

Guia do usuário

AWS Amplify Hospedagem



AWS Amplify Hospedagem: Guia do usuário

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é AWS Amplify hospedagem?	1
Estruturas compatíveis	1
Atributos do Amplify Hosting	2
Comece a usar o Amplify Hosting	2
Crie um back-end	3
Preços do Amplify Hosting	3
Tutoriais de introdução	4
Implantar uma aplicação Next.js	4
Etapa 1: Conectar um repositório	4
Etapa 2: confirme as configurações de compilação	5
Etapa 3: Implantar um aplicativo	6
Etapa 4: (opcional) limpar os recursos	7
Adicione recursos ao seu aplicativo	7
Implantar um aplicativo Nuxt.js	8
Implantar um aplicativo Astro.js	8
Implantar um SvelteKit aplicativo	11
Implantação de aplicativos renderizados do lado do servidor	15
Next.js	16
Suporte ao atributo Next.js	17
Implantação de um SSR aplicativo Next.js no Amplify	18
Migrando um SSR aplicativo Next.js 11 para a computação do Amplify Hosting	21
Adicionar SSR funcionalidade a um aplicativo Next.js estático	23
Tornando as variáveis de ambiente acessíveis aos runtimes do lado do servidor	25
Implantando um aplicativo Next.js em um monorepo	27
Nuxt.js	28
Astro.js	28
SvelteKit	29
Implemente um SSR aplicativo no Amplify	29
SSRrecursos suportados	30
Compatibilidade com versão Node.js para aplicações Next.js	31
Otimização de imagem para SSR aplicativos	31
Amazon CloudWatch Logs para SSR aplicativos	32
Amplify: suporte ao Next.js 11 SSR	32
Preços para SSR aplicativos	41

Solução de problemas de SSR implantações	41
Avançado: adaptadores de código aberto	41
Especificação de implantação	41
Implantação de um servidor Express	66
Otimização de imagens para autores de estruturas	72
Usando adaptadores de código aberto	81
Configurar domínios personalizados	83
Compreender a DNS terminologia e os conceitos	84
Terminologia do DNS	84
DNSverificação	85
Processo de ativação de domínio personalizado do Amplify Hosting	85
Usando TLS certificadosSSL/	86
Adicione um domínio personalizado gerenciado pelo Amazon Route 53	87
Adicione um domínio personalizado gerenciado por um DNS provedor terceirizado	88
Atualizar DNS registros para um domínio gerenciado pelo GoDaddy	94
Atualizar DNS registros de um domínio gerenciado pelo Google Domains	97
Atualizar o TLS certificadoSSL/para um domínio	100
Gerenciar subdomínios	101
Para adicionar somente um subdomínio	101
Para adicionar um subdomínio de vários níveis	102
Para adicionar ou editar um subdomínio	102
Subdomínios Wildcard	103
Para adicionar ou excluir um subdomínio curinga	103
Configure subdomínios automáticos para um domínio personalizado do Amazon Route 53	104
Pré-visualizações da Web com subdomínios	105
Solucionar problemas de domínios personalizados	105
Configurar as definições de compilação	106
Comandos e configurações de especificação de compilação	106
Configurações de compilação específicas de ramificação	109
Navegar para uma subpasta	110
Implantando o back-end com o front-end para um aplicativo de primeira geração	110
Definir a pasta de saída	111
Instalar pacotes como parte da compilação	111
Usar um registro privado de npm	111
Instalar pacotes do SO	112
Armazenamento de chave-valor para cada compilação	112

Ignorar a compilação para um commit	113
Desativar compilações automáticas	113
Ativar ou desativar a compilação e implantação de front-end com base em diff	113
Ativar ou desativar compilações de back-end baseadas em diferenças para um aplicativo de primeira geração	114
Configurações de compilação do Monorepo	115
Sintaxe YAML da especificação de compilação do Monorepo	116
Definindo a variável de ambiente AMPLIFY_MONOREPO_APP_ROOT	119
Configurar aplicativos Turborepo e pnpm monorepo	121
Implantações de ramificação de atributo	122
Fluxos de trabalho de equipe com aplicativos Amplify Gen 2 completos	123
Fluxos de trabalho de equipe com aplicativos Amplify Gen 1 completos	123
Fluxo de trabalho de ramificação de atributo	123
GitFlow fluxo de trabalho	129
Sandbox de desenvolvedor	130
Implantações de ramificação de atributo com base em padrão	132
Implantações de ramificações de atributos baseadas em padrões para um aplicativo conectado a um domínio personalizado	133
Geração automática em tempo de construção da configuração do Amplify (somente aplicativos de primeira geração)	133
Compilações condicionais de back-end (somente aplicativos de primeira geração)	135
Use os back-ends do Amplify em todos os aplicativos (somente aplicativos de primeira geração)	136
Reutilize back-ends ao criar um aplicativo	136
Reutilize back-ends ao conectar uma ramificação a um aplicativo existente	137
Edite um front-end existente para apontar para um back-end diferente	138
Construindo um back-end	139
Crie um back-end para um aplicativo de 2ª geração	139
Crie um back-end para um aplicativo de primeira geração	139
Pré-requisitos	139
Etapa 1: Implantar um front-end	140
Etapa 2: criar um back-end	141
Etapa 3: Conectar o back-end ao front-end	142
Próximas etapas	144
Implantações manuais	145
Implantação manual de arrastar e soltar	145

