma `WEB`. Um aplicativo SSR usando o Next.js versão 11 está configurado para o tipo de plataforma `WEB_DYNAMIC`. Para aplicativos implantados no 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
```

Adicione 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 AWS Management Console 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 aplicativo estático do Next.js ao qual você deseja adicionar um perfil 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.

Atualize as configurações da compilação

Antes de reimplantar seu aplicativo com a funcionalidade SSR, você deve 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, Configurar as definições de compilação.

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/**/*
```

Atualize o 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 compilação para `"next build"` indicar que o aplicativo Next.js 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 do servidor Next.js 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 ao Next.js, é possível modificar o arquivo de especificação de compilação do Amplify para defini-las nos arquivos de ambiente que o Next.js reconhece. Isso permite que o Amplify carregue essas variáveis de ambiente antes de compilar o aplicativo. 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 DB_HOST -e DB_USER -e DB_PASS >> .env.production
        - env | grep -e NEXT_PUBLIC_ >> .env.production
```

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

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

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

```
DB_HOST=localhost
DB_USER=myuser
DB_PASS=mypassword
```

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_SEARCH_KEY=asdfiojslf
NEXT_PUBLIC_SEARCH_ENDPOINT=https://search-url
```

Variáveis de ambiente SSR para monorepos

Se você estiver implantando um aplicativo SSR em um monorepo e quiser tornar variáveis de ambiente específicas acessíveis ao Next.js, deverá prefixar o `.env.production` arquivo com a raiz do aplicativo. O exemplo de especificação de compilação a seguir para um aplicativo Next.js 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 DB_HOST -e DB_USER -e DB_PASS >> apps/app/.env.production
            - env | grep -e NEXT_PUBLIC_ >> apps/app/.env.production
            - npx nx build app
      artifacts:
        baseDirectory: dist/apps/app/.next
        files:
          - '**/*'
      cache:
        paths:
          - node_modules/**/*
      buildPath: /
    appRoot: apps/app
```

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

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

Implantando um aplicativo Next.js 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. Observe que os aplicativos pnpm e Turborepo exigem configuração adicional. Para ter mais informações, consulte Configurações de compilação do Monorepo.

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

Amazon CloudWatch Logs para aplicativos SSR

Amplify envia informações sobre seu tempo de execução do Next.js 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 Adicionar um perfil de serviço.

Amplify: suporte ao Next.js 11

Se você implantou um aplicativo Next.js 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 Next.js 11). 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 aplicativos Next.js 11 para o provedor de SSR gerenciado por computação do Amplify Hosting. Para ter mais informações, consulte Migrando um aplicativo Next.js 11 para a computação do Amplify Hosting.

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

Atributos compatíveis

- Páginas renderizadas do lado do servidor (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
- Regeneração estática incremental (ISR) sob demanda
- Roteamento de domínio internacionalizado (i18n)
- Detecção automática de localidade internacionalizada (i18n)
- Middleware
- Middleware de borda
- Rotas de API do Edge

Preços dos aplicativos Next.js 11

Ao implantar seu aplicativo SSR Next.js 11, 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 aplicativos SSR Next.js 11

O Amplify requer permissões AWS Identity and Access Management (IAM) para implantar um aplicativo SSR. Sem as permissões mínimas exigidas, você receberá um erro ao tentar implantar seu aplicativo SSR. Para fornecer ao Amplify as permissões necessárias, você deve 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 Adicionar um perfil de serviço. 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:ListCertificates
acm:RequestCertificate
cloudfront:CreateCloudFrontOriginAccessIdentity
cloudfront:CreateDistribution
cloudfront:CreateInvalidation
cloudfront:GetDistribution
cloudfront:GetDistributionConfig
cloudfront:ListCloudFrontOriginAccessIdentities
cloudfront:ListDistributions
cloudfront:ListDistributionsByLambdaFunction
cloudfront:ListDistributionsByWebACLId
cloudfront:ListFieldLevelEncryptionConfigs
cloudfront:ListFieldLevelEncryptionProfiles
cloudfront:ListInvalidations
cloudfront:ListPublicKeys
cloudfront:ListStreamingDistributions
cloudfront:UpdateDistribution
cloudfront:TagResource
cloudfront:UntagResource
cloudfront:ListTagsForResource
cloudfront:DeleteDistribution
iam:AttachRolePolicy
iam:CreateRole
iam:CreateServiceLinkedRole
iam:GetRole
iam:PutRolePolicy
iam:PassRole
iam:UpdateAssumeRolePolicy
iam:DeleteRolePolicy
lambda:CreateFunction
lambda:EnableReplication
lambda:DeleteFunction
lambda:GetFunction
lambda:GetFunctionConfiguration
lambda:PublishVersion
lambda:UpdateFunctionCode
lambda:UpdateFunctionConfiguration
lambda:ListTags
lambda:TagResource
lambda:UntagResource
lambda:ListEventSourceMappings
lambda:CreateEventSourceMapping
route53:ChangeResourceRecordSets
route53:ListHostedZonesByName
```

```
route53:ListResourceRecordSets
s3:CreateBucket
s3:GetAccelerateConfiguration
s3:GetObject
s3:ListBucket
s3:PutAccelerateConfiguration
s3:PutBucketPolicy
s3:PutObject
s3:PutBucketTagging
s3:GetBucketTagging
sqs:CreateQueue
sqs:DeleteQueue
sqs:GetQueueAttributes
sqs:SetQueueAttributes
amplify:GetApp
amplify:GetBranch
amplify:UpdateApp
amplify:UpdateBranch
```

Solução de problemas de implantações do Next.js 11

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

- Regiões não compatíveis

Seu diretório de saída foi substituído

O diretório de saída de um aplicativo Next.js 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 Configurar as definições de compilação.

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/**/*
```

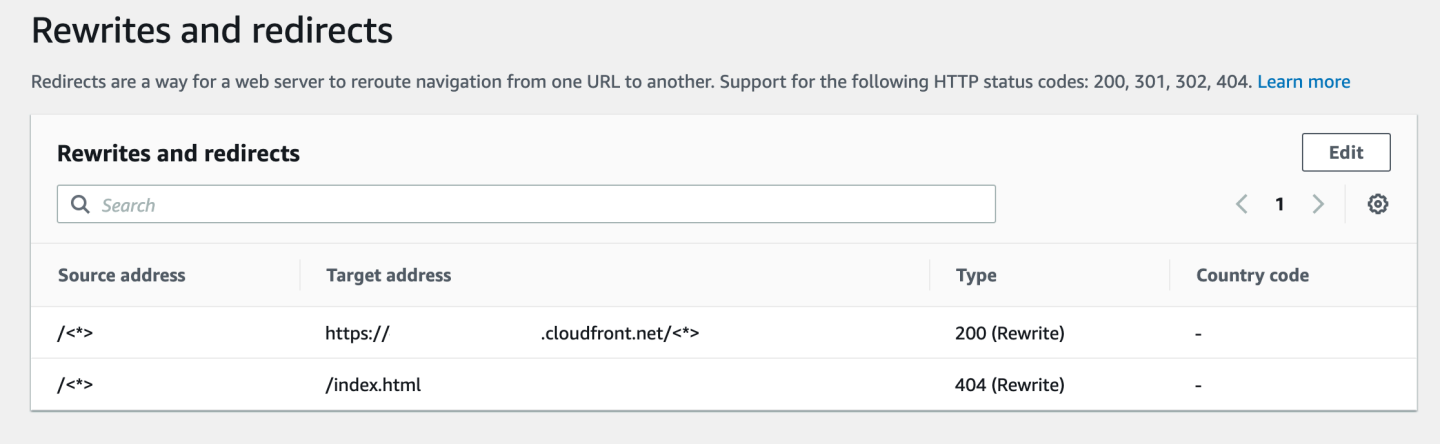
Você recebe um erro 404 após implantar seu site SSR

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, Seu diretório de saída foi substituído.

Seu 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.



Seu aplicativo é muito grande para ser implantado

O Amplify limita o tamanho de uma implantação de SSR a 50 MB. Se você tentar implantar um aplicativo SSR Next.js no Amplify e receber uma mensagem de erro `RequestEntityTooLargeException`, 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
      },
}
```

Sua compilação falha com um erro de falta de memória

O Next.js permite que você armazene 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 Definição de 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 cache do Next.js para melhorar o desempenho da compilação, consulte AWS CodeBuild no site do Next.js.

Seu aplicativo 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.

Seu aplicativo armazena arquivos estáticos em uma pasta com um caminho reservado

O 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 aplicativo Next.js 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, você deverá renomear a pasta `static` antes de implantar com o Amplify.

Seu aplicativo 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 variáveis de ambiente não são transferidas para as funções do Lambda

As variáveis de ambiente que você especifica no console do Amplify para um aplicativo SSR não são transferidas para as funções do aplicativo. AWS Lambda Consulte Tornando as variáveis de ambiente acessíveis aos runtimes do lado do servidor, para obter instruções detalhadas sobre como adicionar variáveis de ambiente que é possível referenciar a partir de suas funções do Lambda.

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

Quando você implanta um aplicativo Next.js, 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.

Seu aplicativo Next.js usa atributos não compatíveis

Os aplicativos implantados com o Amplify oferecem suporte às versões principais do Next.js até a versão 11. Para obter uma lista detalhada dos atributos do Next.js compatíveis e incompatíveis com o Amplify, consulte supported features.

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

As imagens em seu aplicativo Next.js não estão carregando

Quando você adiciona imagens ao seu aplicativo Next.js usando o componente `next/image`, 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 AWS em que o Amplify 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.

Configurar domínios personalizados

É possível conectar um aplicativo que você implantou com o Amplify Hosting a um domínio personalizado. Quando você usa o Amplify para implantar seu aplicativo web, o Amplify o hospeda para você no `amplifyapp.com` domínio padrão 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`.

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 domínio credenciados terceirizados, consulte o Diretório de Registradores Credenciados no site da ICANN.

Ao configurar seu domínio personalizado, você pode usar o certificado gerenciado padrão que o Amplify fornece para você ou pode usar seu próprio certificado personalizado. Você pode alterar o certificado em uso para o domínio a qualquer momento. Para obter informações detalhadas sobre o gerenciamento de certificados, consulteUsando certificados SSL/TLS.

Antes de continuar com a configuração de um domínio personalizado, verifique se você atendeu aos seguintes pré-requisitos.

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

  Para obter mais informações sobre como concluir essa etapa, consulteConceitos básicos do Amplify Hosting.
- Você tem um 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.

Tópicos

- Usando certificados SSL/TLS
- Adicione um domínio personalizado gerenciado pelo Amazon Route 53
- Adicione um domínio personalizado gerenciado por um provedor DNS de terceiros
- Atualizar registros DNS para um domínio gerenciado pelo GoDaddy
- Atualizar registros DNS de um domínio gerenciado pelo Google Domains
- Atualizar o certificado SSL/TLS de um domínio
- Gerenciar subdomínios
- Subdomínios Wildcard
- Configure subdomínios automáticos para um domínio personalizado do Amazon Route 53
- Solucionar problemas de domínios personalizados

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 domínio personalizado do Amplify Hosting

Quando você adiciona um domínio personalizado com o Amplify Hosting, há várias etapas a serem concluídas antes que você possa visualizar seu aplicativo usando seu domínio personalizado. A lista a seguir descreve cada etapa do processo de configuração 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, você deve adicionar manualmente o registro de verificação de DNS fornecido no console do Amplify ao seu domínio com um provedor de DNS terceirizado.

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

Ativação do domínio

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

Usando certificados SSL/TLS

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

Com um certificado gerenciado, o Amplify emite um certificado SSL/TLS 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. Você deve excluir e adicionar o domínio novamente no console do Amplify.

Para usar um certificado personalizado, você deve obter um certificado da autoridade de certificação terceirizada de sua escolha. Em seguida, importe o certificado para AWS Certificate Manager o. O ACM é um serviço que permite provisionar, gerenciar e implantar com facilidade certificados SSL/TLS públicos e privados para uso com seus recursos internos Serviços da AWS 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. Você pode usar um caractere curinga no início do seu nome de domínio para cobrir vários subdomínios. Por exemplo, se o seu domínio for`example.com`, você pode incluir o domínio `*.example.com` curinga. Isso abrangerá subdomínios como e. `product.example.com` `api.example.com`

Depois que seu certificado personalizado estiver disponível no ACM, você poderá selecioná-lo durante o processo de configuração do domínio. Para obter instruções sobre como importar certificados para AWS Certificate Manager, consulte Importação de certificados AWS Certificate Manager no Guia do AWS Certificate Manager usuário.

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 gerencia as renovações automaticamente. Você é responsável por renovar seus certificados personalizados e importá-los novamente.

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

Adicione um domínio personalizado gerenciado pelo Amazon Route 53

Para adicionar um domínio personalizado gerenciado pelo Route 53

1. Faça login no AWS Management Console 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. Você pode escolher o domínio que deseja usar 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.com e https://example.com com um 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 certificado SSL/TLS 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 o certificado gerenciado Amplify.
   - Use um certificado personalizado de terceiros.
     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.

Adicione um domínio personalizado gerenciado por um provedor DNS de terceiros

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 o GoDaddy Google Domains, consulte the section called “Atualizar registros DNS para um domínio gerenciado pelo GoDaddy” ou consulte the section called “Atualizar registros

DNS de um domínio gerenciado pelo Google Domains" os procedimentos específicos desses provedores.

Adicionar um domínio personalizado gerenciado por um provedor DNS de terceiros

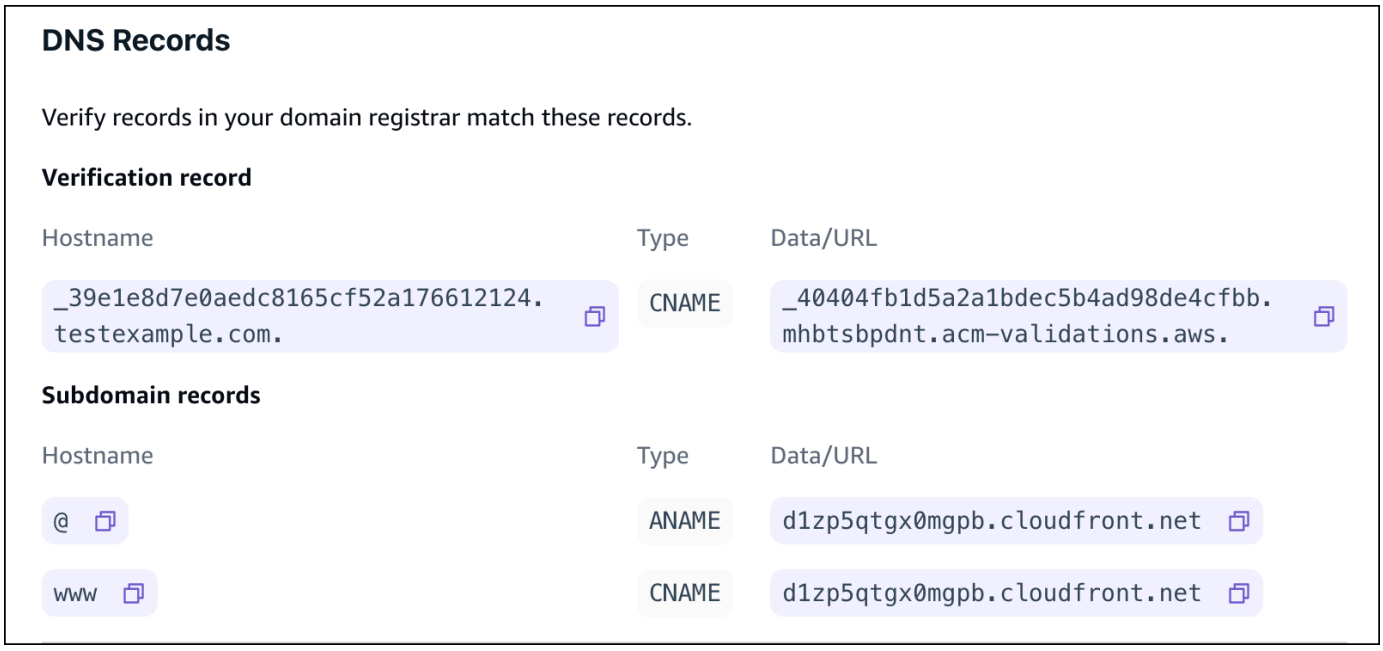
1. Faça login no AWS Management Console e abra o console do Amplify.
2. Escolha o aplicativo ao qual você deseja adicionar um domínio personalizado.
3. No painel de navegação, escolha Hospedagem, Domínios personalizados.
4. Na página Domínios personalizados, escolha Adicionar domínio.
5. Insira o nome 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. Acesse o 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. Vá para a etapa sete.
   - Para continuar com a configuração manual
     a. Escolha a configuração manual
     b. Escolha Configurar domínio.
     c. Vá 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.com e https://example.com com um 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 certificado SSL/TLS 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 o certificado gerenciado Amplify.
   - Use um certificado personalizado de terceiros.
     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 escolher Configuração manual, na etapa seis, deverá atualizar seus registros DNS com seu provedor de domínio terceirizado.

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



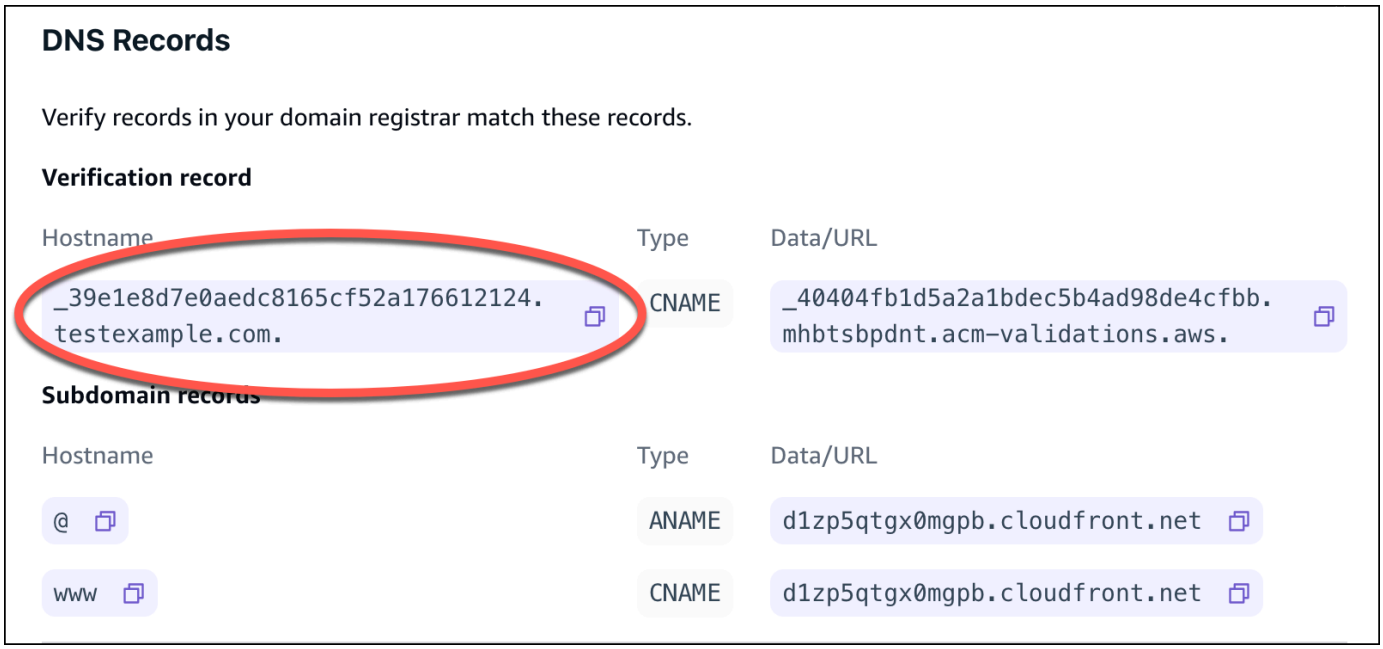
11. Execute um destes procedimentos:
    - Se você estiver usando GoDaddy, acesseAtualizar registros DNS para um domínio gerenciado pelo GoDaddy.

- Se você estiver usando o Google Domains, acesse Atualizar registros DNS de um domínio gerenciado pelo Google Domains.
- 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.