Implantação manual do Amazon S3 ou URL	146
Solução de problemas de acesso ao bucket do Amazon S3	147
Botão de implantação com apenas um clique	148
Adicione o botão Implantar no Amplify Hosting ao seu repositório ou blog	148
Configuração do acesso ao GitHub	150
Instalando e autorizando o aplicativo Amplify GitHub para uma nova implantação	150
Migrar um aplicativo OAuth existente para o aplicativo Amplify GitHub	152
Configurar o aplicativo Amplify GitHub para implantações de AWS CloudFormation, CLI e SDK	153
Configurar visualizações na web com o aplicativo Amplify GitHub	154
Visualizações de solicitação pull	155
Ativar visualizações na web	155
Acesso à pré-visualização na web com subdomínios	157
end-to-end Teste E	158
Tutorial: Configurar end-to-end testes com o Cypress	158
Adicione testes ao seu aplicativo Amplify existente	158
Desabilitar testes	160
Usar redirecionamentos	162
Tipos de redirecionamentos	162
Criando e editando redirecionamentos	163
Ordem de redirecionamentos	165
Parâmetros de consulta	165
Redirecionamentos e regravações simples	165
Redireciona para aplicativos de única página web (SPA)	168
Regravação de proxy reverso	168
Barras finais e URLs limpos	169
Espaços reservados	169
Strings de consulta e parâmetros de caminho	170
Redirecionamentos baseados em região	171
Expressões curinga em redirecionamentos e regravações	171
Acesso restrito	173
Variáveis de ambiente	175
Variáveis de ambiente do Amplify	175
Definição de variáveis de ambiente	181
Acesse variáveis de ambiente no momento da compilação	182
Tornando as variáveis de ambiente acessíveis aos runtimes do lado do servidor	183

Crie um novo ambiente de back-end com parâmetros de autenticação para login social	183
Variáveis de ambiente da estrutura de front-end	185
Gerenciando segredos do ambiente	185
Defina e acesse segredos do ambiente para um aplicativo de primeira geração	185
Acessar segredos de ambiente	186
Segredos do ambiente do Amplify	186
Cabeçalhos personalizados	188
Formato YAML de cabeçalho personalizado	188
Configuração de cabeçalhos personalizados	189
Migração de cabeçalhos personalizados	191
Cabeçalhos personalizados monorepo	192
Exemplo de cabeçalhos de segurança	193
Cabeçalhos de controle de cache personalizados	193
Webhooks recebidos	195
Monitorar	196
Monitoramento com CloudWatch	196
Metrics	196
Alarmes	199
Amazon CloudWatch Logs para aplicativos SSR	200
Logs de acesso	201
Como analisar os logs de acesso	202
Crie notificações	203
Configurar notificações por e-mail	203
Compilações personalizadas	204
Imagens de compilação personalizada	204
Requisitos de imagens de compilação personalizada	204
Configuração de uma imagem de compilação personalizada	205
Atualizações de pacote ao vivo	206
Configuração de atualizações de pacote ao vivo	207
Adicionar um perfil de serviço	208
Criar um perfil de serviço	208
Prevenção do problema do “confused deputy”	209
Gerenciando a configuração do cache	210
Como o Amplify aplica a configuração de cache	212
Entendendo as políticas de cache gerenciado do Amplify	213
Gerenciando cookies de chave de cache	216

Incluindo ou excluindo cookies da chave de cache	217
Alterando a configuração do cookie da chave de cache para um aplicativo	218
Gerenciar desempenho do aplicativo	220
Usar cabeçalhos para controlar a duração do cache	220
Configurando o Cache-Control cabeçalho para aumentar o desempenho do aplicativo	221
Log de chamadas de API do Amplify usando o AWS CloudTrail	222
Amplifique as informações em CloudTrail	222
Noções básicas sobre as entradas de arquivos de log do Amplify	223
Segurança	227
Identity and Access Management	227
Público	228
Autenticando com identidades	229
Gerenciando acesso usando políticas	232
Como o Amplify funciona com IAM	235
Exemplos de políticas baseadas em identidade	242
AWSPolíticas gerenciadas	245
Solução de problemas	260
Proteção de dados	262
Criptografia em repouso	263
Criptografia em trânsito	264
Gerenciamento de chave de criptografia	264
Compliance Validation	264
Segurança da infraestrutura	265
Logging e monitoramento	266
Prevenção do problema do substituto confuso entre serviços	267
Melhores práticas de segurança	269
Usando cookies com o domínio padrão do Amplify	269
Cotas	270
Solução de problemas	273
Problemas gerais	273
HTTPCódigo de status 429 (muitas solicitações)	273
AL2023 construir imagem	274
Como faço para executar as funções do Amplify com o tempo de execução do Python?	274
Como faço para executar comandos que exigem privilégios de superusuário ou root?	275
Domínios personalizados	275
Preciso verificar se meu CNAME problema está resolvido	276

Meu domínio hospedado com terceiros está paralisado no estado de Verificação pendente	276
Meu domínio hospedado com o Amazon Route 53 está paralisado no estado de verificação pendente	277
Eu recebo um CNAMEAlreadyExistsException erro	278
Eu recebo um erro de verificação adicional necessária	279
Eu recebo um erro 404 no CloudFront URL	279
Recebo SSL certificados ou HTTPS erros ao visitar meu domínio	280
Renderização do lado do servidor () SSR	281
Você está usando um adaptador de framework	282
APIAs rotas Edge fazem com que sua compilação do Next.js falhe	282
A regeneração estática incremental sob demanda não está funcionando para seu aplicativo	282
A saída de compilação do seu aplicativo excede o tamanho máximo permitido	282
Sua compilação falha com um erro de falta de memória	38
O tamanho da HTTP resposta é muito grande	285
AWS AmplifyReferência de host	286
Suporte a AWS CloudFormation	286
Suporte a AWS Command Line Interface	286
Suporte para marcação de recursos	286
Amplify Hosting API	286
Histórico do documento	287
.....	ccci

Bem-vindo ao AWS Amplify Hosting

O Amplify Hosting fornece um fluxo de trabalho baseado em Git para hospedar aplicativos web sem servidor completos com implantação contínua. 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 oferece suporte a muitas SSR estruturas comuns, estruturas de aplicativo (SPA) de página única e geradores de sites estáticos, incluindo os seguintes.