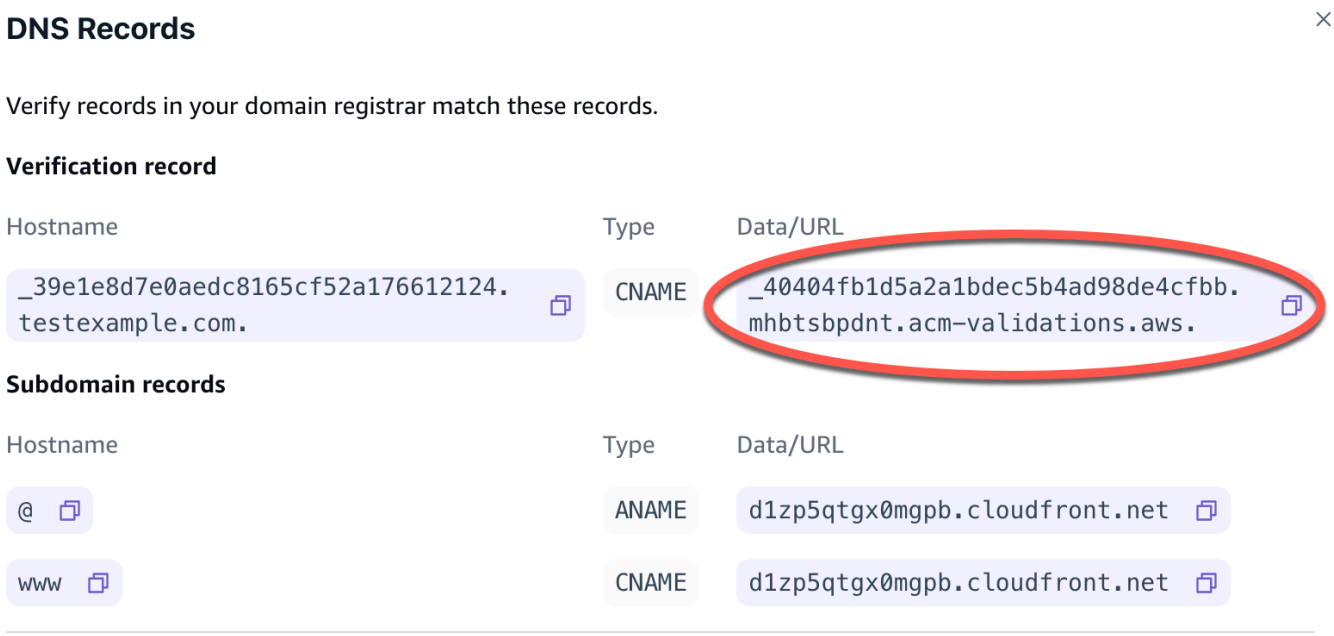
    Se o console do Amplify exibir um registro DNS para verificar a propriedade do seu subdomínio, como _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, insira somente o nome do subdomínio do registro CNAME. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

    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 o valor do registro CNAME. **`_cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`**

    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 certificado SSL/TLS 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. Você deve excluir e adicionar o domínio novamente no console do Amplify.

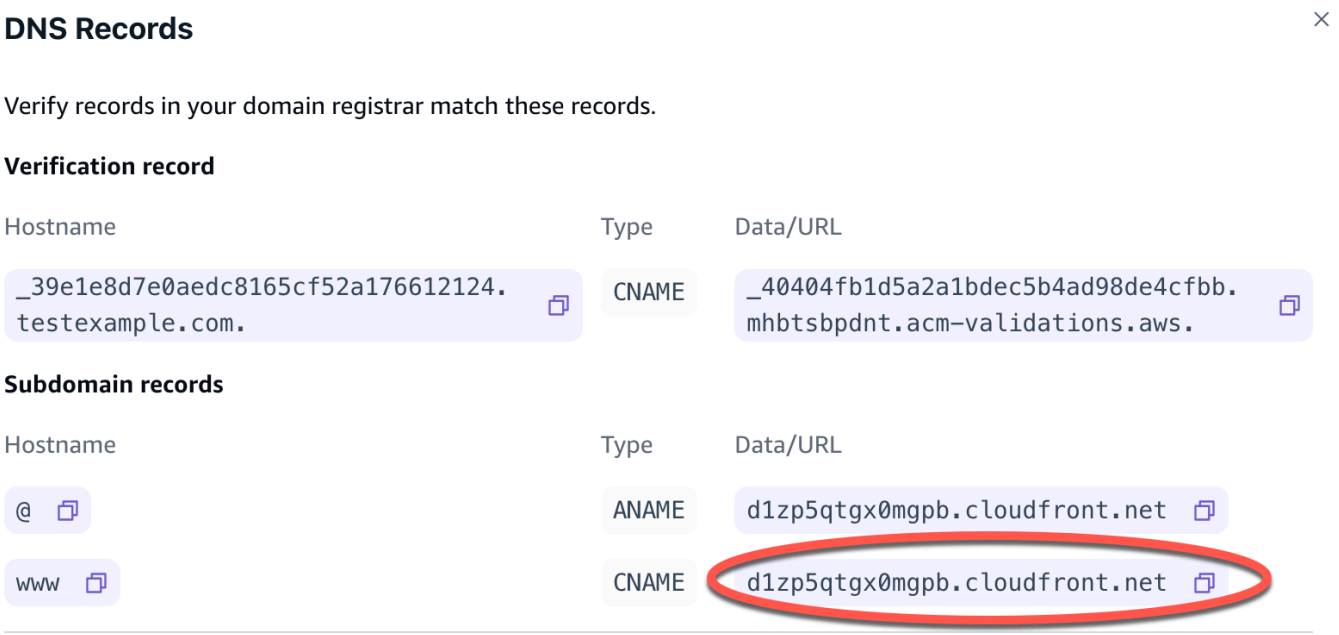
> **Important**
>
> É importante que você execute essa etapa logo após adicionar seu domínio personalizado no console do Amplify. O AWS Certificate Manager (ACM) começa imediatamente a tentar verificar a propriedade. Com o tempo, as verificações se tornam menos frequentes. Se você adicionar ou atualizar 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 direcionar seus subdomínios para o domínio Amplify. Por exemplo, se seu subdomínio for www.exemplo.com, insira www como nome do subdomínio.

    Se o console do Amplify exibir o domínio do seu aplicativo como d111111abcdef8.cloudfront.net, insira o domínio do Amplify. **`d111111abcdef8.cloudfront.net`**

    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 do nome de domínio a ser usado.



15. Configure o registro ANAME/ALIAS 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 DNS que não têm suporte para ANAME/ALIAS, é altamente recomendável migrar o DNS para o Route 53. Para obter mais informações, consulte Como configurar o Amazon Route 53 como seu serviço de DNS.

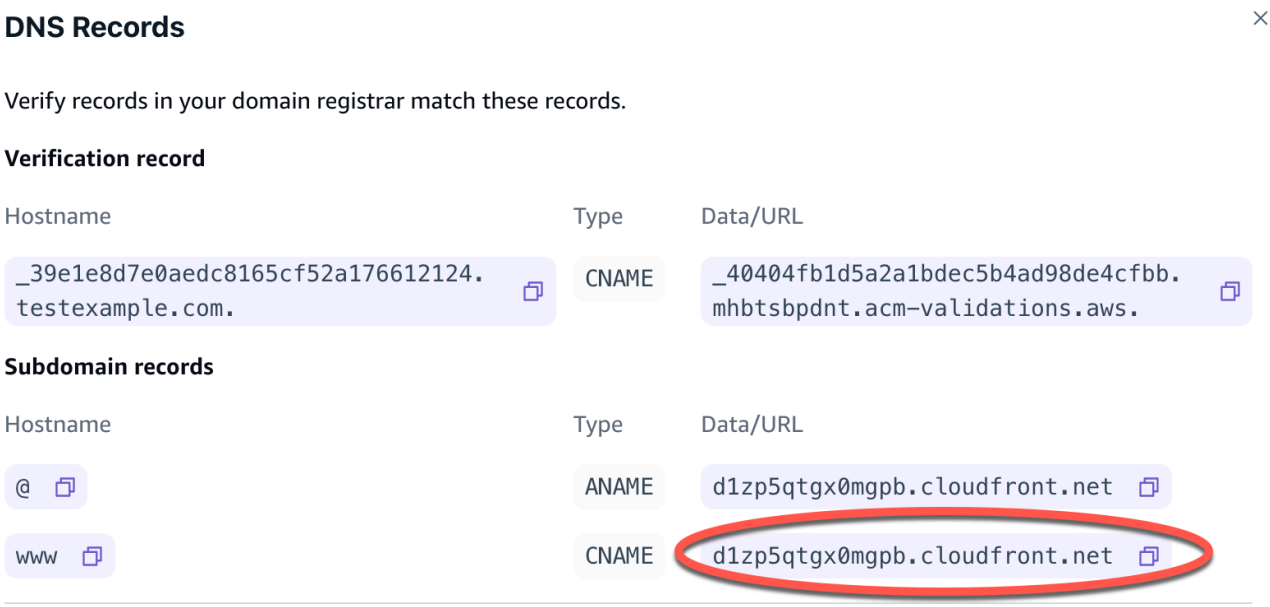
> **Note**
>
> A verificação de propriedade de domínio e a propagação de DNS para domínios de terceiros pode levar até 48 horas. Para obter ajuda na resolução de erros que ocorrem, consulte Solução de problemas de domínios personalizados.

Atualizar registros DNS para um domínio gerenciado pelo 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 “Adicione um domínio personalizado gerenciado por um provedor DNS de terceiros”.
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 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 do seu aplicativo como d111111abcdef8.cloudfront.net, insira Value. **`d111111abcdef8.cloudfront.net`**

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



e. Escolha Salvar.

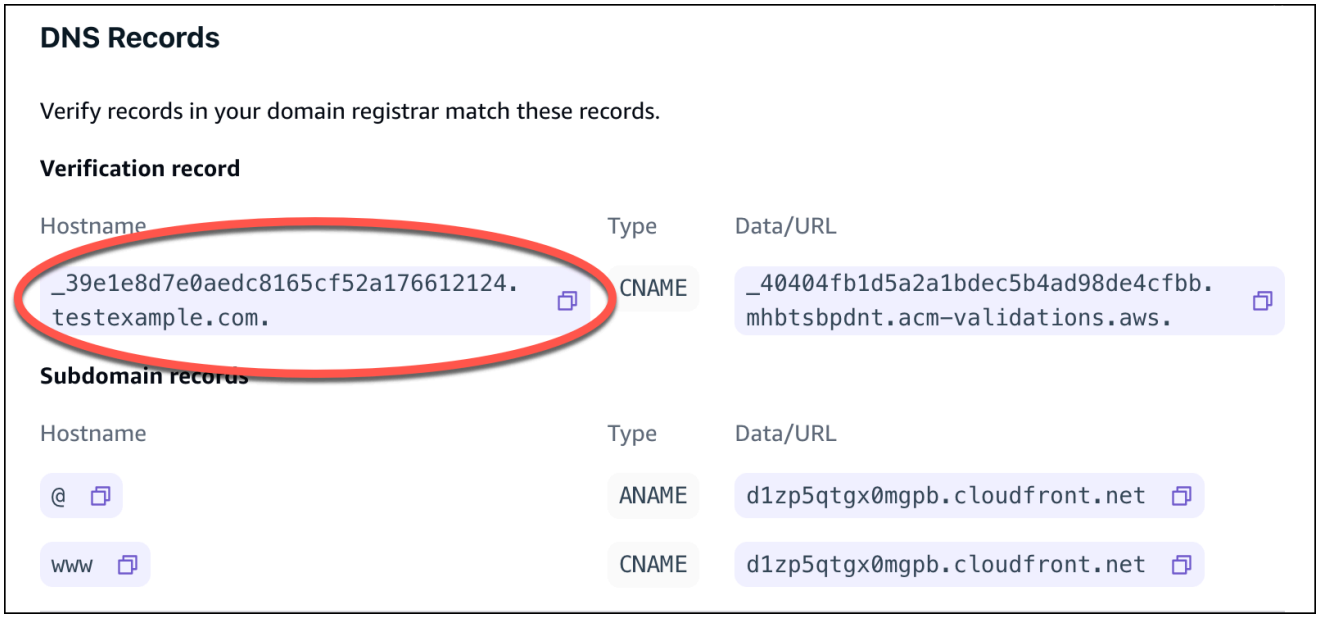
6. Crie o segundo registro CNAME para apontar para o servidor de validação AWS Certificate Manager (ACM). Um único ACM validade gera um certificado SSL/TLS para seu domínio.

   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, insira somente para Nome. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

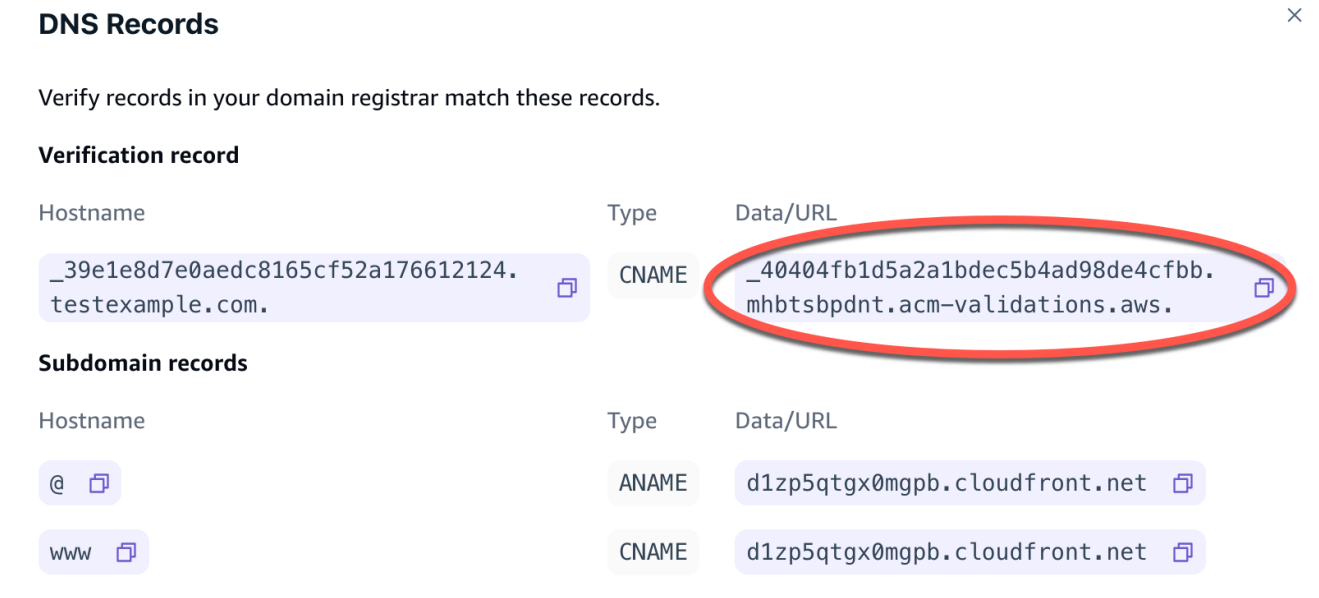
      A captura de tela a seguir mostra a localização do registro de verificação a ser usado.



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. Você deve excluir e adicionar o domínio novamente no console do Amplify.

7. Essa etapa não é necessária para subdomínios. GoDaddy não suporta registros ANAME/ALIAS. Para provedores DNS que não têm suporte para ANAME/ALIAS, é altamente recomendável migrar o 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 para o qual encaminhar (por exemplo, www.exemplo.com) para o URL de destino.
   d. Em Tipo de encaminhamento, escolha Temporário (302).
   e. Escolha Salvar.

Atualizar registros DNS de um domínio gerenciado pelo Google Domains

Para adicionar um domínio personalizado gerenciado pelo Google Domains

1. Antes de atualizar seus registros DNS com o Google Domains, conclua as etapas de um a nove do procedimento Para adicionar um domínio personalizado gerenciado por um provedor de DNS terceirizado.
2. Faça login na sua conta em https://domains.google.com e escolha Meus domínios no painel de navegação esquerdo.
3. Na sua lista de domínios, encontre o domínio a ser adicionado e escolha Gerenciar.

4. No painel de navegação à esquerda, escolha DNS. O Google exibe os Registros de recursos do seu domínio. Você precisa adicionar dois novos registros CNAME.
5. Crie o primeiro registro CNAME para direcionar todos os subdomínios para o domínio Amplify da seguinte forma:

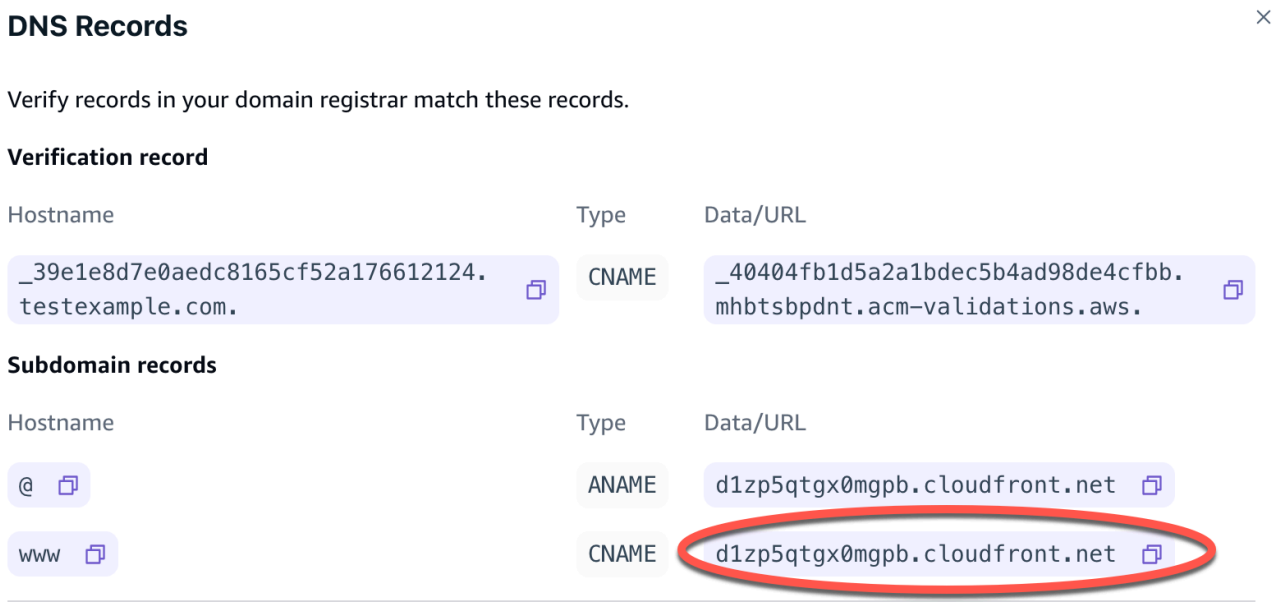
   a. Em Nome do host, insira somente o nome do subdomínio. Por exemplo, se seu subdomínio for www.exemplo.com, digite www como Nome do host.

   b. Para Tipo, escolha CNAME.

   c. Em Dados, insira o valor que está disponível no console do Amplify.

      Se o console do Amplify exibir o domínio do seu aplicativo como d111111abcdef8.cloudfront.net, digite d111111abcdef8.cloudfront.net para Data.

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

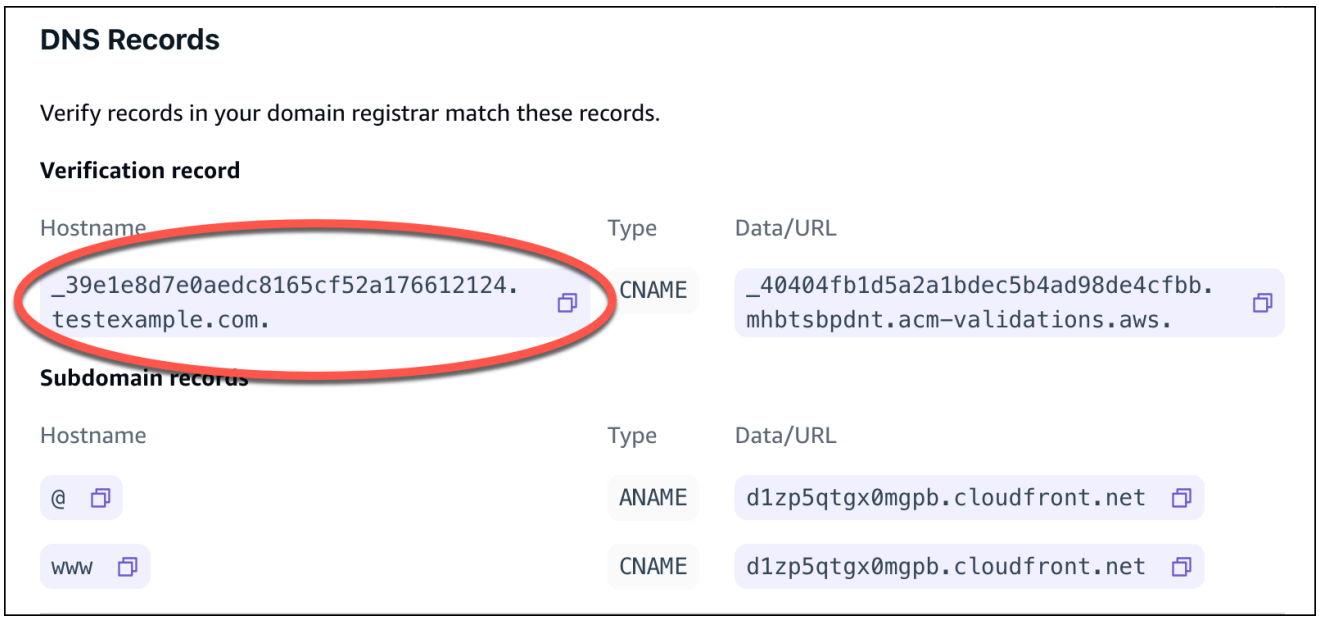


6. Crie o segundo registro CNAME para apontar para o servidor de validação AWS Certificate Manager (ACM). Um único ACM validade gera um certificado SSL/TLS para seu domínio.

   a. Em Nome do host, insira o subdomínio.

      Por exemplo, se o registro DNS no console do Amplify para verificar a propriedade do seu subdomínio for _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, insira somente _c3e2d7eaf1e656b73f46cd6980fdc0e como nome do host.

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

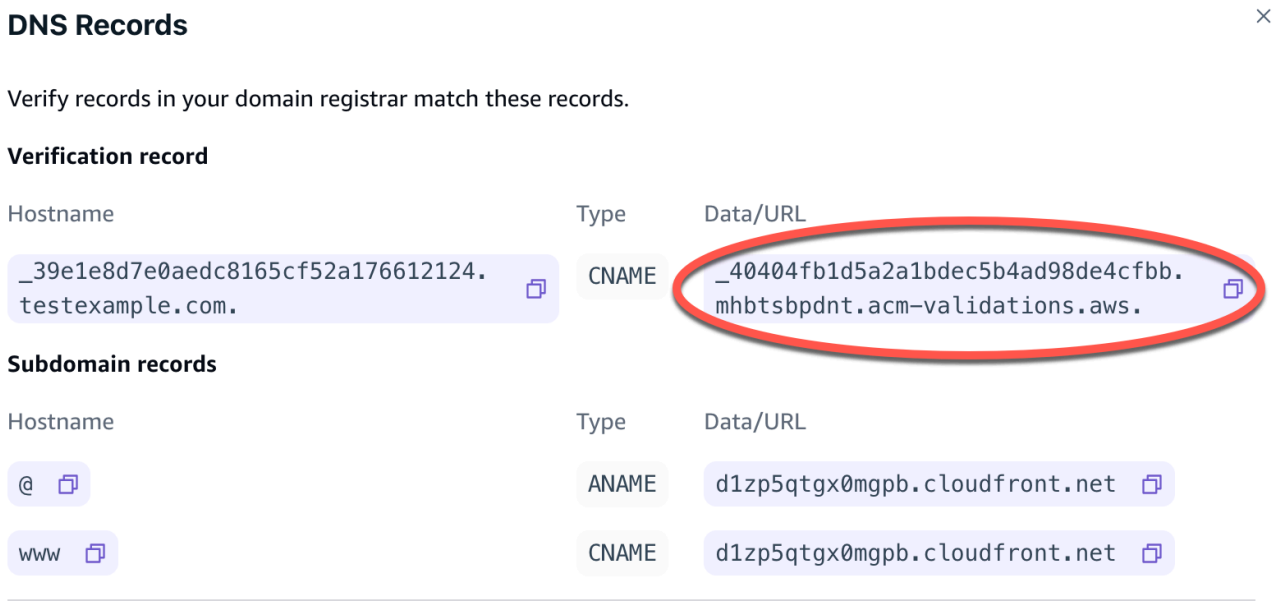


b. Para Tipo, escolha CNAME.

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

Por exemplo, se o servidor de validação for _cf1z2npwt9vzexample93c1j4xzc92wl.2te3iym6kzr.acm-validations.aws., digite _cf1z2npwt9vzexample93c1j4xzc92wl.2te3iym6kzr.acm-validations.aws. para Dados.

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



7. 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. Você deve excluir e adicionar o domínio novamente no console do Amplify.

8. O suporte do Google Domains para registros ANAME/ALIAS está em versão prévia. Para provedores DNS que não têm suporte para ANAME/ALIAS, é altamente recomendável migrar o 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 manter o Google Domains como seu provedor e atualizar o domínio raiz, configure o encaminhamento de um domínio. Localize a página do site do seu domínio do Google. Em seguida, escolha Encaminhar domínio e configure seu encaminhamento na página de encaminhamento pela Web.

> **Note**
>
> As atualizações nas configurações de DNS para um domínio do Google podem levar até 48 horas para serem efetivadas. Para obter ajuda na resolução de erros que ocorrem, consulte Solução de problemas de domínios personalizados.

Atualizar o certificado SSL/TLS de um domínio

Você pode alterar o certificado SSL/TLS que está em uso para um domínio a qualquer momento. Por exemplo, você pode deixar de usar um certificado gerenciado para usar um certificado personalizado. Você também pode alterar o certificado personalizado que está em uso para o domínio. Para obter mais informações sobre certificados, consulte Usando 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 o certificado de um domínio

1. Faça login no AWS Management Console e abra o console do Amplify.
2. Escolha o aplicativo que você deseja atualizar.

3. No painel de navegação, escolha Hospedagem, Domínios personalizados.
4. Na página Domínios personalizados, escolha Configuração do 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 o certificado gerenciado 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 alterar um certificado personalizado para um certificado personalizado diferente
     - Para Certificado SSL personalizado, selecione o novo certificado a ser usado na lista.
6. Escolha Salvar. Os detalhes do 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 AWS Management Console 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, insira 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 AWS Management Console 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, insira beta como o 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 AWS Management Console 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 do domínio.
5. Na seção Subdomínios, você pode editar seus subdomínios existentes conforme necessário.
6. (Opcional) Para adicionar um novo subdomínio, escolha Adicionar novo.
7. Escolha Salvar.

Subdomínios Wildcard

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 AWS Management Console 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 do domínio.
5. Na seção Subdomínios, você pode 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.

Configure 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 AWS Management Console 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 do domínio.
5. Na seção Criação automática de subdomínio, 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 ativar a criação automática de subdomínios usando as instruções anteriores, as visualizações da web do pull request do seu aplicativo também estarão acessíveis com 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 um aplicativo no console AWS Amplify , consulte os tópicos a seguir nesta seção para obter ajuda na solução de problemas.

Se você não encontrar uma solução para seu problema aqui, entre em contato com o AWS Support. Para obter mais informações, consulte Criação de um caso de suporte no Guia do usuário do AWS Support .

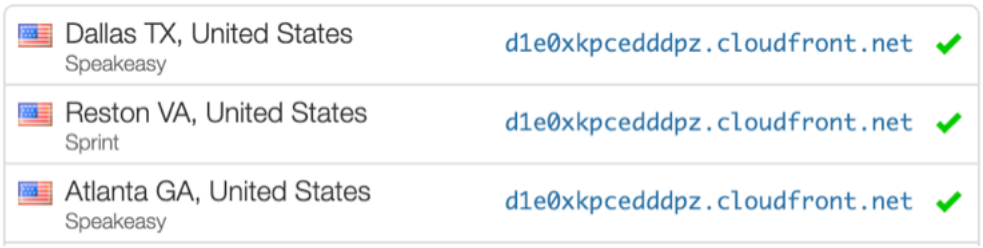
Tópicos

- Como faço para verificar se o CNAME foi 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
- Eu recebo um erro CNAME AlreadyExistsException
- 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

Como faço para verificar se o CNAME foi resolvido?

1. Depois de atualizar seus registros DNS com seu provedor de domínio terceirizado, é possível usar uma ferramenta como dig ou um site gratuito, como 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.

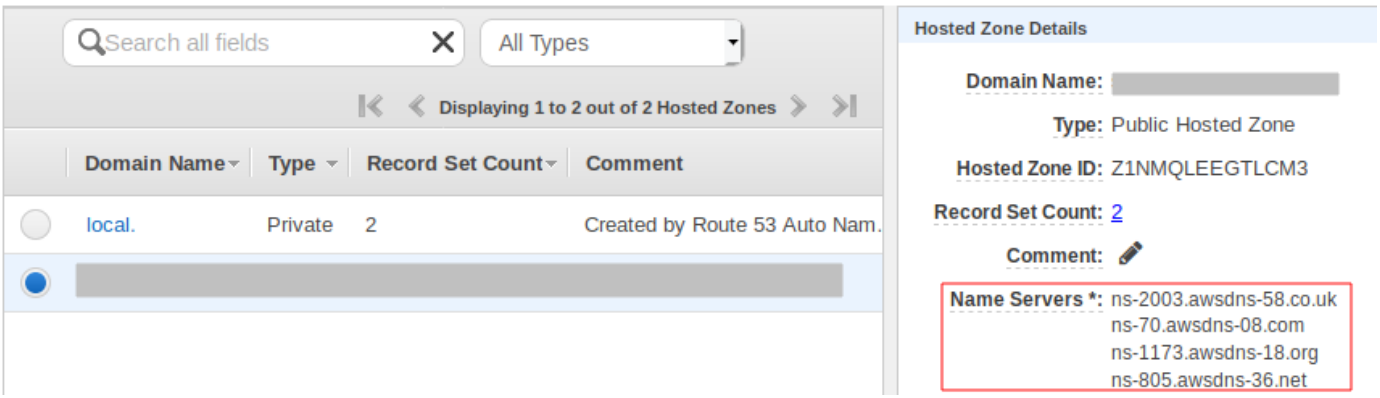
> **Important**
>
> É importante atualizar seus registros CNAME assim que você criar seu domínio personalizado. Depois que o aplicativo é criado no console do Amplify, o registro CNAME é verificado em intervalos de alguns minutos para determinar se ocorre solução. Se não ocorrer depois de uma hora, a 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 CNAMEalgumas horas depois de criar seu aplicativo, essa é a causa mais provável de seu aplicativo ficar preso 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.

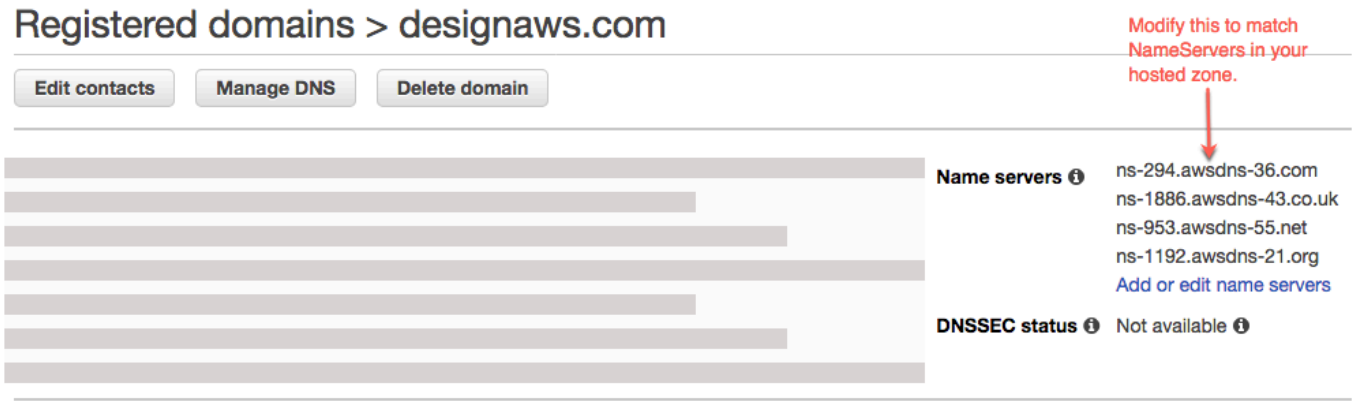
Meu domínio hospedado com o Amazon Route 53 está paralisado no estado de verificação pendente

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 AWS Support. Para obter mais informações, consulte Criação de um caso de suporte no Guia do usuário do AWS Support .

Eu recebo um erro CNAME AlreadyExistsException

Se você receber um AlreadyExistsException erro CNAME, 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 Execute as seguintes etapas para diagnosticar a causa do erro.

1. Faça login no CloudFrontconsole da Amazon e verifique se você não tem esse domínio implantado em nenhuma 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. 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.exemplo.com para outro aplicativo, não poderá usar www.exemplo.com com o aplicativo que você está conectando no momento. É possível usar outros subdomínios, como blog.example.com.

4. 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 AWS Support. Para obter mais informações, consulte Criação de um caso de suporte no Guia do usuário do AWS Support .

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 AWS Support. 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 domínio pertencentes à Amazon, 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, você deve 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 AWS Management Console 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.

   - Adicione um domínio personalizado gerenciado pelo Amazon Route 53.
   - Adicione um domínio personalizado gerenciado por um provedor DNS de terceiros.
   - Atualizar registros DNS para um domínio gerenciado pelo GoDaddy.
   - Atualizar registros DNS de um domínio gerenciado pelo Google Domains.

- Entre em contato AWS Support para que seu certificado SSL seja reemitido.

Configurar as definições de compilação

Quando você implanta um aplicativo com o Amplify Hosting, ele detecta automaticamente a estrutura de front-end e as configurações de compilação associadas, inspecionando o arquivo `package.json` em seu repositório. 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.

Você pode editar as configurações de criação de um aplicativo no console do Amplify escolhendo Hospedagem e, em seguida, Configurações de criação no painel de navegação. As configurações são aplicadas a todas as ramificações do aplicativo, exceto às ramificações que têm um arquivo `amplify.yml` salvo no repositório.

> **Note**
>
> As configurações de compilação ficam visíveis no menu Hospedagem do console do 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.

Comandos e configurações de especificação de compilação

O YAML de especificação de compilação contém uma coleção de comandos de compilação e configurações relacionadas que o Amplify usa para executar a compilação. A lista a seguir descreve essas configurações e como elas são usadas.

versão

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.

back-end

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

front-end

Executa comandos de compilação de front-end.

teste

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

fases da compilação

O front-end, o back-end 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

O campo cache do buildspec é usado para armazenar em cache dependências de tempo de construção, como a pasta node_modules, e é sugerido automaticamente com base no gerenciador de pacotes e na estrutura em que o aplicativo do cliente está incorporado. Durante a primeira compilação, todos os caminhos aqui são armazenados em cache e, nas compilações subsequentes, reinflamos o cache e usamos essas dependências em cache sempre que possível para acelerar o tempo de construção.

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

Sintaxe YAML de especificação de compilação

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

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

Configurações de compilação específicas de ramificação

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
```

Navegar para 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
```

Implantando o back-end com o front-end para um aplicativo de primeira geração

> **Note**
>
> Esta seção se aplica somente aos aplicativos Amplify Gen 1. Um back-end de primeira geração 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 back-end. As configurações de compilação abaixo determinam automaticamente o ambiente de back-end 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 imagem AL2023 do Amplify executa seu código com um nome de usuário sem privilégios. `amplify` O Amplify concede a esse usuário privilégios para executar comandos do sistema operacional usando o comando Linux. `sudo` Se quiser instalar pacotes de sistema operacional para dependências ausentes, você pode usar comandos como `yum` e `rpm` with`sudo`.

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

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

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 um commit

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

Desativar compilações automáticas

É possível configurar o Amplify para desabilitar compilações automáticas em cada confirmação de código. Para configurar, escolha Configurações do aplicativo, Configurações de ramificação e, em seguida, localize a seção Ramificações que lista as ramificações conectadas. Selecione uma ramificação e, em seguida, escolha Ações, Desativar criação automática. Novos commits nessa ramificação não iniciarão mais uma nova compilação.

Ativar ou desativar a compilação e implantação de front-end com base em diff

É possível configurar o Amplify para usar compilações de front-end 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 front-end e não atualiza seu aplicativo hospedado.

Para configurar o front-end baseado em diff, criar e implantar

1. Faça login AWS Management Console e abra o console do Amplify.
2. Escolha o aplicativo para o qual configurar a compilação e implantação de front-end 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 front-end com base em diff.
   - Para habilitar a compilação e implantação de front-end 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 front-end 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`.

Ativar ou desativar compilações de back-end baseadas em diferenças para um aplicativo de primeira geração

> **Note**
>
> Esta seção se aplica somente aos aplicativos Amplify Gen 1. Um back-end de primeira geração 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 back-end baseadas em diff usando a variável de ambiente `AMPLIFY_DIFF_BACKEND`. Quando você habilita compilações de back-end 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 back-end e não atualiza seus atributos de back-end. 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 back-end, as compilações condicionais de back-end não funcionarão. Se quiser que esses comandos personalizados sejam executados, você deve movê-los para a fase de front-end das configurações de compilação no arquivo `amplify.yml` do seu aplicativo.

Para configurar compilações de back-end baseadas em diff

1. Faça login AWS Management Console e abra o console do Amplify.

2. Escolha o aplicativo para o qual configurar as compilações de back-end 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 back-end baseadas em diff.
   - Para habilitar compilações de back-end baseadas em diff
     a. Na seção Gerenciar variáveis, em Variável, insira `AMPLIFY_DIFF_BACKEND`.
     b. Em Valor, insira `true`.
   - Para desativar as compilações de back-end 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.

Configurações de compilação do 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 ter 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.

Sintaxe YAML da especificação de compilação do 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. Você deve 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.

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*
```

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 AWS Management Console e abra o console do Amplify.
2. Escolha Criar novo aplicativo no canto superior direito.
3. Na página Comece a criar 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 filial a ser usada.

    c. Selecione Meu aplicativo é um monorepo

    d. Insira o caminho para seu aplicativo em seu monorepo, por exemplo, **apps/app1**.

    e. Escolha Próximo.

5. Na página de configurações do aplicativo, você pode usar as configurações padrão ou personalizar as configurações de compilação do seu aplicativo. Na seção Variáveis de ambiente, Amplify define o caminho `AMPLIFY_MONOREPO_APP_ROOT` 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 AWS Management Console 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, escolha 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, você deve 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, você deve 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
```

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 branches 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 recursos adicionais.

Para adicionar uma ramificação a um aplicativo

1. Escolha o aplicativo ao qual você deseja adicionar uma ramificação.
2. Escolha Configurações do aplicativo e, em seguida, Configurações da ramificação.
3. Na página de configurações da ramificação, escolha Adicionar ramificação.
4. Selecione uma ramificação do seu repositório.
5. Escolha Adicionar ramificação.
6. Reimplante seu aplicativo.

Depois de adicionar uma ramificação, seu aplicativo tem duas implantações disponíveis nos domínios padrão do Amplify, como https://main.appid.amplifyapp.com e https://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 os back-ends do Amplify em todos os aplicativos (somente aplicativos de primeira geração)

Fluxos de trabalho de equipe com aplicativos Amplify Gen 2 completos

AWS O Amplify Gen 2 apresenta uma experiência de desenvolvedor TypeScript baseada em código para definir back-ends. Para saber mais sobre fluxos de trabalho fullstack com os aplicativos do Amplify Gen 2, consulte Fluxos de trabalho do Fullstack nos documentos do Amplify.

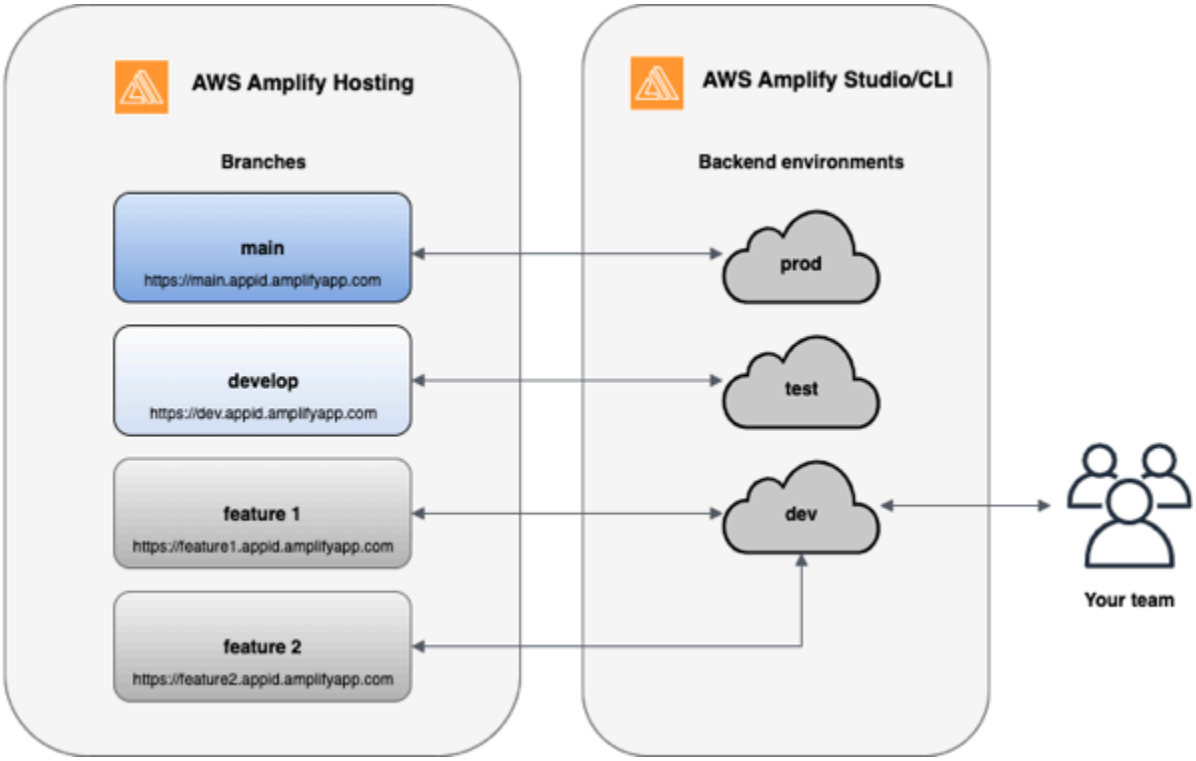
Fluxos de trabalho de equipe com aplicativos Amplify Gen 1 completos

A implantação de uma ramificação de atributos consiste em um ambiente de front-end e um ambiente de back-end opcional. O front-end é construído e implantado em uma rede global de entrega de conteúdo (CDN), enquanto o back-end é implantado pelo Amplify Studio ou pela CLI do Amplify para AWS. Para saber como configurar esse cenário de implantação, consulteConstruindo um back-end para um aplicativo.