SSREstruturas

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

SPAestruturas

- React
- Angular
- Vue.js
- Ionic
- Brasa

Geradores de sites estáticos

- Onze
- Gatsby
- Hugo
- Jekyll

- VuePress

Atributos do Amplify Hosting

[Ramificações 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](#)

Pré-visualize as alterações durante as análises de código.

[End-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 SEO classificações 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 [Começando a implantar um aplicativo no 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

AWS 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.

Para entender melhor a abordagem Gen 2 do Amplify que prioriza o código, consulte o Workshop [Amplify Gen 2 no site do Workshop Studio](#).AWS Neste tutorial abrangente, você cria um aplicativo sem servidor com React e Next.js e aprende a usar as bibliotecas de dados e autenticação do Amplify Gen 2 e a biblioteca Amplify UI para adicionar funcionalidade ao aplicativo.

Se você estiver procurando a documentação para criar back-ends para um aplicativo de primeira geração, usando o 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.](#)

Começando a implantar um aplicativo no Amplify Hosting

Para ajudar você a entender como o Amplify Hosting funciona, os tutoriais a seguir orientam você na criação e implantação de aplicativos criados usando estruturas SSR comuns compatíveis com o Amplify.

Tutoriais

- [Implante um aplicativo Next.js no Amplify Hosting](#)
- [Implante um aplicativo Nuxt.js no Amplify Hosting](#)
- [Implante um aplicativo Astro.js no Amplify Hosting](#)
- [Implemente um SvelteKit aplicativo no Amplify Hosting](#)

Implante um aplicativo Next.js no Amplify Hosting

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

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-app](#) instruções 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 Amplify serviço.

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, consulte [Configurar 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](#) (). 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 solicitações 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, API chaves ou parâmetros da conexão do banco de dados. 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 um nome 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 domínio personalizado URL, como `https://www.example.com`. Para obter mais informações, consulte [Configurar domínios personalizados](#).

Visualizações de solicitação pull

As visualizações prévias de pull request da 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 um aplicativo Nuxt.js no Amplify Hosting

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

Para implantar um aplicativo Nuxt no Amplify Hosting

1. Faça login no 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. Se você quiser que o Amplify seja capaz de entregar registros de aplicativos para o Amazon CloudWatch Logs, você deve habilitar isso explicitamente no console. Abra a seção Configurações avançadas e escolha Habilitar registros de SSR aplicativos na seção Implantação do Server-Side Rendering () SSR.
6. Escolha Próximo.
7. Na página Revisar, escolha Salvar e implantar.

Implante um aplicativo Astro.js no Amplify Hosting

Use as instruções a seguir para implantar um aplicativo Astro.js no Amplify Hosting. Você pode usar um aplicativo existente ou criar um aplicativo inicial usando um dos exemplos oficiais fornecidos pelo Astro. Para criar um aplicativo inicial, consulte [Usar um tema ou modelo inicial](#) na documentação do Astro.

Para implantar um site Astro com SSR o Amplify Hosting, você deve adicionar um adaptador ao seu aplicativo. Não mantemos um adaptador de propriedade da Amplify para a estrutura Astro. Este tutorial usa o `astro-aws-amplify` adaptador que foi criado por um membro da comunidade. Esse adaptador está disponível em astro-aws-amplifygithub.com/alexnguyennz/ no site. GitHub AWS não mantém esse adaptador.

Para implantar um aplicativo Astro no Amplify Hosting

1. Em seu computador local, navegue até o aplicativo Astro para implantar.
2. Para instalar o adaptador, abra uma janela de terminal e execute o comando a seguir. Este exemplo usa o adaptador de comunidade disponível em astro-aws-amplifygithub.com/alexnguyennz/. Você pode substituir `astro-aws-amplify` com o nome do adaptador que você está usando.

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

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

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

import sitemap from '@astrojs/sitemap';

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

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

Agora você está pronto para implantar seu aplicativo Astro no Amplify.

5. Faça login no AWS Management Console e abra o console do [Amplify](#).
6. Na página Todos os aplicativos, escolha Criar novo aplicativo.
7. Na página Comece a criar com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.

8. Na página Adicionar ramificação do repositório, faça o seguinte:
 - a. Selecione o nome do repositório a ser conectado.
 - b. Selecione o nome da ramificação do repositório para se conectar.
 - c. Escolha Próximo.
9. Na página de configurações do aplicativo, localize a seção Configurações de compilação. Em Diretório de saída do Build, insira **.amplify-hosting**.