O Amplify Hosting implanta continuamente atributos de back-end como APIs do GraphQL e funções do Lambda com sua implantação de ramificação de atributo. É possível usar os seguintes modelos de ramificação para implantar seu back-end e front-end com Amplify Hosting.

Fluxo de trabalho de ramificação de atributo

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

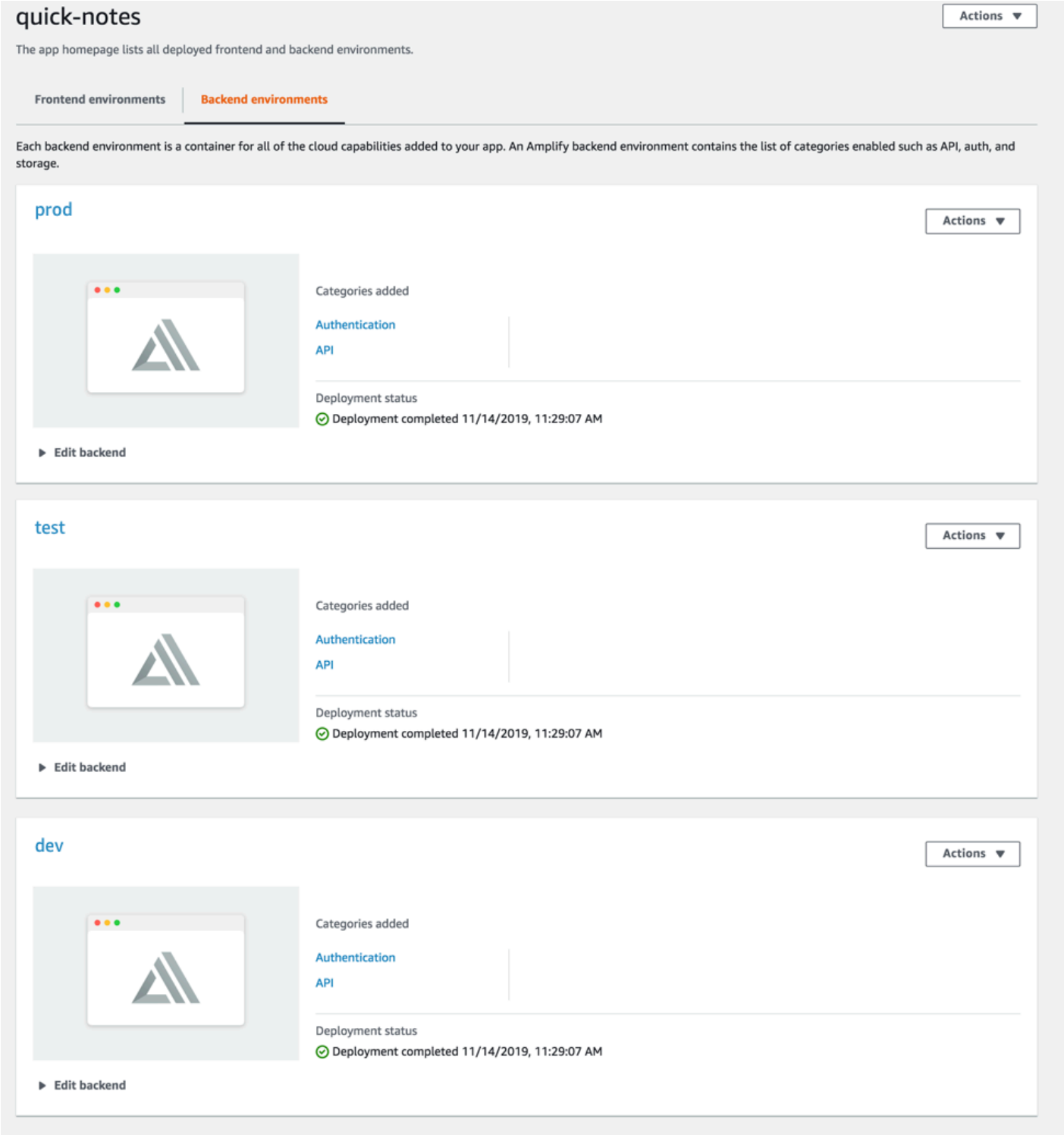
```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: test
...
amplify push
```

```
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).

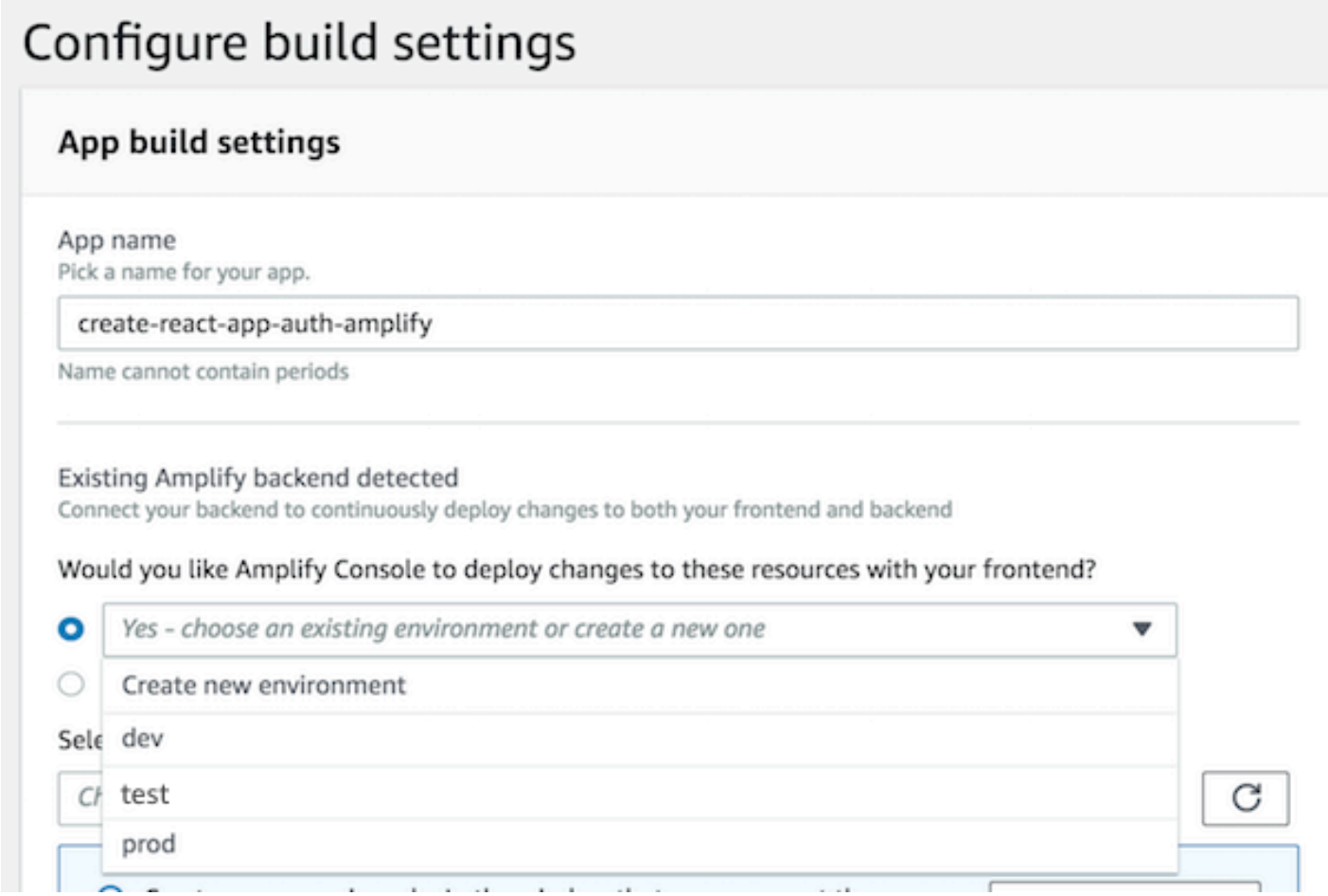
```
git commit -am 'Added dev, test, and prod environments'
git push origin main
```

5. Visite Amplify no AWS Management Console 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 back-end criados na guia Ambientes de back-end.

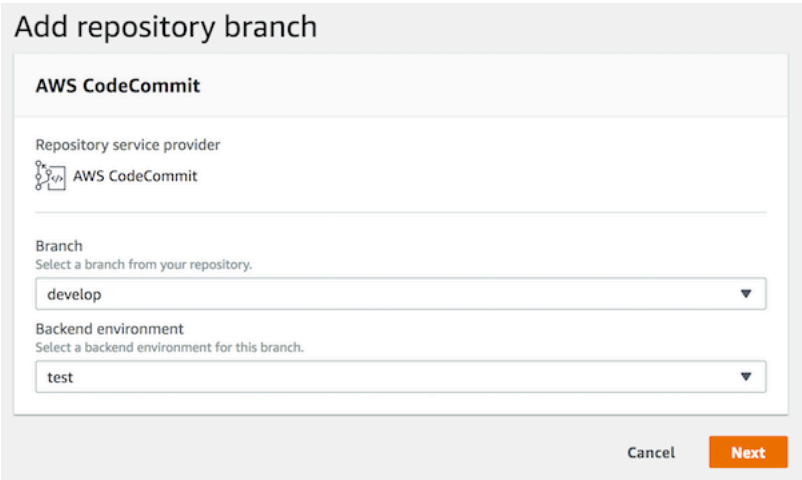


6. Mude para a guia Ambientes de front-end e conecte seu provedor de repositório e a ramificação principal.
7. Na tela de configurações de compilação, escolha um ambiente de back-end existente para configurar a implantação contínua com a ramificação principal. Escolha prod no menu suspenso

e conceda o perfil de serviço para o Amplify. Escolha Salvar e implantar. Depois que a compilação for concluída, haverá uma implantação de ramificação principal disponível em https://master.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 back-end test.



9. O Amplify agora está configurado. É possível começar a trabalhar em novos atributos em um atributo de atributo. Adicione a funcionalidade de back-end usando o ambiente de back-end 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 atributo estará acessível em https://newinternet.appid.amplifyapp.com para compartilhar com sua 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 back-end da nuvem (você sempre pode criar executando ‘amplify env checkout prod’ e executando ‘amplify env add’).

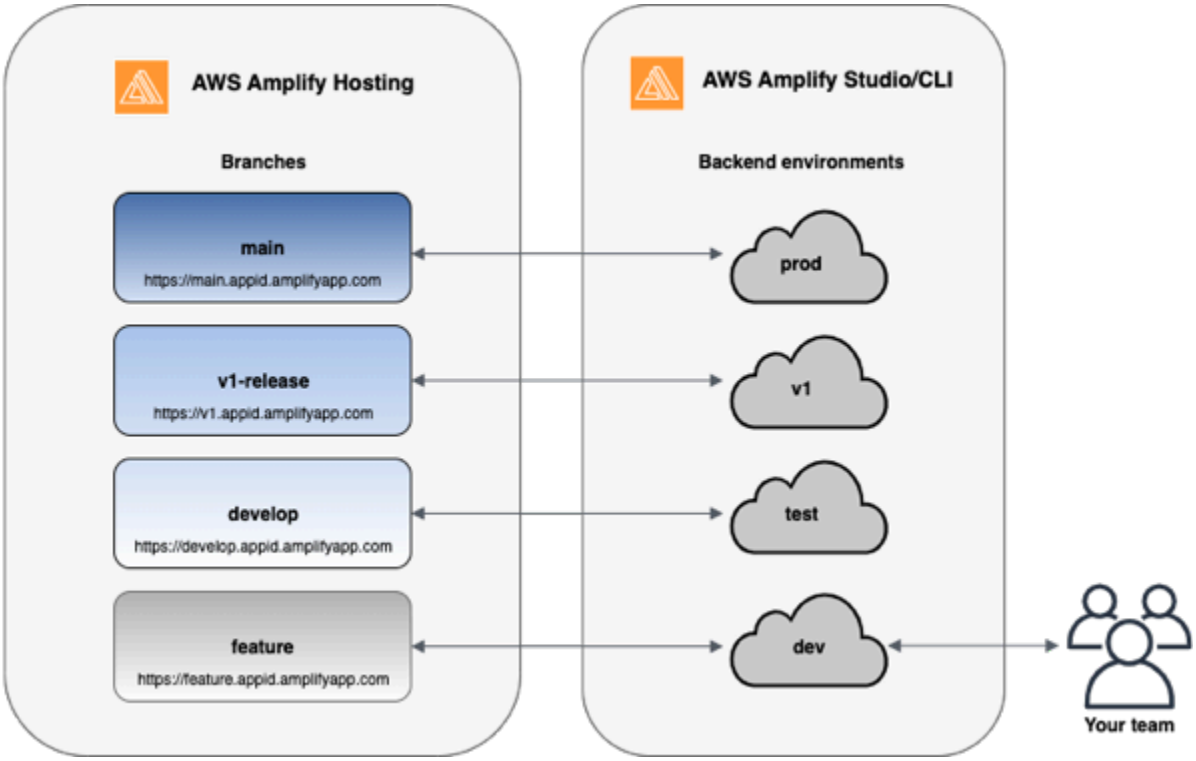
```
git push origin --delete newinternet
```

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

GitFlow fluxo de trabalho

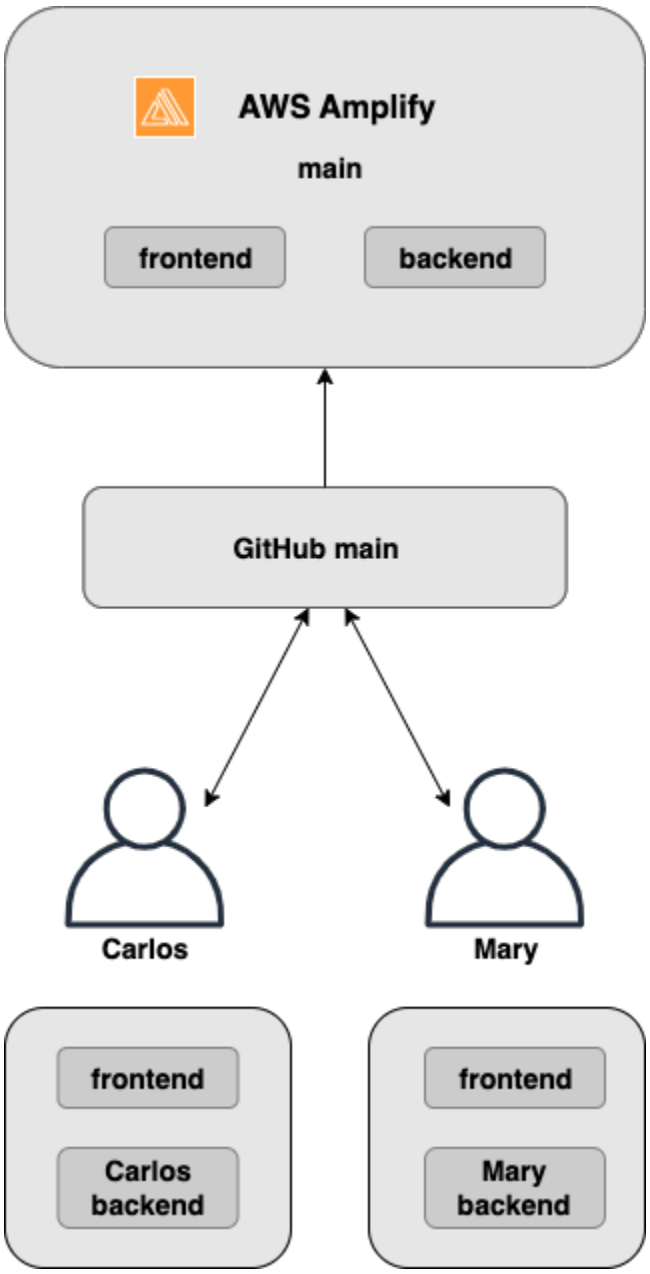
GitFlow usa duas ramificações para registrar o histórico do projeto. A ramificação principal rastreia somente o código de lançamento, e a ramificação de desenvolvimento é usada como uma ramificação de integração para novos recursos. GitFlow simplifica o desenvolvimento paralelo ao isolar o novo desenvolvimento 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 back-end. 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 back-end ou front-end 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 back-end 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 back-end 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. Depois que a compilação for concluída, haverá uma implantação de ramificação principal disponível em https://master.appid.amplifyapp.com com um novo ambiente de back-end que é vinculado à ramificação.
6. Conecte a ramificação de desenvolvimento no Amplify (suponha que a ramificação de desenvolvimento e a ramificação principal sejam iguais neste momento) 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**' para implantar automaticamente branches do Git que começam com 'release' em uma URL compartilhável. Essa postagem do blog descreve como usar esse atributo com diferentes fluxos de trabalho de equipe.

1. Escolha Configurações do aplicativo > Configurações de ramificação > Editar.
2. Selecione Detecção automática de ramificações para conectar automaticamente as ramificações ao Amplify que correspondam a um conjunto de padrões.
3. 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 que começam com a palavra “lançamento”.
   - **`release*/`** – Implantar todas as ramificações que correspondem a um padrão “release /”.

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

4. Configure a proteção automática por senha para todas as filiais criadas automaticamente selecionando Controle de acesso por detecção automática de filiais.
5. Para aplicativos de primeira geração criados com um back-end Amplify, você pode escolher criar um novo ambiente para cada ramificação conectada ou direcionar todas as ramificações para um back-end existente.
6. 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 Configure 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 Adicione 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.

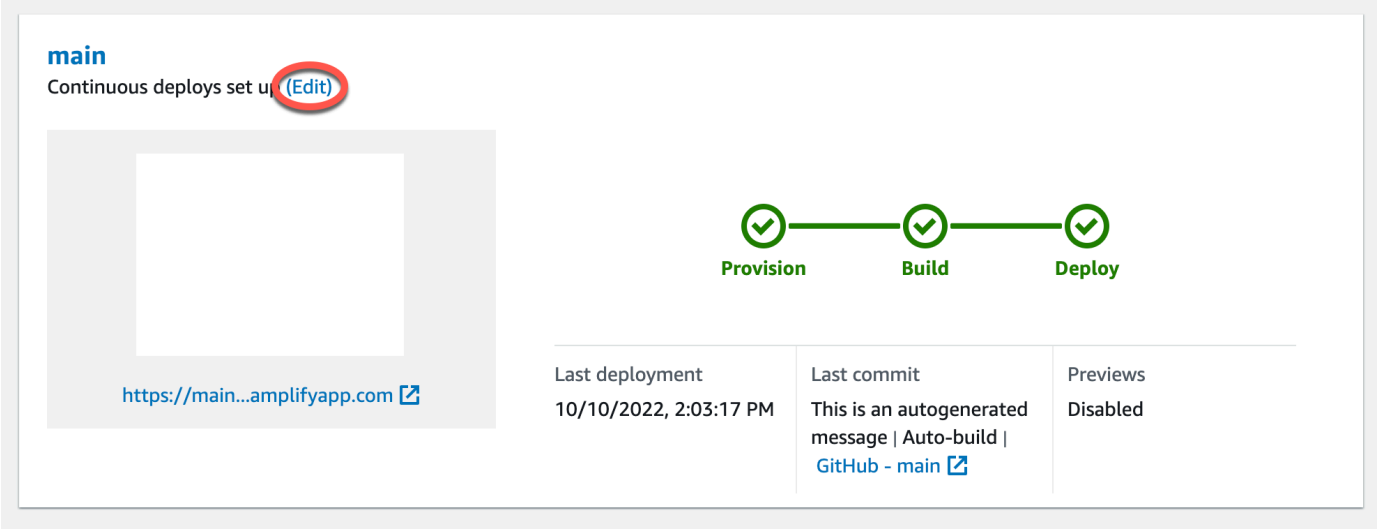
Geração automática em tempo de construção da configuração do Amplify (somente aplicativos de primeira geração)

> **Note**
>
> As informações nesta seção são somente para aplicativos de primeira geração. Se você quiser implantar automaticamente alterações de infraestrutura e código de aplicativo a partir de ramificações de recursos para um aplicativo de 2ª geração, consulte Implantações de ramificações do Fullstack na documentação do Amplify

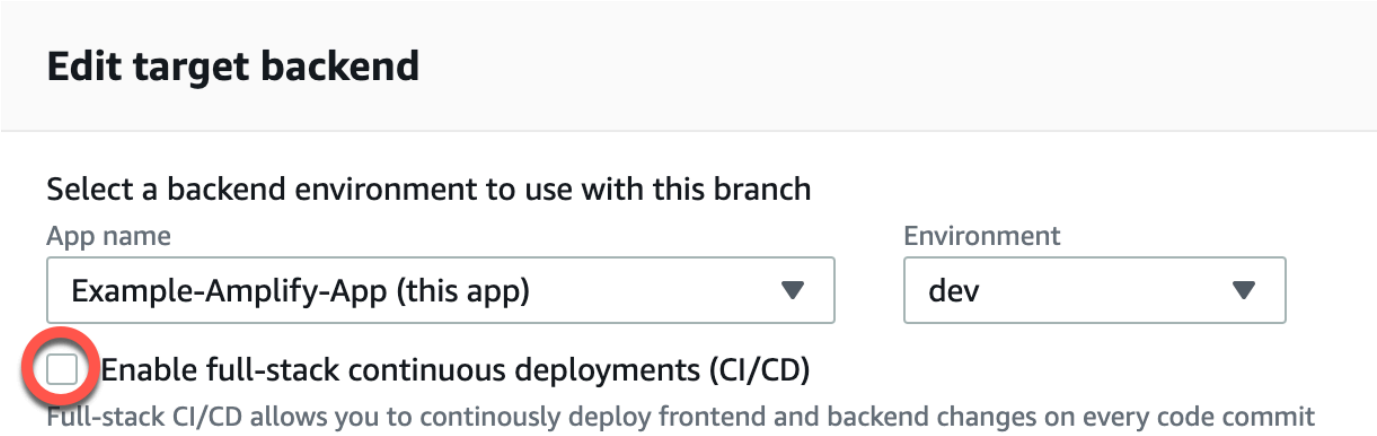
O Amplify suporta a geração automática em tempo de construção do arquivo de configuração `aws-exports.js` do Amplify para aplicativos de primeira geração. Ao desativar as implantações de CI/CD de pilha completa, você permite que seu aplicativo gere automaticamente o arquivo `aws-exports.js` e garante que as atualizações não sejam feitas em seu back-end no momento da compilação.

Para gerar automaticamente **`aws-exports.js`** no momento da construção

1. Faça login AWS Management Console 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 back-end de destino, desmarque Habilitar implantações contínuas de pilha completa (CI/CD) para desativar a CI/CD de pilha completa para esse back-end.



6. Selecione um perfil de serviço existente para dar ao Amplify as permissões necessárias para fazer alterações no back-end 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 Adicionar um perfil de serviço.
7. Escolha Salvar. O Amplify aplica essas alterações na próxima vez que você criar o aplicativo.

Compilações condicionais de back-end (somente aplicativos de primeira geração)

> **Note**
>
> As informações nesta seção são somente para aplicativos de primeira geração. O Amplify Gen 2 apresenta uma experiência de desenvolvedor TypeScript baseada em código. Portanto, esse recurso não é necessário para back-ends de segunda geração.

O Amplify oferece suporte a construções condicionais de back-end em todas as ramificações de um aplicativo de primeira geração. Para configurar compilações de back-end condicionais, defina a variável de ambiente `AMPLIFY_DIFF_BACKEND` como `true`. Habilitar compilações condicionais de back-end ajudará a acelerar as compilações em que as alterações são feitas somente no front-end.

Quando você habilita compilações de back-end 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 back-end e não atualiza seus atributos de back-end. 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 Ativar ou desativar compilações de back-end baseadas em diferenças para um aplicativo de primeira geração.

Se você atualmente tem comandos personalizados especificados nas configurações de compilação da sua fase de back-end, as compilações condicionais de back-end não funcionarão. Se quiser que esses comandos personalizados sejam executados, você deve movê-los para a fase de front-end das configurações de compilação no arquivo `amplify.yml` do seu aplicativo. Para obter mais informações sobre como atualizar o arquivo `amplify.yml`, consulte Comandos e configurações de especificação de compilação.

Use os back-ends do Amplify em todos os aplicativos (somente aplicativos de primeira geração)

> **Note**
>
> As informações nesta seção são somente para aplicativos de primeira geração. Se você quiser compartilhar recursos de back-end para um aplicativo de segunda geração, consulte Compartilhar recursos entre filiais nos documentos do Amplify

O Amplify permite que você reutilize ambientes de back-end existentes em todos os seus aplicativos de primeira geração em uma determinada região. É possível fazer isso ao criar um aplicativo, conectar uma nova ramificação a um aplicativo existente ou atualizar um front-end existente para apontar para um ambiente de back-end diferente.

Reutilize back-ends ao criar um aplicativo

Para reutilizar um back-end ao criar um aplicativo Amplify

1. Faça login AWS Management Console e abra o console do Amplify.
2. Para criar um back-end 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 front-end ao seu novo back-end, 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 back-end. É possível escolher o aplicativo atual ou qualquer outro aplicativo na região atual.

b. Em Ambiente, selecione o nome do ambiente de back-end a ser adicionado. É possível usar um ambiente existente ou criar outro.

c. Por padrão, o CI/CD de pilha completa está desativado. Desativar o CI/CD de pilha completa 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 back-end.

d. Selecione um perfil de serviço existente para dar ao Amplify as permissões necessárias para fazer alterações no back-end 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 Adicionar um perfil de serviço.

e. Escolha Próximo.

7. Escolha Salvar e implantar.

Reutilize back-ends ao conectar uma ramificação a um aplicativo existente

Para reutilizar um back-end ao conectar uma ramificação a um aplicativo Amplify existente

1. Faça login AWS Management Console 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 back-end. É possível escolher o aplicativo atual ou qualquer outro aplicativo na região atual.
7. Em Ambiente, selecione o nome do ambiente de back-end 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 back-end do seu aplicativo, o console solicitará que você execute essa tarefa. Para obter mais informações sobre como criar um perfil de serviço, consulte Adicionar um perfil de serviço.
9. Por padrão, o CI/CD de pilha completa está desativado. Desativar o CI/CD de pilha completa faz com que o aplicativo seja executado no modo pull only. No momento da compilação, o Amplify

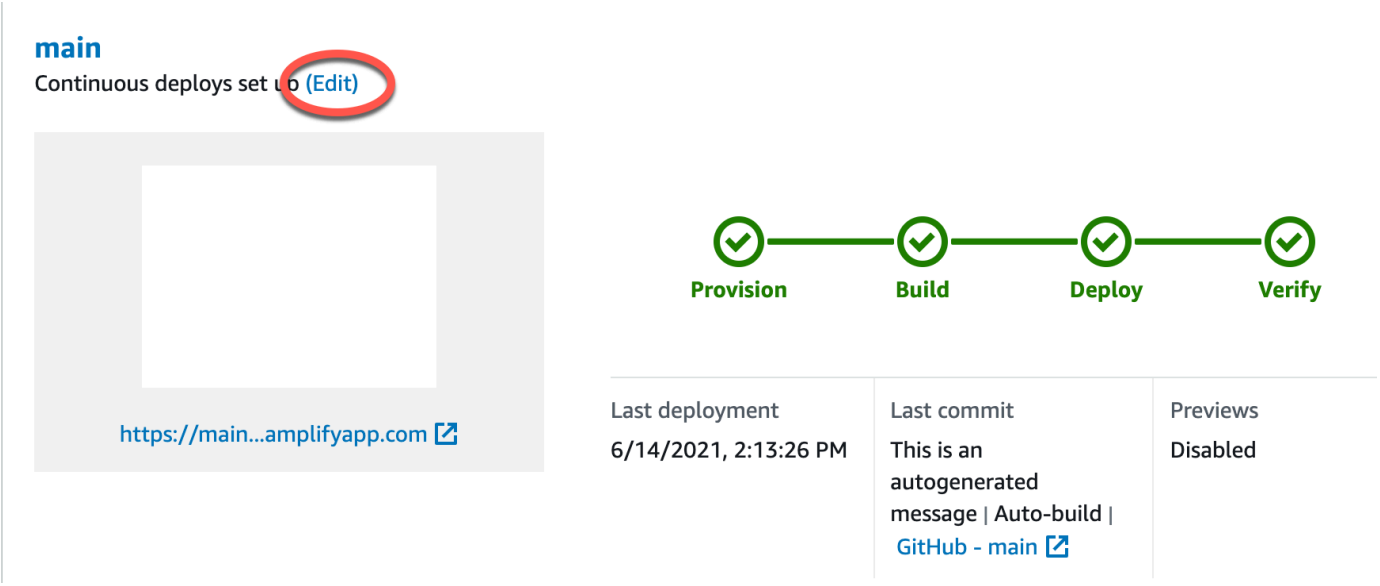
gerará automaticamente somente o arquivo `aws-exports.js`, sem modificar seu ambiente de back-end.

10. Escolha Próximo.
11. Escolha Salvar e implantar.

Edite um front-end existente para apontar para um back-end diferente

Para editar um aplicativo Amplify de front-end para apontar para um back-end diferente

1. Faça login AWS Management Console e abra o console do Amplify.
2. Escolha o aplicativo para o qual editar o back-end.
3. Escolha a guia Ambientes de hospedagem.
4. Localize a ramificação a ser editada e escolha Editar.



5. Na página Selecionar um ambiente de back-end para usar com essa ramificação, em Nome do aplicativo, selecione o aplicativo de front-end para o qual você deseja editar o ambiente de back-end. É possível escolher o aplicativo atual ou qualquer outro aplicativo na região atual.
6. Em Ambiente de back-end, selecione o nome do ambiente de back-end a ser adicionado.
7. Por padrão, a CI/CD de pilha completa está ativada. Desmarque essa opção para desativar o CI/CD de pilha completa para esse back-end. Desativar o CI/CD de pilha completa 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 back-end.
8. Escolha Salvar. O Amplify aplica essas alterações na próxima vez que você criar o aplicativo.

Construindo um back-end para um aplicativo

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 experiência de desenvolvedor TypeScript baseada em código que prioriza a definição de back-ends. Para saber como usar o Amplify Gen 2 para criar e conectar um back-end ao seu aplicativo, consulte Criar e conectar back-end nos documentos do Amplify.

Se você estiver procurando a documentação para criar um back-end para um aplicativo de primeira geração, usando a CLI e o Amplify Studio, consulte Criar e conectar back-end nos documentos do Amplify de primeira geração.

Tópicos

Crie um back-end para um aplicativo de 2ª geração

Para ver um tutorial que orienta você nas etapas de criação de um aplicativo fullstack do Amplify Gen 2 com um back-end TypeScript baseado, consulte Primeiros passos na documentação do Amplify.

Crie um back-end para um aplicativo de primeira geração

Neste tutorial, você configurará um fluxo de trabalho completo de CI/CD com o Amplify. Você implantará um aplicativo de front-end no Amplify Hosting. Em seguida, você criará um back-end usando o Amplify Studio. Por fim, você conectará o back-end da nuvem ao aplicativo de front-end.

Pré-requisitos

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

Inscreva-se para um Conta da AWS

Se você ainda não é AWS cliente, precisa criar um Conta da AWS seguindo as instruções on-line. A inscrição permite que você acesse o Amplify e outros AWS serviços que você pode usar com seu aplicativo.

Crie um repositório Git

O Amplify suporta GitHub Bitbucket e. GitLab AWS CodeCommit Envie seu aplicativo para o seu repositório Git.

Instale a interface de linha de comando (CLI) do Amplify

Para obter instruções, consulte Instalar a Amplify CLI na documentação do Amplify Framework.

Etapa 1: Implantar um front-end

Se você tiver um aplicativo de front-end existente em um repositório git que deseja usar neste exemplo, siga as instruções para implantar um aplicativo de front-end.

Se precisar criar um novo aplicativo de front-end para usar neste exemplo, siga as instruções do Create React App na documentação Create React App.

Para implantar um aplicativo de front-end

1. Faça login no AWS Management Console 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, consulteConfigurar o acesso do Amplify aos repositórios do 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 página Set-Cookie na Mozilla Developer Network.

Etapa 2: criar um back-end

Agora que você implantou um aplicativo de front-end no Amplify Hosting, é possível criar um back-end. Use as instruções a seguir para criar um back-end com um banco de dados simples e endpoint da API do GraphQL.

Para criar um back-end

1. Faça login no AWS Management Console 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 back-end 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 back-end de teste padrão no Amplify Studio.

O Amplify Studio é uma interface visual para criar e gerenciar seu back-end e acelerar o desenvolvimento de sua interface de usuário de front-end. Para obter mais informações sobre como usar o Amplify Studio, consulte a documentação do Amplify Studio.

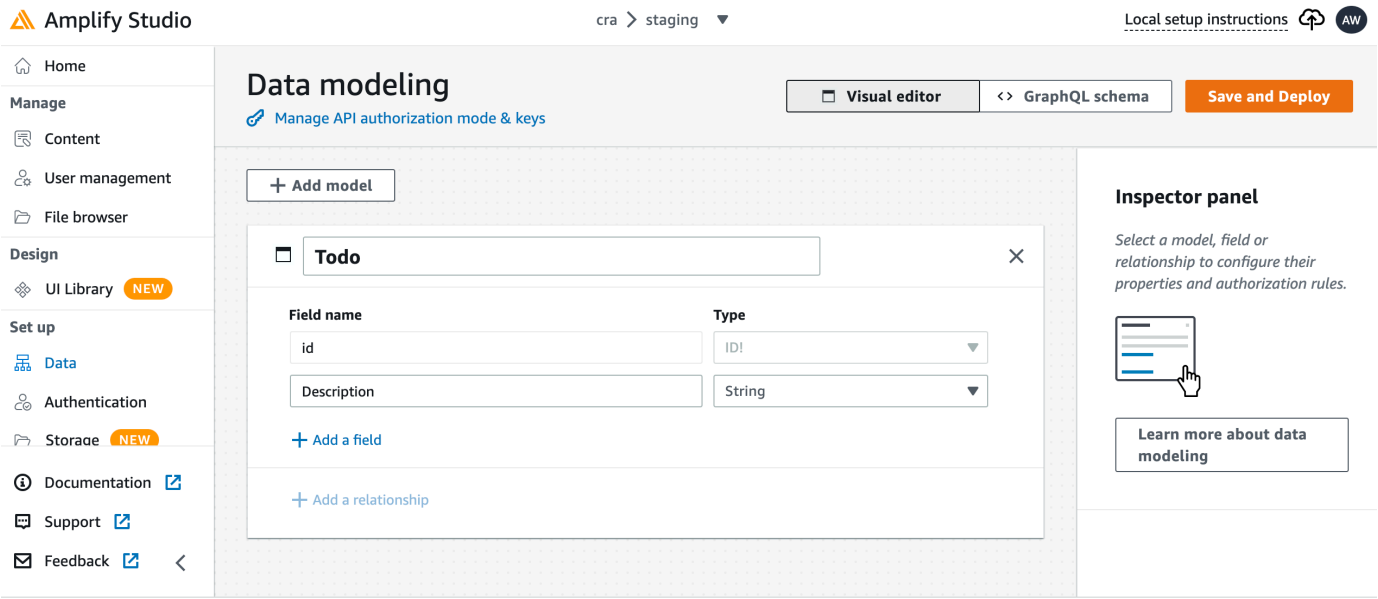
Use as instruções a seguir para criar um banco de dados simples usando a interface do construtor visual do Amplify Studio.

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. Amplify usa AWS CloudFormation para implantar seu back-end, o que permite que você armazene sua definição de back-end como. infrastructure-as-code

Etapa 3: Conectar o back-end ao front-end

Agora que você implantou um front-end e criou um back-end em nuvem que contém um modelo de dados, você precisa conectá-los. Use as instruções a seguir para reduzir sua definição de back-end ao seu projeto de aplicativo local com a Amplify CLI.

Para conectar um back-end de nuvem a um front-end 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 back-end do seu projeto.

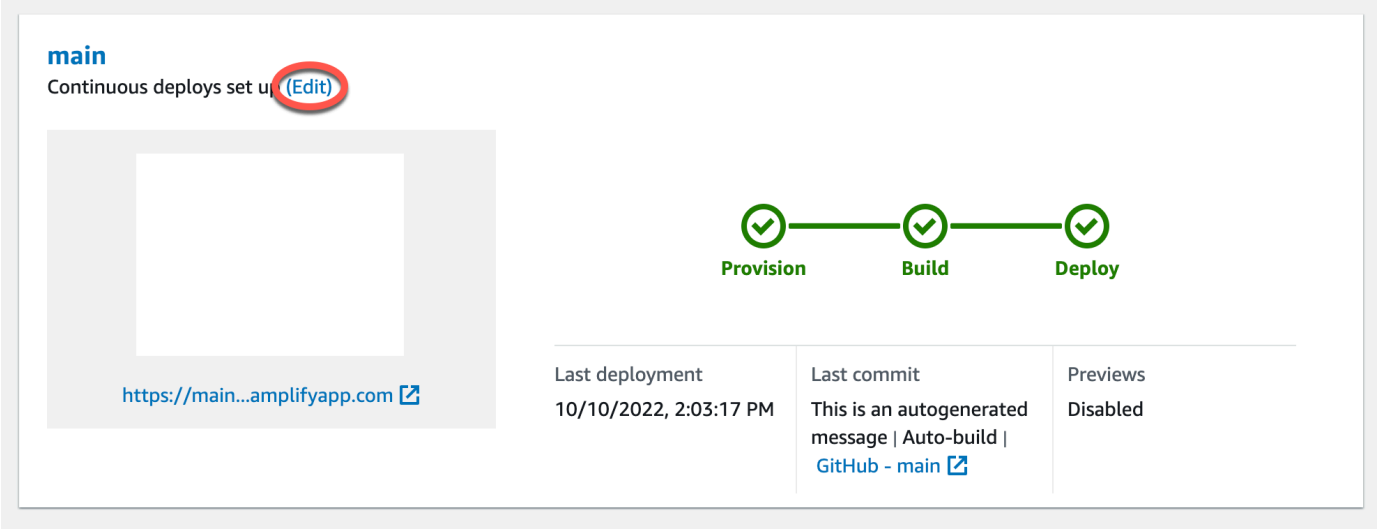
```
amplify pull --appId abcd1234 --envName staging
```

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 back-end ao fluxo de trabalho de implantação contínua. Use as instruções a seguir para conectar uma ramificação de front-end a um back-end no console do Amplify Hosting.

Para conectar uma ramificação de aplicativos de front-end e um back-end 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 back-end de destino, em Ambiente, selecione o nome do back-end a ser conectado. Neste exemplo, escolha o back-end de teste padrão que você criou na Etapa 2.

   Por padrão, a CI/CD de pilha completa está ativada. Desmarque essa opção para desativar o CI/CD de pilha completa para esse back-end. Desativar o CI/CD de pilha completa 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 back-end.

4. Em seguida, você deve configurar um perfil de serviço para dar ao Amplify as permissões necessárias para fazer alterações no back-end do seu aplicativo. É possível usar um perfil de serviço existente ou criar outro. Para obter instruções, consulte Adicionar um perfil de serviço.
5. Depois de adicionar um perfil de serviço, retorne à janela Editar back-end de destino e escolha Salvar.
6. Para concluir a conexão do back-end de teste padrão à ramificação principal do aplicativo de front-end, 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 back-end.

Crie uma interface de usuário de front-end no Amplify Studio

Use o Studio para criar sua interface de usuário de front-end com um conjunto de componentes de ready-to-use interface e, em seguida, conecte-a ao back-end do seu aplicativo. Para obter mais informações e tutoriais, consulte o guia do usuário do Amplify Studio na documentação do Amplify Framework.

Implantações manuais

As implantações manuais permitem publicar seu aplicativo web com o Amplify Hosting sem conectar um provedor do Git. É possível arrastar e soltar uma pasta 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.

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 ter mais informações, consulte Implante aplicativos renderizados do lado do servidor com o Amplify Hosting.

Implantação manual de arrastar e soltar

Para implantar manualmente um aplicativo usando arrastar e soltar

1. Faça login AWS Management Console e abra o console do Amplify.
2. No canto superior direito, escolha Criar novo aplicativo.
3. Na página Comece a criar com o Amplify, escolha Implantar sem Git. Em seguida, escolha Próximo.
4. Na seção Iniciar uma implantação manual, em Nome do aplicativo, insira o nome do seu aplicativo.
5. Em Nome da filial, 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 soltar ou use Escolher pasta.zip para selecionar o arquivo do seu computador. O arquivo que você arrasta e solta ou seleciona deve ser uma pasta zip que contenha o conteúdo da saída da compilação.
8. Escolha Salvar e implantar.

Implantação manual do Amazon S3 ou URL

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

1. Faça login AWS Management Console e abra o console do Amplify.
2. No canto superior direito, escolha Criar novo aplicativo.
3. Na página Comece a criar com o Amplify, escolha Implantar sem Git. Em seguida, escolha Próximo.
4. Na seção Iniciar uma implantação manual, em Nome do aplicativo, insira o nome do seu aplicativo.
5. Em Nome da filial, 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. Para Amazon S3 Bucket, selecione o nome do bucket Amazon S3 na lista. As listas de controle de acesso (ACLs) devem estar habilitadas para o bucket selecionado. Para ter mais informações, consulte Solução de problemas de acesso ao bucket do Amazon S3.
     b. Em Arquivo Zip, selecione o nome do arquivo zip a ser implantado.
   - Qualquer URL
     - Em URL do recurso, insira a URL do arquivo compactado 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

Ao criar um bucket do Amazon S3, você usa a configuração de Propriedade de Objeto do Amazon S3 para controlar se as listas de controle de acesso (ACLs) estão habilitadas ou desabilitadas para o bucket. Para implantar manualmente um aplicativo no Amplify a partir de um bucket do Amazon S3, as ACLs devem estar habilitadas no bucket.

Se você receber um erro `AccessControlList` ao implantar a partir de um bucket do Amazon S3, o bucket foi criado com as ACLs desativadas e você deve habilitá-las 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.

Implantar no botão Amplify

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:

Adicione o botão Implantar no Amplify Hosting ao seu 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>
```

Configurar o acesso do Amplify aos repositórios do GitHub

O Amplify agora usa o atributo Aplicativos GitHub para autorizar o acesso somente de leitura do Amplify aos repositórios do GitHub. Com o aplicativo Amplify GitHub, as permissões são mais refinadas, permitindo que você conceda ao Amplify acesso somente aos repositórios que você especificar. Para saber mais sobre os aplicativos do GitHub, consulte Sobre os aplicativos do GitHub no site do GitHub.