Você também deve atualizar os comandos de criação de front-end do aplicativo. No momento, você não pode adicionar novos comandos à especificação de compilação na implantação. Depois de implantar o aplicativo, você deve atualizar a especificação de compilação usando as instruções na etapa 13.
10. Se você quiser que o Amplify seja capaz de entregar registros de aplicativos para o Amazon CloudWatch Logs, você deve habilitar isso explicitamente no console. Abra a seção Configurações avançadas e escolha Habilitar registros de SSR aplicativos na seção Implantação do Server-Side Rendering () SSR.
11. Escolha Próximo.
12. Na página Revisar, escolha Salvar e implantar.
13. Depois que seu aplicativo for implantado, atualize os comandos de compilação de front-end na especificação de compilação. Escolha Hospedagem e, em seguida, escolha Configurações de criação.
14. Na página Configurações de compilação, escolha Editar.
15. No `amplify.yml` arquivo, localize a seção de comandos de construção de front-end. Digite **mv node_modules ./amplify-hosting/compute/default**

Seu arquivo de configurações de compilação deve ter a seguinte aparência.

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

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

16. Escolha Salvar.

17. Reimplante o aplicativo.

Implemente um SvelteKit aplicativo no Amplify Hosting

Use as instruções a seguir para implantar um SvelteKit aplicativo no Amplify Hosting. Você pode usar seu próprio aplicativo ou criar um aplicativo inicial. Para obter mais informações, consulte [Criação de um projeto](#) na SvelteKit documentação.

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

Para implantar um SvelteKit aplicativo no Amplify Hosting

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

```
npm install amplify-adapter
```

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

```
import adapter from 'amplify-adapter';
```

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

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

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

export default config;
```

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

Faça login no AWS Management Console e abra o console do [Amplify](#).

6. Na página Todos os aplicativos, escolha Criar novo aplicativo.
7. Na página Comece a criar com o Amplify, escolha seu provedor de repositório Git e escolha Avançar.
8. Na página Adicionar ramificação do repositório, faça o seguinte:
 - a. Selecione o nome do repositório a ser conectado.
 - b. Selecione o nome da ramificação do repositório para se conectar.
 - c. Escolha Próximo.
9. Na página de configurações do aplicativo, localize a seção Configurações de compilação. Em Diretório de saída do Build, insira **build**.

Você também deve atualizar os comandos de criação de front-end do aplicativo. No momento, você não pode adicionar novos comandos à especificação de compilação na implantação.

Depois de implantar o aplicativo, você deve atualizar a especificação de compilação usando as instruções na etapa 13.

10. Se você quiser que o Amplify seja capaz de entregar registros de aplicativos para o Amazon CloudWatch Logs, você deve habilitar isso explicitamente no console. Abra a seção Configurações avançadas e escolha Habilitar registros de SSR aplicativos na seção Implantação do Server-Side Rendering () SSR.
11. Escolha Próximo.
12. Na página Revisar, escolha Salvar e implantar.
13. Depois que seu aplicativo for implantado, atualize os comandos de compilação de front-end na especificação de compilação. Escolha Hospedagem e, em seguida, escolha Configurações de criação.
14. Na página Configurações de compilação, escolha Editar.
15. No `amplify.yml` arquivo, localize a seção de comandos de construção de front-end. Digite **`cd build/compute/default/ - npm i --production`** e.

Seu arquivo de configurações de compilação deve ter a seguinte aparência.

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

  artifacts:
    baseDirectory: build
    files:
      - '**/*'

  cache:
    paths:
      - '.npm/**/*'
```

16. Escolha Salvar.

17. Reimplante o aplicativo.

Implantação de 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 suporta qualquer SSR estrutura baseada em Javascript com um adaptador de compilação de código aberto que transforma a saída de compilação de um aplicativo na estrutura de diretórios que a Amplify Hosting espera. Por exemplo, você pode implantar aplicativos criados com o Nuxt, o Astro e as SvelteKit estruturas instalando os adaptadores disponíveis.

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

Você pode implantar as seguintes estruturas no Amplify Hosting com configuração mínima.

Next.js

- O Amplify suporta aplicativos Next.js 14 sem a necessidade de um adaptador. Para começar, consulte o [Amplifique o suporte para Next.js](#).

Nuxt.js

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

Astro.js

- O Amplify suporta implantações de aplicativos Astro.js com um adaptador comunitário. Para começar, consulte o [Amplifique o suporte para Astro.js](#).

SvelteKit

- O Amplify suporta implantações SvelteKit de aplicativos com um adaptador comunitário. Para começar, consulte o [Amplifique o suporte para SvelteKit](#).

Adaptadores de código aberto

- Use um adaptador de código aberto - Para obter instruções sobre como usar qualquer adaptador que não esteja na lista anterior, consulte [Usando adaptadores de código aberto](#).
- Crie um adaptador de estrutura - Os autores da estrutura que desejam integrar os recursos fornecidos por uma estrutura podem usar a especificação de implantação do Amplify Hosting

para configurar sua saída de compilação de acordo com a estrutura que a Amplify espera. Para obter mais informações, consulte [Usando a especificação de implantação do Amplify Hosting para configurar a saída da compilação](#).

- Configure um script de pós-compilação - Você pode usar a especificação de implantação do Amplify Hosting para manipular sua saída de compilação conforme necessário para cenários específicos. Para obter mais informações, consulte [Usando 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](#).

Tópicos

- [Amplifique o suporte para Next.js](#)
- [Amplifique o suporte para Nuxt.js](#)
- [Amplifique o suporte para Astro.js](#)
- [Amplifique o suporte para SvelteKit](#)
- [Implemente um SSR aplicativo no Amplify](#)
- [SSRrecursos suportados](#)
- [Preços para SSR aplicativos](#)
- [Solução de problemas de SSR implantações](#)
- [Avançado: adaptadores de código aberto](#)

Amplifique o suporte para Next.js

O Amplify oferece suporte à implantação e hospedagem de aplicativos web renderizados (SSR) do lado do servidor criados usando Next.js. Next.js é uma estrutura React para desenvolvimento SPAs com JavaScript. 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 a geração estática de sites (SSG) e SSR em um único projeto. SSGas páginas são pré-renderizadas no momento da criação e SSR as páginas 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 HTML conteúdo 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 na SEO classificação do site. A pré-renderização também melhora SEO ao permitir que os robôs dos mecanismos de pesquisa encontrem e rastreiem o conteúdo de um site com facilidade. HTML

O Next.js fornece suporte analítico integrado para medir várias métricas de desempenho, como Time to first byte (TTFB) e First contentful paint (FCP). Para obter mais informações sobre 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 Next.js 12, 13 e 14. 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, 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 gerenciado de computação SSR Amplify Hosting.

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

Atributos compatíveis

- Páginas renderizadas do lado do servidor () SSR
- Páginas estáticas
- API rotas
- Rotas dinâmicas
- Detecção de todas as rotas
- SSG (Geração estática)
- Regeneração estática incremental () ISR
- Roteamento de subcaminhos internacionalizado (i18n)
- Roteamento de domínio internacionalizado (i18n)
- Detecção automática de localidade internacionalizada (i18n)
- Middleware
- Variáveis de ambiente

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

Atributos não compatíveis

- Edge API Routes (o middleware Edge não é suportado)
- Regeneração estática incremental sob demanda () ISR
- 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 Next.js para redimensioná-lo e otimizá-lo em um AVIF formato Webp ou, em seguida, exibi-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.

Implantação de um SSR aplicativo Next.js no 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 totalmente os recursos necessários para implantar um aplicativo. SSR SSRs aplicativos em sua conta do Amplify que você implantou antes de 17 de novembro de 2022 estão usando o provedor Classic (somente Next.js 11). SSR

É altamente recomendável que você migre aplicativos usando o Classic (somente Next.js 11) SSR para o provedor de computação Amplify Hosting SSR. 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 SSR aplicativo Next.js 11 para a computação do Amplify Hosting](#).

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

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

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. Na lista Repositórios atualizados recentemente, selecione o nome do repositório a ser conectado.
 - b. Na lista Ramificação, selecione o nome da ramificação do repositório a ser conectada.
 - c. Escolha Próximo.
5. O aplicativo requer uma função IAM de serviço que o Amplify assume ao ligar para 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 criação do aplicativo no arquivo `package.json`

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

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

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

Amplifique as configurações de compilação de um aplicativo Next.js

Depois de inspecionar o `package.json` arquivo do seu aplicativo, o Amplify verifica as configurações de 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 obter mais informações, consulte [Configurar as definições de compilação](#).

Se o Amplify detectar que você está implantando um SSR aplicativo Next.js e nenhum `amplify.yml` arquivo estiver presente, ele gerará uma especificação de construção para o aplicativo e definirá como `baseDirectory: .next`. Se você estiver implantando um aplicativo em que um arquivo `amplify.yml` esteja presente, as configurações da compilação no arquivo substituirão todas as configurações da compilação no console. Portanto, 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 construção são para um aplicativo Next.js que oferece suporte SSG a SSR páginas.

```
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 SSG aplicativo, ele gera uma especificação de construção para o aplicativo e define como `baseDirectory: out`. Se você estiver implantando

um aplicativo em que um arquivo `amplify.yml` está presente, deverá definir manualmente o valor `baseDirectory` para `out` no arquivo.

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 construção são para um aplicativo Next.js que oferece suporte somente a SSG páginas.

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

Migrando um SSR 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 SSR provedor 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 Next.js 12, 13 e 14. O console exibe um banner de informações identificando aplicativos com ramificações que são implantadas usando o SSR provedor anterior do Amplify, Classic (somente Next.js 11). É altamente recomendável que você migre seus aplicativos para o provedor de computação Amplify Hosting SSR.

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

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

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

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 o Next.js versão 12 ou posterior.

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 Classic (somente Next.js 11) SSR. No banner, escolha Migrar.
5. Na janela de confirmação da migração, selecione as três instruções e escolha Migrar.
6. O Amplify criará e reimplantarão seu aplicativo para concluir a migração.

Revertendo uma migração 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 SSG aplicativo é definido com o valor da plataformaWEB. Um SSR aplicativo usando a versão 11 do Next.js é definido com o valor da plataformaWEB_DYNAMIC. Um SSR aplicativo Next.js 12 ou posterior é definido com o valor da plataformaWEB_COMPUTE.

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
```

Adicionar SSR funcionalidade a um aplicativo Next.js estático

Você pode adicionar SSR funcionalidade a um aplicativo Next.js estático (SSG) existente implantado com o Amplify. Antes de iniciar o processo de conversão do SSG aplicativo em SSR, atualize o aplicativo para usar a versão 12 ou posterior do Next.js e adicione SSR funcionalidades. 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 SSG aplicativo está configurado para o tipo de plataforma `WEB`. Um SSR aplicativo usando a versão 11 do Next.js está definido como tipo de plataforma `WEB_DYNAMIC`. Para aplicativos implantados no Next.js 12 ou posterior usando a computação SSR gerenciada pelo Amplify Hosting, o tipo de plataforma está definido como `WEB_COMPUTE`.

Quando você implantou seu aplicativo como um SSG aplicativo, 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 uma função de serviço a um SSG aplicativo 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 SSR funcionalidade, você deve atualizar as configurações de 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 construção `"next build"` para indicar que o aplicativo Next.js oferece suporte a ambas SSG e SSR páginas.