Quando você conecta um novo aplicativo armazenado em um repositório do GitHub, por padrão, o Amplify usa o aplicativo GitHub para acessar o repositório. No entanto, os aplicativos existentes do Amplify que você conectou anteriormente a partir dos repositórios do GitHub usam o OAuth para acesso. O CI/CD continuará funcionando para esses aplicativos, mas é altamente recomendável que você os migre para usar o novo aplicativo Amplify GitHub.

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 Amplify GitHub. 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`, substituindo *REGION* pela região em que você implantará seu aplicativo Amplify. Por exemplo, para instalar o aplicativo Amplify GitHub na região Oeste dos EUA (Oregon), acesse https://github.com/apps/aws-amplify-us-west-2.

Tópicos

- Instalando e autorizando o aplicativo Amplify GitHub para uma nova implantação
- Migrar um aplicativo OAuth existente para o aplicativo Amplify GitHub
- Configurar o aplicativo Amplify GitHub para implantações de AWS CloudFormation, CLI e SDK
- Configurar visualizações na web com o aplicativo Amplify GitHub

Instalando e autorizando o aplicativo Amplify GitHub para uma nova implantação

Ao implantar um novo aplicativo no Amplify a partir do código existente em um repositório do GitHub, use as instruções a seguir para instalar e autorizar o aplicativo GitHub.

Para instalar e autorizar o aplicativo Amplify GitHub

1. Faça login no AWS Management Console e abra o console do Amplify.
2. Na página Todos os aplicativos, escolha Novo aplicativo e, em seguida, Hospedar aplicativo web.
3. Na página Conceitos básicos do Amplify Hosting, escolha GitHub e, em seguida, escolha Continuar.
4. Se for a primeira vez que você conecta um repositório do GitHub, uma nova página é aberta no seu navegador em GitHub.com, solicitando permissão para autorizar na sua conta do GitHub AWS Amplify. Escolha Authorize.
5. Em seguida, você deve instalar o aplicativo Amplify GitHub na sua conta do GitHub. Uma página é aberta no GitHub.com solicitando permissão para instalar e autorizar na AWS Amplify sua conta do GitHub.
6. Selecione a conta do GitHub em que deseja instalar o aplicativo Amplify GitHub.
7. Faça um dos seguintes 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, selecione Próximo.
14. Na página Revisar, escolha Salvar e implantar.

Migrar um aplicativo OAuth existente para o aplicativo Amplify GitHub

Os aplicativos existentes do Amplify que você conectou anteriormente dos repositórios do GitHub usam o OAuth para acesso ao repositório. É altamente recomendável a migração desses aplicativos para usar o Amplify GitHub App.

Use as instruções a seguir para migrar um aplicativo e excluir o webhook OAuth correspondente na sua conta do GitHub. Observe que o procedimento de migração varia dependendo se o aplicativo Amplify GitHub já está instalado. Depois de migrar seu primeiro aplicativo e instalar e autorizar o aplicativo GitHub, você só precisa atualizar as permissões do repositório para migrações de aplicativos subsequentes.

Para migrar um aplicativo do OAuth para o aplicativo GitHub

1. Faça login no AWS Management Console 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 aplicativo GitHub e escolha Iniciar migração.
4. Na página Instalar e autorizar o aplicativo GitHub, escolha Configurar aplicativo GitHub.
5. Uma nova página é aberta no seu navegador em GitHub.com, solicitando permissão para autorizar AWS Amplify na sua conta do GitHub. Escolha Authorize.
6. Selecione a conta do GitHub em que deseja instalar o aplicativo Amplify GitHub.
7. Faça um dos seguintes 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 será redirecionado para a página Instalar e autorizar o aplicativo GitHub do seu aplicativo no console do Amplify. Se a autorização do GitHub 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.

Configurar o aplicativo Amplify GitHub para implantações de AWS CloudFormation, CLI e SDK

Os aplicativos existentes do Amplify que você conectou anteriormente dos repositórios do GitHub usam o OAuth para acesso ao repositório. Isso pode incluir aplicativos que você implantou usando a Command Line Interface (CLI) do Amplify, AWS CloudFormation ou os SDKs. É altamente recomendável a migração desses aplicativos para usar o novo Amplify GitHub App. A migração deve ser realizada no console do Amplify no AWS Management Console. Para obter instruções, consulte Migrar um aplicativo OAuth existente para o aplicativo Amplify GitHub.

É possível usar AWS CloudFormation, Amplify CLI e os SDKs para implantar um novo aplicativo Amplify que usa o aplicativo GitHub para acesso ao repositório. Esse processo exige que você primeiro instale o aplicativo Amplify GitHub na sua conta do GitHub. Em seguida, você precisará gerar um token de acesso pessoal na sua conta do GitHub. Por fim, implante o aplicativo e especifique o token de acesso pessoal.

Instale o aplicativo Amplify GitHub na sua conta

1. Abra um navegador da web e navegue até o local de instalação do aplicativo Amplify GitHub na região AWS em que você implantará seu aplicativo.

   Use o formato `https://github.com/apps/aws-amplify-REGION/installations/new`, substituindo *REGION* 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 conta do GitHub em que deseja instalar o aplicativo Amplify GitHub.
3. Faça um dos seguintes 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 Install.

Gere um token de acesso pessoal na sua conta do GitHub

1. Faça login em sua conta do GitHub.

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 do GitHub Apps, 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, AWS CloudFormation ou os SDKs.

Depois que o aplicativo Amplify GitHub for instalado na sua conta do GitHub e você tiver gerado um token de acesso pessoal, você poderá implantar um novo aplicativo com a Amplify CLI ou os SDKs AWS CloudFormation. Use o campo `accessToken` para especificar o token de acesso pessoal criado no procedimento anterior. Para obter mais informações, consulte CreateApp na referência da API do Amplify e AWS::Amplify::App no Guia do usuário do AWS CloudFormation.

O comando CLI a seguir implanta um novo aplicativo Amplify que usa o aplicativo GitHub para acesso ao repositório. Substitua *myapp-using-githubapp*, *https://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
```

Configurar visualizações na web com o aplicativo Amplify GitHub

Uma prévia na web implanta cada pull request (PR) feita no seu repositório do GitHub em uma URL de visualização exclusiva. As visualizações prévias agora usam o aplicativo Amplify GitHub para acessar seu repositório do GitHub. Para obter instruções sobre como instalar e autorizar o aplicativo GitHub para visualizações na web, consulte Ativar visualizações na web.

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 solicitações pull (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 aplicativos com ambientes de back-end provisionados usando a CLI do Amplify ou o Amplify Studio, cada pull request (somente repositórios Git privados) cria um back-end temporário que é excluído quando o PR é fechado.

Quando as visualizações na web são ativadas para seu aplicativo, cada PR conta para a cota do Amplify de 50 filiais por aplicativo. Para evitar exceder essa cota, certifique-se de fechar seus 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.

Ativar visualizações na web

Para aplicativos armazenados em um GitHub repositório, as visualizações 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 usando OAuth para acesso, primeiro você deve migrar o aplicativo para usar o aplicativo Amplify. GitHub Para obter instruções de migração, consulte Migrar um aplicativo OAuth existente para o aplicativo Amplify GitHub.

> **Important**
>
> Para fins de segurança, é possível ativar as visualizações da web em todos os aplicativos com repositórios privados, mas não em todos os aplicativos 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 back-ends 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.

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 Visualizações prévias, selecione uma ramificação na lista e escolha Editar configurações.
4. Na página Gerenciar configurações de visualização, ative as visualizações de pull request. Depois, selecione Confirm (Confirmar).
5. Para aplicativos de pilha completa, realize uma das seguintes ações:
   - Escolha Criar um novo ambiente de back-end 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 back-end 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 visualizações na web para pull requests podem ser acessadas com subdomínios para um aplicativo Amplify conectado 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 Configure subdomínios automáticos para um domínio personalizado do Amazon Route 53.

Adicione testes end-to-end Cypress ao 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.

Tutorial: Configurar end-to-end testes com o Cypress

O Cypress é uma estrutura de teste JavaScript baseada que permite executar testes E2E em um navegador. Para ver um tutorial que demonstra como configurar testes E2E, consulte a postagem do blog Executando testes end-to-end Cypress para sua implantação completa de CI/CD com o Amplify.

Adicione testes ao seu aplicativo 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.

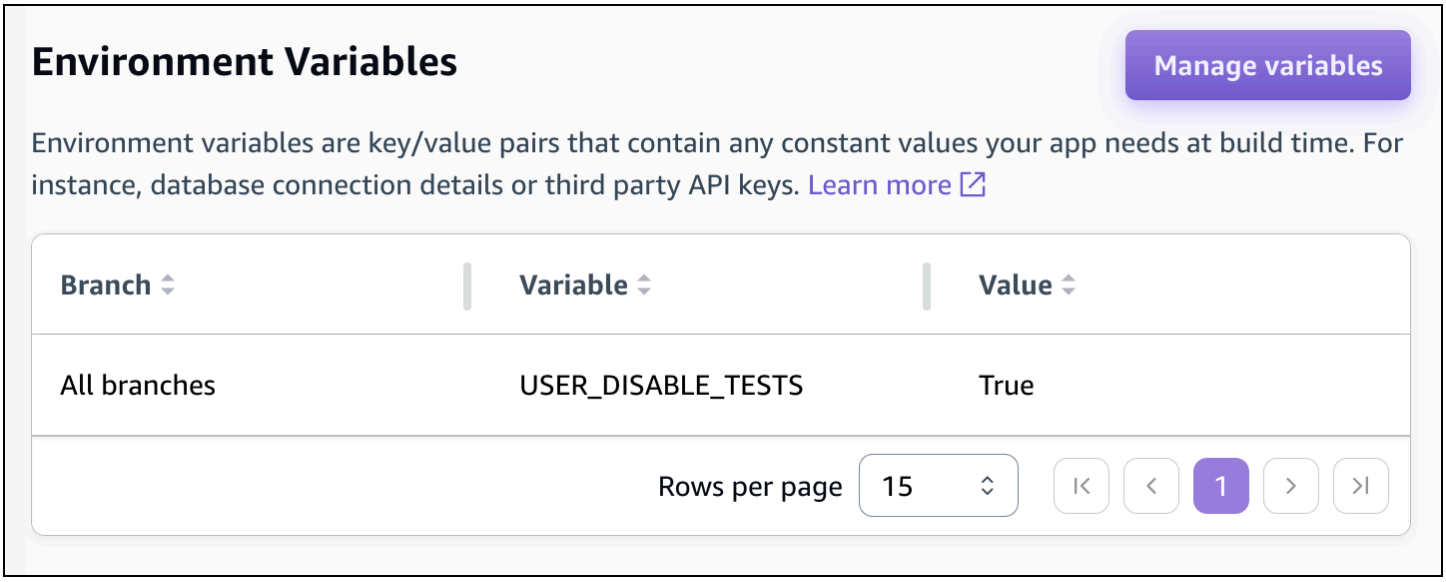
```
test:
  phases:
    preTest:
      commands:
        - npm ci
        - npm install -g pm2
        - npm install -g wait-on
        - npm install mocha mochawesome mochawesome-merge mochawesome-report-generator
        - pm2 start npm -- start
        - wait-on http://localhost:3000
    test:
      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'
```

Desabilitar testes

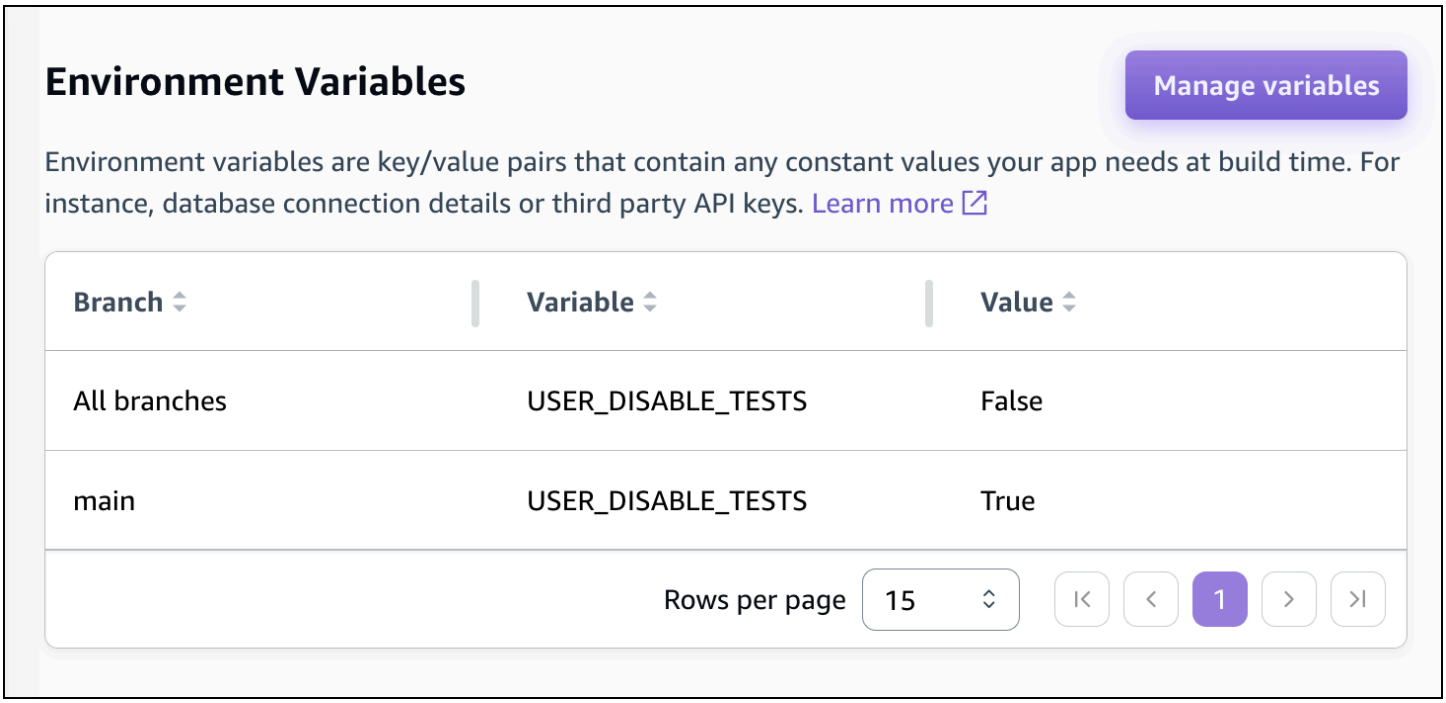
Depois que a configuração de teste for adicionada às suas configurações da compilação `amplify.yml`, a etapa `test` será executada em cada compilação, em cada ramificação. Se você quiser desabilitar globalmente a execução de testes ou executar testes apenas para ramificações específicas, é possível usar 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`.

Usar redirecionamentos

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.

Tipos de redirecionamentos

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 comparação entre A/B da interface de um 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.

Criando e editando redirecionamentos

É possível criar e editar redirecionamentos para um aplicativo 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 AWS Management Console 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. O procedimento para adicionar um redirecionamento varia dependendo se você deseja adicionar regras individualmente ou fazer uma edição em massa:
   - Para criar um redirecionamento individual, escolha Adicionar reescrita.
     a. Para um endereço original, insira o endereço que o usuário solicitou.
     b. Em Endereço de destino, insira o endereço de destino que renderiza o conteúdo para o usuário.
     c. Em Tipo, escolha o tipo de redirecionamento na lista.
     d. (Opcional) Em Código do país, insira uma condição de código do país de duas letras.
   - Para editar redirecionamentos em massa, escolha Abrir editor de texto.
     - Adicione ou atualize manualmente os redirecionamentos no editor JSON de regravações e redirecionamentos.
6. Escolha Salvar.

Ordem de redirecionamentos

Os redirecionamentos são aplicados do topo da lista para baixo. 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
```

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.

Redirecionamentos e regravações simples

Esta seção inclui exemplos de código para cenários comuns de redirecionamento.

> **Note**
>
> A correspondência de domínio do endereço original não diferencia maiúsculas de minúsculas.

É possível usar o código de 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)`	

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

É possível usar o código de 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)`	

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

É possível usar o código de exemplo a seguir para redirecionar todo o tráfego para index.html como uma regravação. 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)`	

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

É possível usar o código de exemplo a seguir para usar uma regravação 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)`	

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

É possível usar o código de 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)`	

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

É possível usar o código de 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)`	

JSON [{"source": "/<*>", "status": "404", "target": "/404.html", "condition": null}]

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 redirecionamento	Código do país
`</^[^.]+$\|\.(?!(css\|gif\|ico\|jpg\|js\|png\|txt\|svg\|woff\|woff2\|ttf\|map\|json\|webp)$)([^.]+$)/>`	`/index.html`	`200`	

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.

Endereço original	Endereço de destino	Tipo de redirecionamento	Código do país
`/images/<*>`	`https://images.otherdomain.com/<*>`	`rewrite (200)`	

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

Barras finais e URLs limpos

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 URLs limpos 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 código de 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)`	

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

Strings de consulta e parâmetros de caminho

É possível usar o código de 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 redirecionamento	Código do país
`/docs?id=<my-blog-id-value`	`/documents/<my-blog-post-id-value>`	`permanent redirect (301)`	

JSON [{"source": "/docs?id=<my-blog-id-value>", "status": "301", "target": "/documents/<my-blog-id-value>", "condition": null}]

> **Note**
>
> 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 código de 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)`	

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

Redirecionamentos baseados em região

É possível usar o código de 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>

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

Expressões curinga em redirecionamentos e regravações

Você pode usar a expressão curinga,<*>, no endereço original para redirecionar ou reescrever. Você deve colocar a expressão no final do endereço original e ela deve ser exclusiva. O Amplify ignora os endereços originais que incluem 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)`	

Endereço original	Endereço de destino	Tipo de redirecio namento	Código do país
`/docs/<*>/content/<*>`	`/documents/<*>/content/<*>`	`permanent redirect (301)`	

Restringir o acesso a ramificações

Se você estiver trabalhando em recursos não lançados, poderá proteger com senha as ramificações de recursos para restringir o acesso a usuários específicos. 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.

Você pode definir uma senha que se aplique a uma filial individual ou globalmente a todas as filiais conectadas. Quando o controle de acesso está habilitado no nível global e na filial, a senha no nível da filial tem precedência sobre uma senha no nível global (aplicativo).

Para definir senhas em ramificações de atributos

1. Faça login AWS Management Console 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, em seguida, escolha Controle de acesso.
4. Na seção Configurações de controle de acesso, escolha Gerenciar acesso.
5. Na página Gerenciar controle de acesso, faça o seguinte.
   - Para definir um nome de usuário e uma senha que se apliquem a todas as filiais conectadas
     - Ative Gerenciar acesso para todas as filiais. Por exemplo, se você tiver ramificações principal, de desenvolvimento e de recursos conectadas, poderá aplicar o mesmo nome de usuário e senha para todas as ramificações.
   - Para aplicar um nome de usuário e senha a uma filial individual
     a. Desative Gerenciar acesso para todas as filiais.
     b. Localize a filial que você deseja gerenciar. Para 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.

Variáveis de ambiente

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 `AWS` prefixo. Esse prefixo é 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 um aplicativo de 2ª geração, use o recurso de 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 ter mais informações, consulte Gerenciando segredos do ambiente.

Variáveis de ambiente do Amplify

As seguintes variáveis de ambiente podem ser acessadas por padrão no console do Amplify.

Nome da variável	Descrição	Valor de exemplo
_BUILD_TIMEOUT	A duração do tempo limite de compilação em minutos	`30`
_LIVE_UPDATES	A ferramenta será atualizada para a versão mais recente.	`[{"name":"Amplify CLI","pkg":"@aws-amplify/cli","type"`

Nome da variável	Descrição	Valor de exemplo
		`:"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 Desabilitar testes.	`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-2:1234567890 12:appname/branch/ ...`
AWS_CLONE_URL	O URL de clone usado para buscar o conteúdo do repositório Git	`git@github.com:<us er-name>/<repo-nam e>.git`
AWS_COMMIT_ID	O ID de confirmação da compilação atual “HEAD” para recompilações	`abcd1234`

Nome da variável	Descrição	Valor de exemplo
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 do pull request da compilação de pré-visualização da web do pull request. 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 prévia de pull request enviada para uma ramificação do aplicativo no console do Amplify.	`featureA`
AWS_PULL_REQUEST_DESTINATION_BRANCH	O nome da ramificação do aplicativo no console do Amplify para a qual uma pull request da ramificação de recursos está sendo enviada.	`main`
AMPLIFY_AMAZON_CLIENT_ID	O ID do cliente da Amazon	`123456`
AMPLIFY_AMAZON_CLIENT_SECRET	O segredo do cliente da Amazon	`example123456`
AMPLIFY_FACEBOOK_CLIENT_ID	O ID do cliente do Facebook	`123456`

Nome da variável	Descrição	Valor de exemplo
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 front-end baseada em diff. Para ter mais informações, consulte Ativar ou desativar a compilação e implantação de front-end com base em diff.	`true`
AMPLIFY_DIFF_DEPLOY_ROOT	O caminho a ser usado para comparações de implantação de front-end baseadas em diff, em relação à raiz do seu repositório.	`dist`
AMPLIFY_DIFF_BACKEND	Ative ou desative as compilações de back-end baseadas em diff. Isso se aplica somente aos aplicativos de primeira geração. Para obter mais informações, consulte Ativar ou desativar compilações de back-end baseadas em diferenças para um aplicativo de primeira geração.	`true`

Nome da variável	Descrição	Valor de exemplo
AMPLIFY_BACKEND_PULL_ONLY	O Amplify gerencia essa variável de ambiente. Isso se aplica somente aos aplicativos de primeira geração. Para obter mais informações, consulte Edite um front-end existente para apontar para um back-end diferente.	`true`
AMPLIFY_BACKEND_APP_ID	O Amplify gerencia essa variável de ambiente. Isso se aplica somente aos aplicativos de primeira geração. Para obter mais informações, consulte Edite um front-end existente para apontar para um back-end diferente.	`abcd1234`
AMPLIFY_SKIP_BACKEND_BUILD	Se você não tiver uma seção de back-end em sua especificação de compilação e quiser desativar as compilações de back-end, defina essa variável de ambiente como `true`. Isso se aplica somente aos aplicativos de primeira geração.	`true`
AMPLIFY_ENABLE_DEBUG_OUTPUT	Defina essa variável `true` para imprimir um rastreamento de pilha nos registros. Isso é útil para depurar erros de compilação de back-end.	`true`

Nome da variável	Descrição	Valor de exemplo
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`
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`

Nome da variável	Descrição	Valor de exemplo
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. Para obter mais informações, consulte Limite de permissões IAM do perfis gerados 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 tokens OAuth, não uma chave de AWS acesso e uma chave secreta.

Definição de variáveis de ambiente

Use as instruções a seguir para definir variáveis de ambiente para um aplicativo 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 AWS Management Console 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.

Acesse variáveis de ambiente no momento da compilação

Para acessar uma variável de ambiente durante uma compilação, edite as configurações da compilação para incluir a variável de ambiente nos comandos de compilação.

Cada comando em sua configuração de compilação é executado dentro de um shell Bash. Para obter mais informações sobre como trabalhar com variáveis de ambiente no Bash, consulte Expansões do Shell do Manual do GNU Bash.

Para editar as configurações da compilação para incluir uma variável de ambiente

1. Faça login no AWS Management Console e abra o console do Amplify.
2. No console do Amplify, escolha Hospedagem e, em seguida, selecione Configurações de criação.
3. Na seção Especificação de compilação do aplicativo, escolha Editar.
4. Adicione a variável de ambiente ao comando de compilação. Agora você deve ser capaz de acessar a variável de ambiente durante a próxima compilação. Este exemplo altera o

comportamento do npm (BUILD_ENV) e adiciona um token de API (TWITCH_CLIENT_ID) para um serviço externo a um arquivo de ambiente para uso posterior.

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

5. Escolha Salvar.

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

Por padrão, um componente do servidor Next.js não tem acesso às variáveis de ambiente do seu aplicativo. 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 ao Next.js, você deve modificar o arquivo de especificação de compilação do Amplify para definir as variáveis de ambiente nos arquivos de ambiente que o Next.js reconhece. Isso permite que o Amplify carregue as variáveis de ambiente antes de compilar o aplicativo. Para obter mais informações sobre como modificar sua especificação de compilação, veja exemplos de como adicionar variáveis de ambiente na seção de comandos de compilação.

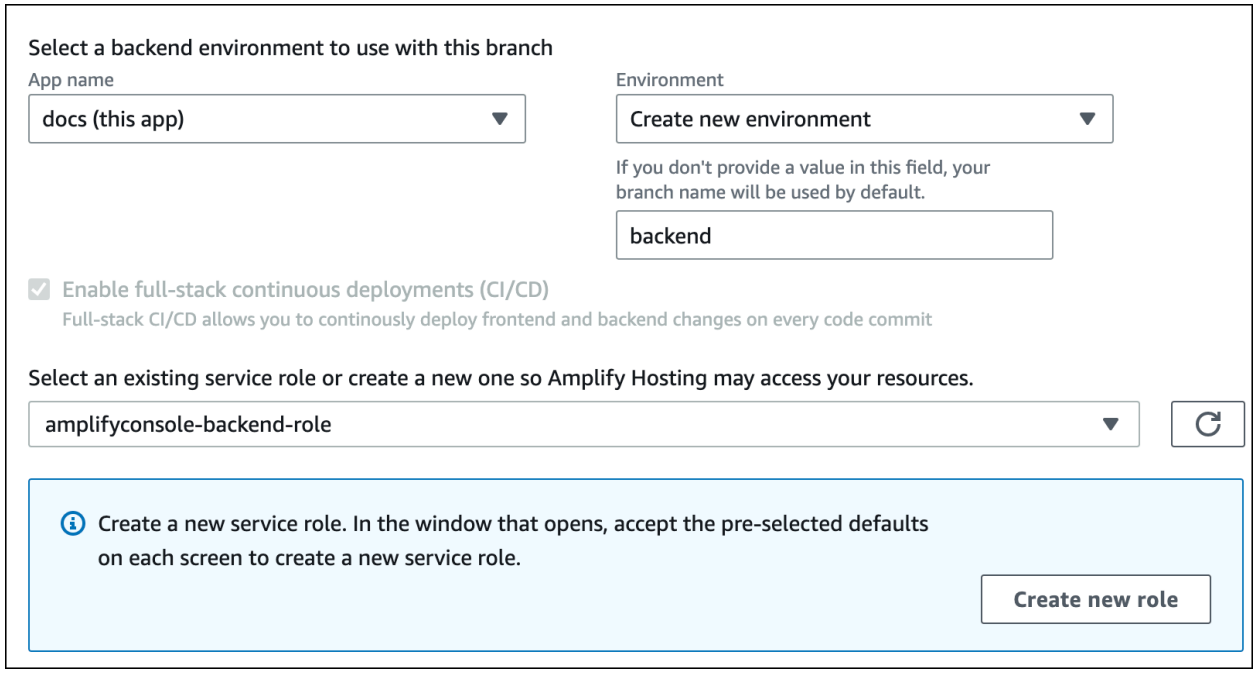
Crie um novo ambiente de back-end com parâmetros de autenticação para login social

Para conectar uma ramificação a um aplicativo

1. Faça login no AWS Management Console 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 back-end para usar com esta ramificação. Em Ambiente, escolha Criar novo ambiente e insira o nome do seu ambiente de back-end. A captura de tela a seguir mostra a

seção Selecionar um ambiente de back-end para usar com esta ramificação da página Configurações de compilação com a inserção do nome **backend** do ambiente de back-end.



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 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.

Variáveis de ambiente da estrutura de front-end

Se você estiver desenvolvendo seu aplicativo com uma estrutura de front-end 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 front-end. Para entender os efeitos do uso dessas variáveis de ambiente para armazenar valores, consulte a documentação da estrutura de front-end que você está usando.

Armazenar valores confidenciais, como chaves de API, dentro dessas variáveis de ambiente prefixadas da estrutura de front-end não é uma prática recomendada e é altamente desencorajado. Para obter um exemplo de uso das variáveis de ambiente de tempo de compilação do Amplify para essa finalidade, consulte Acesse variáveis de ambiente no momento da compilação.

Gerenciando segredos do ambiente

Com o lançamento do Amplify Gen 2, o fluxo de trabalho para 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 um aplicativo Amplify Gen 2, consulte Segredos e variáveis de ambiente na documentação do Amplify.

Os segredos de ambiente de um aplicativo de primeira geração 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.

Defina e acesse segredos do ambiente para um aplicativo de primeira geração

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 AWS Management Console 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.

Acessar segredos de ambiente

Acessar um segredo de ambiente para um aplicativo de primeira geração durante uma compilação é semelhante ao acesso a variáveis de ambiente, exceto que os segredos do ambiente são armazenados `process.env.secrets` como uma string JSON.

Segredos do 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	Descrição	Valor de exemplo
AMPLIFY_SIWA_CLIENT_ID	O login com o ID do cliente da Apple	`com.yourapp.auth`

Nome da variável	Descrição	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-----

Cabeçalhos personalizados

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. Você pode especificar cabeçalhos no console do Amplify ou baixando e editando o arquivo `customHttp.yml` de um aplicativo 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 um aplicativo editando a especificação de compilação (buildspec) no AWS Management Console ou baixando e atualizando o `amplify.yml` arquivo e salvando-o no diretório raiz do projeto. Os cabeçalhos personalizados especificados dessa forma devem ser migrados do buildspec e do arquivo `amplify.yml`. Para obter instruções, consulte Migração de cabeçalhos personalizados.

Formato 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 personalizados são aplicados a todos os caminhos de arquivo de URL que correspondem ao padrão.

headers

Defina cabeçalhos que correspondem ao padrão de arquivo.

chave

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 maneiras de especificar cabeçalhos HTTP personalizados para um aplicativo Amplify. Você pode especificar cabeçalhos no console do Amplify ou pode especificar cabeçalhos baixando e editando o arquivo de um aplicativo e salvando-o no diretório raiz `customHttp.yml` do seu projeto.

Para definir cabeçalhos personalizados para um aplicativo e salvá-los no console

1. Faça login AWS Management Console 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 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. Escolha Salvar.
7. Reimplante o aplicativo para aplicar os novos cabeçalhos personalizados.
   - Para um aplicativo CI/CD, navegue até a filial 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 um aplicativo e salvá-los na raiz do seu repositório

1. Faça login AWS Management Console 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 aplicativo de CI/CD, 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 `customHttp.yml` arquivo e implantados no diretório raiz do aplicativo substituem os cabeçalhos personalizados definidos na seção Cabeçalhos personalizados no console do Amplify.

Migração de cabeçalhos personalizados

Anteriormente, cabeçalhos HTTP personalizados eram especificados para um aplicativo editando o buildspec no console do Amplify ou baixando e atualizando o `amplify.yml` arquivo e salvando-o no diretório raiz do projeto. É altamente recomendável que você migre seus cabeçalhos personalizados do buildspec 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 AWS Management Console e abra o console do Amplify.
2. Escolha o aplicativo no qual realizar a migração personalizada do cabeçalho.
3. No painel de navegação, escolha Hosting, Build settings. 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 AWS Management Console 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 aplicativo de CI/CD, 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 `customHttp.yml` arquivo e implantados no diretório raiz do aplicativo substituem os cabeçalhos personalizados definidos na seção Cabeçalhos personalizados do console do Amplify.

Cabeçalhos personalizados 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 Formato YAML de cabeçalho personalizado.

- Você pode especificar cabeçalhos personalizados para um aplicativo em um monorepo usando a seção Cabeçalhos personalizados do console do Amplify. Você deve reimplantar seu aplicativo 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 `customHttp.yml` arquivo substituem quaisquer cabeçalhos personalizados especificados usando a seção Cabeçalhos personalizados do console do Amplify.

Exemplo de cabeçalhos 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'"
```

Cabeçalhos de controle de cache personalizados

Os aplicativos hospedados com o Amplify respeitam `Cache-Control` os cabeçalhos enviados pela origem, a menos que você os substitua por cabeçalhos personalizados definidos por você. O Amplify aplica somente cabeçalhos personalizados do Cache-Control para respostas bem-sucedidas 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 Usar cabeçalhos para controlar a duração do cache.

Webhooks recebidos

Configure um webhook de entrada no console do Amplify para iniciar uma compilação sem comprometer o código no seu repositório Git. É possível usar gatilhos de webhook com ferramentas CMS headless (como Contentful ou 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 no AWS Management Console 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 criação.
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 sem cabeçalho ou outro serviço para iniciar as compilações.
    - Execute o comando curl em uma janela de terminal para iniciar uma nova compilação.

Monitorar

AWS Amplify emite métricas pela Amazon CloudWatch e fornece registros de acesso com informações detalhadas sobre solicitações feitas ao seu aplicativo. Use os tópicos desta seção para aprender a usar essas métricas e logs para monitorar seu aplicativo.

Tópicos

Monitoramento com CloudWatch

AWS Amplify é integrado à Amazon CloudWatch, permitindo que você monitore métricas para seus aplicativos Amplify quase em tempo real. É possível criar 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.

Metrics

O Amplify suporta seis CloudWatch métricas no `AWS/AmplifyHosting` namespace para monitorar tráfego, erros, transferência de dados e latência de 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.

Nem todas as estatísticas são aplicáveis a todas as métricas. Na tabela a seguir, as estatísticas mais relevantes estão listadas na descrição de cada métrica.

Metrics	Descrição
Solicitações	O número total de solicitações de visualiza dores recebidas pelo seu aplicativo. A estatística mais relevante é `Sum`. Use a estatística `Sum` para obter o número total de solicitações.

Metrics	Descrição
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 seu aplicativo (enviados) em bytes usando as solicitações `POST` e `PUT`. 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 ocorrênci as desses erros.
5XXErrors	O número de solicitações que retornaram um erro no intervalo do código de status HTTP 500-599. A estatística mais relevante é `Sum`. Use a estatística `Sum` para obter o total de ocorrênci as desses erros.

Metrics	Descrição
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.

O Amplify fornece as seguintes dimensões CloudWatch métricas.

Dimensão	Descrição
App	Os dados métricos são fornecidos pelo aplicativo.
Conta da AWS	Os dados métricos são fornecidos em todos os aplicativos no Conta da AWS.

Você pode acessar CloudWatch as métricas AWS Management Console em https://console.aws.amazon.com/cloudwatch/. Alternativamente, é possível acessar métricas no console do Amplify usando o procedimento a seguir.

Para acessar métricas usando o console do Amplify

1. Faça login no AWS Management Console e abra o console do Amplify.
2. Escolha o serviço do qual você deseja visualizar as métricas.
3. No painel de navegação, escolha Configurações do aplicativo, Monitoramento.

4. Na página Monitoramento, escolha Métricas.

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 as 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 AWS Management Console e abra o console do Amplify.
2. Escolha o aplicativo no qual deseja definir um alarme.
3. No painel de navegação, escolha Configurações do aplicativo, Monitoramento.
4. Na página Monitoramento, escolha Alarmes.
5. Selecione Criar alarme.
6. 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 consecutivos que devem estar no estado de alarme para invocar o alarme.
iv. Especifique a duração do período de avaliação.

e. Selecione Criar alarme.

> **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.

Amazon CloudWatch Logs para aplicativos SSR

Amplify envia informações sobre seu tempo de execução do Next.js 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 Adicionar um perfil de serviço. Para mais informações sobre como implantar aplicativos renderizados no lado do servidor, consulte Implante aplicativos renderizados do lado do servidor com o Amplify Hosting.

Logs de acesso

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 registros de acesso de um aplicativo até que você exclua o aplicativo. Todos os registros de acesso de um aplicativo estão disponíveis no console do Amplify. No entanto, cada solicitação individual de registros de acesso é limitada a um período de duas semanas especificado por você.

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.

Use o procedimento a seguir para recuperar os registros de acesso de um aplicativo.

Para visualizar logs de acesso

1. Faça login no AWS Management Console e abra o console do Amplify.

2. Escolha o aplicativo para o qual deseja acessar os logs.
3. No painel de navegação, escolha Hospedagem e, em seguida, Monitoramento.
4. Na página Monitoramento, escolha Logs de acesso.
5. Escolha Editar intervalo de tempo.
6. 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 registros.
    b. Em Hora de início, escolha a hora do primeiro dia para iniciar a recuperação do log.
    c. Selecione a opção Confirmar.
7. 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;
```

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.

Configurar 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 no AWS Management Console 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, Criar notificações. Na página Criar notificações, escolha Gerenciar notificações.
4. Na página Gerenciar notificações, escolha Adicionar novo.
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.

Imagens de compilação personalizada e atualizações de pacote ao vivo

Tópicos

Imagens de compilação personalizada

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.

> **Note**
>
> As configurações de compilação ficam visíveis no menu Hospedagem do console do 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.

Requisitos de imagens de compilação personalizada

Para que uma imagem de compilação personalizada funcione como uma imagem de compilação do Amplify, ela deve atender aos seguintes requisitos:

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 pacote com extensão Hugo 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 pacote do Amplify usa NVM para alterar as versões do Node.js 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 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 no AWS Management Console 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 Hosting, Build settings.
5. Na página Configurações da compilação, na seção Configurações da compilação de imagem, escolha Editar.
6. Na página Editar configurações de imagem de compilação, expanda o menu Criar imagem 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.

Atualizações de pacote ao vivo

As atualizações de pacote ao vivo possibilitam que você especifique versões de pacotes e dependências para uso em nossa imagem de compilação padrão 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**
>
> A definição da versão do Node.js como a mais recente fará com que as compilações falhem. Em vez disso, você deverá especificar uma versão exata do Node.js, como `18`, `21.5` ou `v0.1.2`.

Configuração de atualizações de pacote ao vivo

Para configurar atualizações de pacote ao vivo

1. Faça login no AWS Management Console 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 Hosting, Build settings.
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 dinâmicos, escolha Adicionar novo.
6. Em Package, 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.

Adicionar um perfil de serviço

O Amplify exige permissões para implantar recursos do back-end com o front-end. Você usa um perfil de serviço para fazer isso. 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. Neste guia, você aprenderá como criar uma função de serviço do Amplify que tenha permissões administrativas da conta e permita explicitamente acesso direto aos recursos que os aplicativos do Amplify exigem para implantar, criar e gerenciar back-ends.

Criar um perfil de serviço

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 Amplificar e, em seguida, escolha Avançar.
3. Na página Adicionar permissões, escolha Próximo.
4. Na página Nome, visualização e criação, em Nome da função, insira um nome significativo, como**`AmplifyConsoleServiceRole-AmplifyRole`**.
5. Aceite todos os padrões e escolha Criar função.
6. Volte ao console do Amplify para anexar a função ao seu aplicativo.
   - Se você estiver no processo de implantação de um novo aplicativo
     a. Atualize a lista de funções de serviço.
     b. Selecione a função que você acabou de criar. Para este exemplo, deve ser semelhante a AmplifyConsoleServiceRole- AmplifyRole
     c. Escolha Avançar e siga as etapas para concluir a implantação do aplicativo.
   - Se você tiver um aplicativo existente
     a. No painel de navegação, escolha Configurações do aplicativo e, em seguida, Configurações gerais.
     b. Na página Configurações gerais, escolha Editar.
     c. Na página Editar configurações gerais, selecione a função que você acabou de criar na lista de funções de serviço.

d. Escolha Salvar.