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

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

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

SSR variáveis de ambiente para monorepos

Se você estiver implantando um SSR aplicativo 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`. Os aplicativos `Turborepo`

e pnpm requerem configuração adicional. Para obter 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](#).

Amplifique o suporte para Nuxt.js

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

Adaptador

Você pode implantar um aplicativo Nuxt.js no Amplify usando um adaptador predefinido com configuração zero. Para obter mais informações sobre o adaptador, consulte a [documentação do Nuxt](#).

Tutorial

Para saber como implantar um aplicativo Nuxt.js no Amplify, consulte. [Implante um aplicativo Nuxt.js no Amplify Hosting](#)

Demonstração

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

Amplifique o suporte para Astro.js

O Astro é uma estrutura web para criar aplicativos web baseados em conteúdo.

Adaptador

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

Tutorial

Para saber como implantar um aplicativo Astro no Amplify, consulte. [Implante um aplicativo Astro.js no Amplify Hosting](#)

Demonstração

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

Amplifique o suporte para SvelteKit

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

Adaptador

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

Tutorial

Para saber como implantar um SvelteKit aplicativo no Amplify, consulte. [Implemente um SvelteKit aplicativo no Amplify Hosting](#)

Demonstração

Para ver uma demonstração em vídeo, consulte [Como implantar um SvelteKit site \(comAPI\) AWS no YouTube canal da Amazon Web Services](#).

Implemente um SSR aplicativo no Amplify

Você pode usar essas instruções para implantar um aplicativo criado com qualquer estrutura com um pacote de implantação que esteja em conformidade com a saída de compilação que o Amplify espera. Não será necessário usar nenhum adaptador se você estiver implantando uma aplicação Next.js.

Se você estiver implantando um SSR aplicativo que usa um adaptador de estrutura, primeiro instale e configure o adaptador. Para obter instruções, consulte [Usando adaptadores de código aberto](#).

Para implantar um SSR aplicativo 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 Next.js. SSR

Se você estiver implantando um SSR aplicativo que usa um adaptador para outra estrutura, você deve habilitar explicitamente o Amazon CloudWatch Logs. Abra a seção Configurações avançadas e escolha Habilitar registros de SSR aplicativos na seção Implantação do Server-Side Rendering () SSR.

6. O aplicativo requer uma função IAM de serviço que o Amplify assume para entregar registros ao seu. 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.

SSRrecursos suportados

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

O Amplify fornece suporte à versão do Node.js para corresponder à versão do Node.js que foi usada para criar seu aplicativo.

O Amplify fornece um recurso integrado de otimização de imagem compatível com todos os SSR aplicativos. Se você não quiser usar o recurso de otimização de imagem padrão, você pode implementar um carregador de otimização de imagem personalizado.

Tópicos

- [Compatibilidade com versão Node.js para aplicações Next.js](#)
- [Otimização de imagem para SSR aplicativos](#)
- [Amazon CloudWatch Logs para SSR aplicativos](#)
- [Amplify: suporte ao Next.js 11 SSR](#)

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.

Otimização de imagem para SSR aplicativos

O Amplify Hosting fornece um recurso integrado de otimização de imagem compatível com todos os SSR aplicativos. 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 você estiver implementando um carregador personalizado, consulte o tópico a seguir Usando um carregador de imagem personalizado.

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](#).

Amazon CloudWatch Logs para SSR aplicativos

Amplify envia informações sobre seu SSR tempo de execução para o Amazon CloudWatch Logs em seu. Conta da AWS Quando você implanta um SSR aplicativo, o aplicativo exige uma função IAM de serviço que o Amplify assume ao ligar para 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 IAM função para você, a função já terá as permissões para criar CloudWatch registros. Se você criar sua própria IAM função, 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 SSR

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, Classic (somente Next.js 11). A documentação nesta seção se aplica somente aos aplicativos implantados usando o SSR provedor Classic (somente Next.js 11).

Note

É altamente recomendável que você migre seus aplicativos Next.js 11 para o provedor gerenciado de computação SSR Amplify Hosting. Para obter mais informações, consulte [Migrando um SSR aplicativo Next.js 11 para a computação do Amplify Hosting](#).

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

Atributos compatíveis

- Páginas renderizadas do lado do servidor () SSR
- Páginas estáticas
- APIrotas
- 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 sob demanda () ISR
- Roteamento de domínio internacionalizado (i18n)
- Detecção automática de localidade internacionalizada (i18n)
- Middleware
- Middleware de borda
- APIRotas Edge

Preços dos SSR aplicativos Next.js 11

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

- 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 SSR aplicativos Next.js 11

O Amplify requer AWS Identity and Access Management (IAM) permissões para implantar um SSR aplicativo. Sem as permissões mínimas exigidas, você receberá um erro ao tentar implantar seu SSR aplicativo. Para fornecer ao Amplify as permissões necessárias, você deve especificar um perfil de serviço.

Para criar uma função IAM de serviço que o Amplify assume ao ligar para 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 IAM ações, e deve ser considerada tão poderosa quanto a `AdministratorAccess` política. Essa política fornece mais permissões do que o necessário para implantar seu SSR aplicativo.

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 à sua função de serviço, você pode criar sua própria IAM política gerenciada pelo cliente que concede somente as permissões necessárias para implantar seu SSR aplicativo. Consulte [Criação de IAM políticas](#) no Guia do IAM usuário 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 SSR aplicativo.

```
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 SSR implantações do Next.js 11

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

Tópicos

- [Seu diretório de saída foi substituído](#)
- [Você recebe um erro 404 após implantar seu site SSR](#)
- [Seu aplicativo não tem a regra de reescrita para distribuições CloudFront SSR](#)
- [Seu aplicativo é muito grande para ser implantado](#)
- [Sua compilação falha com um erro de falta de memória](#)
- [Seu aplicativo tem SSG ramificações SSR e duas](#)
- [Seu aplicativo armazena arquivos estáticos em uma pasta com um caminho reservado](#)
- [Seu aplicativo atingiu um CloudFront limite](#)
- [As variáveis de ambiente não são transferidas para as funções do Lambda](#)
- [As funções do Lambda@Edge são criadas na região Leste dos EUA \(Norte da Virgínia\)](#)
- [Seu aplicativo Next.js usa atributos não compatíveis](#)
- [As imagens em seu aplicativo Next.js não estão carregando](#)
- [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 SSR aplicativo, o Amplify cria uma regra de reescrita para suas distribuições. CloudFront 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, Regrações 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.

Rewrites and redirects

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

Rewrites and redirects				Edit
<input type="text" value="Search"/>				< 1 > ⚙
Source address	Target address	Type	Country code	
/<*>	https:// .cloudfront.net/<*>	200 (Rewrite)	-	
/<*>	/index.html	404 (Rewrite)	-	

Seu aplicativo é muito grande para ser implantado

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

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

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

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 o [AWS CodeBuild](#) site do Next.js.

Seu aplicativo tem SSG ramificações SSR e duas

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

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 SSR aplicativo não são transferidas para as funções do AWS Lambda aplicativo. 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, você pode migrar o aplicativo para o provedor de computação do Amplify Hosting. SSR Para obter instruções, consulte [Migrando um SSR 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 de documentos.

Regiões não compatíveis

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

Preços para SSR aplicativos

Quando você implanta um SSR aplicativo, a computação do Amplify Hosting gerencia os recursos necessários para implantar o SSR aplicativo para você. Para obter informações sobre as cobranças de computação do Amplify Hosting, consulte [Preços do AWS Amplify](#).

Solução de problemas de SSR implantações

Se você tiver problemas inesperados ao implantar um SSR aplicativo com a computação do Amplify Hosting, [Solução de problemas de aplicativos renderizados do lado do servidor](#) consulte o capítulo de solução de problemas do Amplify.

Avançado: adaptadores de código aberto

Os autores da estrutura 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, poderá desenvolver sua própria solução para gerar a saída da compilação que o Amplify espera.

Tópicos

- [Como usar a especificação de implantação do Amplify Hosting para configurar a saída da compilação](#)
- [Como implantar um servidor Express usando o manifesto de implantação](#)
- [Integração de otimização de imagem para criadores de frameworks](#)
- [Usando adaptadores de código aberto](#)

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

A especificação de implantação do Amplify Hosting é uma especificação baseada em sistema de arquivos que define a estrutura de diretórios que facilita as implantações no Amplify Hosting.

Um framework pode gerar essa estrutura esperada de diretórios como saída do seu comando de compilação, permitindo que o framework aproveite as vantagens dos serviços primitivos do Amplify Hosting. A Amplify Hosting entende a estrutura do pacote de implantação e o implanta adequadamente.

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

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

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

Amplifique SSR o suporte primitivo

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

Ativos estáticos

Fornece aos frameworks a capacidade de hospedar arquivos estáticos.

Computação

Fornece estruturas com a capacidade de executar um HTTP servidor 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 todos os arquivos estáticos acessíveis ao público que devem ser servidos pelo aplicativo URL no `.amplify-hosting/static` diretório. Os arquivos dentro desse diretório serão oferecidos por meio do primitivo de ativos estáticos.

Os arquivos estáticos podem ser acessados na raiz (`/`) do aplicativo URL sem nenhuma alteração em seu conteúdo, nome de arquivo ou extensão. Além disso, os subdiretórios são preservados na URL estrutura 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 primitivo computacional do Amplify Hosting.

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

- É necessário ter um arquivo na raiz do subdiretório `default` para atuar como ponto de entrada para o recurso computacional.
- O arquivo do ponto de entrada deve ser um módulo `Node.js` e deve iniciar um `HTTP` servidor que escuta 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 as estruturas podem agrupar seus `HTTP` servidores da maneira que quiserem. Se for possível iniciar o processo de computação com o comando `node server.js`, com `server.js` 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 JSON exemplo a seguir demonstra o uso do `version` atributo.

```
"version": 1
```

Como usar o atributo routes

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

As regras de roteamento não gravam 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	<p>Define um padrão que corresponde aos caminhos da solicitação recebida (excluindo a string de consulta).</p> <p>O caminho pode ter até 255 caracteres.</p> <p>Um caminho deve começar com a barra /.</p> <p>Um caminho pode conter qualquer um dos seguintes caracteres: [A-Z], [a-z], [0-9], [_ . * \$ / ~ " ' @ : +].</p> <p>Somente os seguintes caracteres curinga são compatíveis para correspondência de padrão:</p>

Chave	Tipo	Obrigatório	Descrição
			<ul style="list-style-type: none">• * (corresponde a 0 ou mais caracteres)• O padrão /* é chamado de padrão abrangente e corresponderá a todas as solicitações recebidas.
target	Destino	Sim	<p>Um objeto que define o destino para o qual rotear a solicitação correspondente.</p> <p>Se houver a especificação de uma rota Compute, deverá haver um ComputeResource correspondente.</p> <p>Se houver a especificação de uma rota ImageOptimization, também deverá haver um imageSettings especificado.</p>


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

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 <code>Static</code> , <code>Compute</code> e <code>ImageOptimization</code> .
src	String	Sim para Compute Não para outros primitivos	<p>Uma string que especifica o nome do subdiretório no pacote de implantação que contém o código executável do primitivo. Válido e necessário somente para o primitivo <code>Compute</code>.</p> <p>O valor deve apontar para um dos recursos computacionais presentes no pacote de implantação. No momento, o único valor compatível para esse campo é <code>default</code>.</p>
cacheControl	String	Não	Uma string que especifica o valor do cabeçalho <code>Cache-Control</code> a ser aplicado à resposta. Válido somente para o estático e os

Chave	Tipo	Obrigatório	Descrição
			<p>ImageOptimization primitivos.</p> <p>O valor especificado é substituído por cabeçalhos personalizados. Para obter mais informações sobre os cabeçalhos personalizados do Amplify Hosting, consulte Cabeçalhos personalizados.</p> <div data-bbox="1187 842 1508 1539" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Esse cabeçalho Cache-Control é aplicado somente a respostas bem-sucedidas com um código de status definido como 200 (OK).</p></div>

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

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