7. O console do Amplify agora tem permissões para implantar recursos de back-end para seu aplicativo.

Prevenção do problema do “confused deputy”

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 ter mais informações, consulte Prevenção do problema do substituto confuso entre serviços.

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 uma função `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 do aplicativo 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 AWS Management Console, consulte Modificar uma função (console) no Guia do usuário do IAM.

Gerenciar desempenho do aplicativo

A arquitetura de hospedagem padrão do Amplify otimiza o equilíbrio entre desempenho de hospedagem e disponibilidade de implantação. Para a maioria dos clientes, recomendamos que você use a arquitetura padrão.

Se você precisar de um controle mais preciso sobre o desempenho de um aplicativo, poderá definir manualmente o `Cache-Control` cabeçalho HTTP para otimizar o desempenho da hospedagem, mantendo o conteúdo em cache na borda da rede de distribuição de conteúdo (CDN) por um intervalo maior.

Usar cabeçalhos para controlar a duração do cache

Os `Cache-Control` cabeçalhos `max-age` e as `s-maxage` diretivas HTTP afetam a duração do armazenamento em cache do conteúdo do seu aplicativo. 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.

Os aplicativos hospedados com o Amplify respeitam `Cache-Control` os cabeçalhos 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 bem-sucedidas com um `200 OK` código de status. 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).

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 Cabeçalhos de controle de cache personalizados.

Configurando o Cache-Control cabeçalho para aumentar o desempenho do aplicativo

Use o procedimento a seguir para definir a `s-maxage` diretiva para manter o conteúdo em cache na borda da CDN por 24 horas.

Para definir um Cache-Control cabeçalho personalizado

1. Faça login no AWS Management Console 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 do cabeçalho personalizado da seguinte forma:
   a. Para `pattern`, insira **`**/*`** para todos os caminhos.
   b. Em `key`, digite **`Cache-Control`**.
   c. Em `value`, digite **`s-maxage=86400`**.
6. Escolha Salvar.
7. Reimplante o aplicativo para aplicar o novo cabeçalho personalizado.

Log de chamadas de API do Amplify usando o 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. Você pode 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, ao criar uma trilha no console, ela 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:

- 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"
}
```

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. Auditores terceirizados 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, dependendo do trabalho que você faz no Amplify.

Usuário do serviço: se você usar o serviço Amplify para fazer o trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que mais recursos do Amplify forem usados para realizar seu trabalho, talvez sejam necessárias permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um atributo no Amplify, consulte Solução de problemas de identidade e acesso do Amplify.

Administrador do serviço: se você for o responsável pelos recursos do Amplify em sua empresa, provavelmente terá acesso total ao Amplify. Cabe a você determinar quais funcionalidades e recursos do Amplify os usuários do serviço deverão acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como a empresa pode usar o IAM com o Amplify, consulte Como o Amplify funciona com o IAM.

Administrador do IAM: se você é um administrador do IAM, talvez queira saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao Amplify. Para visualizar exemplos de políticas

baseadas em identidade do Amplify que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade para o Amplify.

Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação Multifator no Guia do Usuário do AWS IAM Identity Center . Usar a autenticação multifator (MFA) na AWS no Guia do Usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais

do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do Usuário do IAM.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, 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 de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte O que é o Centro de Identidade do IAM? no Manual do Usuário do AWS IAM Identity Center .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários

têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de um perfil) no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma função do IAM no AWS Management Console trocando de funções. Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para o uso de perfis, consulte Utilizar perfis do IAM no Guia do usuário do IAM.

Funções do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidades de terceiros no Guia do Usuário do IAM. Se você usar o Centro de identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Manual do Usuário do AWS IAM Identity Center .
- Permissões temporárias para usuários do IAM –- um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas –- é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). 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.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.

- Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.
- Função de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

- Aplicativos em execução no Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo AWS CLI solicitações de API. AWS É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Utilizar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar perfis do IAM, consulte Quando criar um perfil do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida

ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do Usuário do IAM.

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.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissões para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem presumir os perfis.

As políticas do IAM definem permissões para uma ação independente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criando políticas do IAM no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do Usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas 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 que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para

o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

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 as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um atributo avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do Usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em. AWS Organizations AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada

uma Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte Como os SCPs Funcionam no Manual do Usuário do AWS Organizations .

- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, 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

Atributo do IAM	Suporte do Amplify
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações das políticas	Sim
Atributos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não

Atributo do IAM	Suporte do Amplify
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Sessões de acesso direto (FAS)	Sim
Perfis de serviço	Sim
Perfis vinculados ao serviço	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.

Políticas baseadas em identidade do Amplify

Suporta políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criando políticas do IAM no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da política JSON do IAM no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amplify

Para visualizar exemplos de políticas baseadas em identidade do Amplify, consulte Exemplos de políticas baseadas em identidade para o Amplify.

Políticas baseadas em recursos no Amplify

Oferece compatibilidade com políticas baseadas em recursos	Não

Políticas baseadas em recursos são documentos de políticas 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 que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve 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, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações políticas para Amplify

Oferece compatibilidade 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 você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de

AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

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 instrução, separe-as com vírgulas.

```
"Action": [
      "amplify:action1",
      "amplify:action2"
         ]
```

Para visualizar exemplos de políticas baseadas em identidade do Amplify, consulte Exemplos de políticas baseadas em identidade para o Amplify.

Recursos de políticas para Amplify

Oferece compatibilidade 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. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, 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 Exemplos de políticas baseadas em identidade para o Amplify.

Chaves de condição de política para o Amplify

Suporta 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` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. 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.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. 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 Exemplos de políticas baseadas em identidade para o Amplify.

Listas de controle de acesso (ACLs) no Amplify

Oferece compatibilidade 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 as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Controle de acesso baseado em atributos (ABAC) com o Amplify

Oferece compatibilidade com ABAC (tags em políticas)	Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações onde o gerenciamento de políticas se torna um problema.

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 oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do Usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Utilizar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Amplify

Oferece compatibilidade com credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS “Trabalhe com o IAM” no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte Alternar para um perfil (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Sessões de acesso direto para o Amplify

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado um principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um

serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Perfis de serviço do Amplify

Oferece compatibilidade com funções de serviço	Sim

O perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, 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.

Perfis vinculados ao serviço para o Amplify

Oferece suporte a 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 presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas 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 Service-linked role (Perfil vinculado ao serviço). Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para o Amplify

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do Amplify. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder aos usuários permissão para executar ações nos recursos de que precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem presumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte Criação de políticas do IAM 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

- Melhores práticas de política
- Usando o console do Amplify
- Permitir que usuários visualizem suas próprias permissões

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amplify em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance 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 obter mais informações, 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 atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações 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 –- você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode gravar 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 AWS CloudFormation. Para obter mais informações, 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 100 verificações de política e recomendações acionáveis para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, 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 obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações 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 Se você criar uma política baseada em identidade que seja mais restritiva do 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 a operação de API que estiverem tentando executar.

Com o lançamento do Amplify Studio, a exclusão de um aplicativo ou back-end 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 está 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 mais informações, consulte Adicionando Permissões a um Usuário no Guia do Usuário do IAM.

Permitir que 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 políticas gerenciadas pelo cliente da 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) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

Política gerenciada pela AWS: AdministratorAccess -Amplify

É possível anexar a política `AdministratorAccess-Amplify` a 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 back-ends.

Detalhes da permissão

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 ver as permissões dessa política, consulte AdministratorAccess-Amplify na Referência de política AWS gerenciada.

AWS política gerenciada: AmplifyBackendDeployFullAccess

É possível anexar a política `AmplifyBackendDeployFullAccess` a 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 da permissão

Essa política inclui permissões para fazer o seguinte.

- `Amplify`— Recupere metadados sobre aplicativos implantados.
- `AWS CloudFormation`— Crie, atualize e exclua as pilhas gerenciadas do Amplify.
- `SSM`— Crie, atualize e exclua o armazenamento `String` de parâmetros e os parâmetros SSM gerenciados pelo Amplify. `SecureString`
- `AWS AppSync`— Atualize e recupere recursos de AWS AppSync esquema, resolvedor e função. O objetivo é oferecer suporte à funcionalidade de hotswapping do sandbox de segunda geração.
- `Lambda`— Atualize e recupere a configuração das funções gerenciadas do Amplify. O objetivo é oferecer suporte à funcionalidade de hotswapping do sandbox de segunda geração.
- `Amazon S3`— Recupere 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`— Leia metadados de instâncias de banco de dados, clusters e proxies.
- `Amazon EC2`— Leia as informações da zona de disponibilidade de uma sub-rede.

Para ver as permissões dessa política, consulte AmplifyBackendDeployFullAccessna Referência de política AWS gerenciada.

Amplifique as atualizações das políticas gerenciadas AWS

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 `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
AmplifyBackendDeployFullAccess: atualizar para uma política existente	Adicione uma nova declaração `AmplifyDiscoverRDSVpcConfig` de política 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 `npx amplify generate schema-from-database` comando Amplify Gen 2, que permite aos clientes gerar um esquema de dados Typescript	17 de abril de 2024

Alteração	Descrição	Data
	a partir de um banco de dados SQL existente. Adicione as `ec2:DescribeSubnets` permissões `rds:DescribeDBProxies` `rds:DescribeDBInstances` `rds:DescribeDBClusters` ,`rds:DescribeDBSubnetGroups` ,, e. 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 necessária para provisionar os outros recursos necessários para configurar uma AWS AppSync API apoiada por um banco de dados SQL.	

Alteração	Descrição	Data
AmplifyBackendDeployFullAccess: atualizar para uma política existente	Adicione a ação `cloudformation:DeleteStack` de política para oferecer suporte à exclusão da pilha quando a `DeleteBranch` API for chamada. Adicione a ação `lambda:GetFunction` de política para oferecer suporte às funções de hotswapping. Adicione a ação `lambda:UpdateFunctionConfiguration` política para oferecer suporte às atualizações da função Lambda.	5 de abril de 2024
AdministratorAccess-Amplificar — Atualização de uma política existente	Adicione as `cloudformation:UnTagResource` permissões `cloudformation:TagResource` e para oferecer suporte a chamadas para AWS CloudFormation APIs.	4 de abril de 2024

Alteração	Descrição	Data
AmplifyBackendDeployFullAccess: atualizar 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 o hotswapping de ativos do Amazon S3. Adicione a ação `lambda:UpdateFunctionCode` de política para oferecer suporte às funções de hotswapping.	02 de janeiro de 2024
AmplifyBackendDeployFullAccess: atualizar 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: atualizar para uma política existente	Adicione a permissão `amplify:GetBackendEnvironment` para apoiar a implantação do aplicativo Amplify.	6 de novembro de 2023
AmplifyBackendDeployFullAccess – Nova política	O Amplify adicionou uma nova política com as permissões mínimas necessárias para implantar os recursos de back-end do Amplify.	8 de outubro de 2023

Alteração	Descrição	Data
AdministratorAccess-Amplificar — Atualização de uma política existente	Adicione a permissão `ecr:DescribeRepositories` exigida pela CLI (Interface de Linha de Comando) do Amplify.	1.º de junho de 2023

Alteração	Descrição	Data
AdministratorAccess-Amplificar — Atualização de 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 AWS 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-Amplificar — Atualização de 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-Amplificar — Atualização de 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-Amplificar — Atualização de 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-Amplificar — Atualização de 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-Amplificar — Atualização de 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-Amplificar — Atualização de 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 AWS CloudFormation 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 CloudWatch 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-Amplificar — Atualização de 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 AWS 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 AWS 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 acessem 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 back-end 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 acessem meus recursos do Amplify

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode 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), você pode 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 usar funções e políticas baseadas em recursos 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 AWS IAM Identity Center 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 os recursos. AWS
- 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 certificados SHA-256 que são gerenciados pela Autoridade de Certificação Privada. AWS Certificate Manager Para obter mais informações, consulte Processo de assinatura do Signature versão 4 e O que é o ACM PCA?.

Gerenciamento de chave 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 os 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

Auditores terceirizados avaliam a segurança e a conformidade AWS Amplify como parte de vários programas de AWS conformidade. Eles incluem SOC, PCI, ISO, HIPAA, MTCS, C5, K-ISMS, ENS High, OSPAR, 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 regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

> **Note**
>
> Nem todos Serviços da AWS são elegíveis para a HIPAA. Para mais informações, consulte a Referência dos serviços qualificados pela HIPAA.

- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.
- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

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 suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

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. Monitorar
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log a partir de 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 ter mais informações, consulte Log de chamadas de API do Amplify usando o 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 S oftware-as-a -Service (SaaS) AWS e 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.

Prevenção do problema do substituto confuso entre serviços

O problema de "confused deputy" é uma questão de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar em um problema confuso de delegado. A personificação entre serviços 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 as chaves de contexto de condição `aws:SourceAccount`global `aws:SourceArn`e as chaves de contexto 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 estiver especificando 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.

```
{
  "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.

```
{
  "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 página Set-Cookie na Mozilla Developer Network.

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	Descrição
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 com suporte: 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 gigabites	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	Descrição
Tamanho do artefato de cache	Cada região compatível: 5 gigabites	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 gigabites	Não	O tamanho máximo (em GB) do artefato de cache do ambiente.
Tamanho do arquivo ZIP de implantação manual	Cada região compatível: 5 gigabites	Não	O tamanho máximo (em GB) de um arquivo ZIP de implantação manual.
Máximo de criações de aplicativos por hora	Cada região com suporte: 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	Descrição
Solicitar tokens por segundo	Cada região com suporte: 20.000	Sim	O número máximo de tokens de solicitação por segundo para um aplicativo. O Amplify Hosting aloca tokens às solicitações com base na quantidade de recursos (tempo de processam ento e transferência de dados) que elas consomem.
Subdomínios por domínio	Cada região com suporte: 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 ter mais informações sobre as service quotas do Amplify, consulte AWS Amplify Endpoints e cotas 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 HTTP 429 (muitas 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 seu aplicativo retornar um código de status HTTP 429, as solicitações recebidas excederão a quantidade de tempo de processamento e transferência de dados atribuída ao seu aplicativo. Esse limite de aplicativos é gerenciado pela cota de `REQUEST_TOKENS_PER_SECOND` serviço do Amplify. Para obter mais informações sobre cotas, consulte Service quotas do Amplify Hosting.

Para corrigir esse problema, recomendamos otimizar seu aplicativo para reduzir a duração da solicitação e a transferência de dados para aumentar o RPS do aplicativo. Por exemplo, com os mesmos 20.000 tokens, uma página SSR altamente otimizada que responde em 100 milissegundos pode suportar um RPS mais alto em comparação com uma página com latência superior a 200 milissegundos.

Da mesma forma, um aplicativo que retorna um tamanho de resposta de 1 MB consumirá mais tokens do que um aplicativo que retorna 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.

Solução de problemas de 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

- Como faço para executar as funções do Amplify com o tempo de execução do Python?
- Como faço para executar comandos que exigem privilégios de superusuário ou root

Como faço para executar as funções do Amplify com o tempo de execução do Python?

O Amplify Hosting agora usa a imagem de compilação do Amazon Linux 2023 por padrão quando você implanta um novo aplicativo. 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. Portanto, você não precisa mais atualizar os comandos de compilação na especificação de compilação do seu aplicativo usando as instruções disponíveis nas Perguntas frequentes do Amplify Hosting GitHub .

Por padrão, a versão 3.10 do Python é usada globalmente. Para criar suas funções usando uma versão específica do Python, execute os comandos a seguir no arquivo de especificação de compilação do seu aplicativo.

```
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
```

Como faço para executar comandos que exigem 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 exigem privilégios de superusuário ou root, você deve executar esses comandos usando o comando Linux`sudo`. 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 Linux. `sudo` É uma prática recomendada `sudo` para usar comandos que exigem privilégios de superusuário.

Solucionar problemas de domínios personalizados

Se você encontrar problemas ao conectar um domínio personalizado ao seu aplicativo Amplify, consulte Solucionar problemas de domínios personalizados para obter ajuda.

Solução de problemas de aplicativos renderizados do lado do servidor

Se você encontrar problemas ao implantar um aplicativo SSR no Amplify, consulte para obter ajuda. Solução de problemas de implantações SSR

AWS AmplifyReferência de host

Use os tópicos nesta seção para encontrar material de referência detalhado para AWS Amplify.

Tópicos

Suporte a AWS CloudFormation

Use modelos AWS CloudFormation para provisionar recursos do Amplify, permitindo implantações de aplicativos da web repetíveis e confiáveis. O AWS CloudFormation fornece uma linguagem comum para você descrever e provisionar todos os recursos de infraestrutura em seu ambiente de nuvem, além de simplificar a implantação em várias regiões e/ou contas AWS 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.

Suporte a AWS Command Line Interface

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 a AWS Command Line Interface para marcar 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: 31 de maio de 2024

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	31 de maio de 2024
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	17 de abril de 2024
Capítulo de introdução atualizado	O Conceitos básicos do Amplify Hosting capítulo foi atualizado para usar um aplicativo de exemplo Next.js no tutorial.	12 de abril de 2024
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	5 de abril de 2024

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	4 de abril de 2024
Novo capítulo de solução de problemas	Foi adicionado o Solução de problemas do Amplify Hosting capítulo para descrever como corrigir problemas que você encontra com aplicativos implantados no Amplify Hosting.	2 de abril de 2024
Novo suporte para certificados SSL/TLS personalizados	O Usando certificados SSL/TLS tópico foi adicionado ao Configurar domínios personalizados capítulo para descrever o suporte do Amplify para certificados SSL/TLS personalizados ao conectar um aplicativo a um domínio personalizado.	20 de fevereiro de 2024
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	2 de janeiro de 2024

Alteração	Descrição	Data
Nova compatibilidade com frameworks de SSR	Adição do tópico Compatibilidade do Amplify com frameworks de SSR para descrever a compatibilidade do Amplify com qualquer framework de SSR baseado em JavaScript 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
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	17 de novembro de 2023
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	6 de novembro de 2023

Alteração	Descrição	Data
Novo tópico sobre subdomíni os curinga	Foi adicionado o tópico Subdomínios Wildcard para descrever o suporte para subdomínios curinga em domínios personalizados.	6 de novembro de 2023
Novas políticas gerenciadas	O AWS políticas gerenciad as para AWS Amplify tópico foi atualizado para descrever a nova política AmplifyBa ckendDeployFullAccess AWS gerenciada do Amplify.	8 de outubro de 2023
Novo suporte para lançament o de atributos de estruturas monorepo	O tópico Configurações de compilação do 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 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.	1.º de junho de 2023

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	24 de fevereiro de 2023
Capítulo de renderização no lado do servidor atualizado	O capítulo Implante aplicativos renderizados do lado do servidor com o Amplify Hosting foi atualizado para descrever as mudanças recentes no suporte do Amplify para as versões 12 e 13 do Next.js.	17 de novembro de 2022
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	30 de agosto de 2022
Tópico de políticas gerenciadas atualizado	O tópico Construindo um back-end para um aplicativo foi atualizado para descrever como implantar um back-end usando o Amplify Studio.	23 de agosto de 2022
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	27 de abril de 2022

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	17 de abril de 2022
Lançamento de novo recurso do GitHub aplicativo	Foi adicionado o Configurar o acesso do Amplify aos repositórios do GitHub tópico para descrever o novo GitHub aplicativo para autorizar o acesso do Amplify ao GitHub seu repositório.	5 de abril de 2022
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 componentes de interface do usuário que você pode conectar aos seus dados de back-end.	2 de dezembro de 2021
Tópico de políticas gerenciadas 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

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	8 de novembro de 2021
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	27 de setembro de 2021
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 capítulo Implante aplicativos renderizados do lado do servidor com o Amplify Hosting foi atualizado para descrever o novo suporte para o Next.js. versão 10.x.x e Next.js versão 11.	22 de julho de 2021

Alteração	Descrição	Data
Capítulo de configuração de configurações da compilação atualizado	Foi adicionado o tópico Configurações de compilação do 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 construção da configuração do Amplify (somente aplicativos de primeira geração) para descrever como gerar automaticamente o arquivo `aws-exports.js` no momento da compilação. Foi adicionado o tópico Compilações condicionais de back-end (somente aplicativos de primeira geração) para descrever como habilitar compilações condicionais de back-end. Foi adicionado o tópico Use os back-ends do Amplify em todos os aplicativos (somente aplicativos de primeira geração) para descrever como reutilizar back-ends existentes ao criar um novo aplicativo, conectar uma nova ramificação a um aplicativo existente ou atualizar um front-end existente para apontar para um ambiente de back-end 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 capítulo Implante aplicativos renderizados do lado do servidor com o Amplify Hosting para descrever o suporte do Amplify para aplicativos web que usam renderização do lado do servidor (SSR) e são criados com o 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 front-end web e móveis criarem e gerenciarem back-ends de aplicativos fora do AWS Management Console.	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 Configurar domínios personalizados 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.