```
}
```

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

Estático

Roteia as solicitações para o primitivo de ativos estáticos.

Computação

Roteia as solicitações para o primitivo de computação.

ImageOptimization

Roteia as solicitações para o primitivo de otimização de imagem.

O JSON exemplo a seguir demonstra o uso do routes atributo 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](#).

Usando o `computeResources` atributo

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
<code>name</code>	String	Sim	<p>Especifica o nome do recurso de computação. O nome deve corresponder ao nome do subdiretório dentro de <code>.amplify-hosting/compute-directory</code>.</p> <p>O único valor válido para a versão 1 da especificação de implantação é <code>default</code>.</p>
<code>runtime</code>	ComputeRuntime	Sim	<p>Define o runtime do recurso computacional provisionado.</p> <p>Os valores válidos são <code>nodejs16.x</code>, <code>nodejs18.x</code> e <code>nodejs20.x</code>.</p>
<code>entrypoint</code>	String	Sim	<p>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</p>

Chave	Tipo	Obrigatório	Descrição
			recurso computacional.

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

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

O JSON para o `computeResource` atributo terá a seguinte aparência.

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

Usando o `imageSettings` atributo

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

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

```
type ImageSettings = {
  sizes: number[];
  domains: string[];
  remotePatterns: RemotePattern[];
  formats: ImageFormat[];
  mininumCacheTTL: 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. Semelhante 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.
dangerouslyAllowSVG	Booleano	Sim	Permite a SVG entrada de

Chave	Tipo	Obrigatório	Descrição
			imagemURLs. 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 corresponde a qualquer número de subdomínios. O

Chave	Tipo	Obrigatório	Descrição
			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
<code>name</code>	String	Sim	O nome do framework.
<code>versão</code>	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:

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

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

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

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

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

Roteamento de pasta pública

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

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

1. Use um padrão de caminho para direcionar caminhos de solicitação que contenham extensões de arquivo. Por exemplo, 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 a solicitação retorne para o primitivo computacional quando houver um erro 404 do primitivo estático.

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

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

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

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

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

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

4. Especifique um fallback estático para a rota abrangente. Essa abordagem tem desvantagens que são descritas com mais detalhes na próxima seção [Roteamento abrangente de fallback](#).

Roteamento abrangente de fallback

Para SSR estruturas como Nuxt.js, por exemplo, em que uma rota abrangente é especificada para o destino 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 retornará para o primitivo estático e ainda terminará com um código de status 404, mas a página 404 não será renderizada.

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

Roteamento de caminho base

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

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

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

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

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

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

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

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**— Esta seção é opcional para um Express servidor, a menos que você esteja lidando com a otimização de imagens.
- **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 esta seção para especificar o tempo de execução e o ponto de entrada do seu Express servidor.

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 seu aplicativo no padrão URL fornecido pela Amplify Hosting.

Integração de otimização de imagem para criadores de frameworks

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

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

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

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

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

Entendendo a otimização de imagens API

A otimização de imagem pode ser invocada em tempo de execução por meio do URL domínio do aplicativo 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 consegue otimizar GIF APNG e SVG formatar nem convertê-los em outro formato.
- SVGas imagens não são exibidas a menos que a `dangerouslyAllowSVG` configuração esteja ativada.
- 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 é o único protocolo compatível com o fornecimento de imagens com controle remotoURLs.

HTTPcabeçalhos

O HTTP cabeçalho Aceitar solicitação é usado para especificar os formatos de imagem, expressos como MIME tipos, permitidos pelo cliente (geralmente 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.

URI parâmetros de solicitação

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

Parâmetro de consulta	Tipo	Obrigatório	Descrição	Exemplo
url	String	Sim	Um caminho relativo ou absoluto URL para a imagem de origem. Para um controle remotoURL, os protocolos http e https são suportados. O valor deve ser URL codificado.	?url=http%3A%2F%2Fwww.example.com%2Fbuffalo.png
width	Número	Sim	A largura da imagem otimizada em pixels.	?width=800
height	Número	Não	A altura da imagem otimizada em pixels. Se não for especificada, a imagem passará por ajuste de escala automático para	?height=600

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

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

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

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

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

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

Código de status de resposta.

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

Sucesso - código de HTTP status 200

A solicitação foi atendida com sucesso.

BadRequest - código de HTTP status 400

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

Não encontrado - código HTTP de status 404

A imagem de origem não foi encontrada.

Conteúdo muito grande - código de HTTP status 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 no nosso CDN para que as solicitações subsequentes à mesma imagem, com os mesmos parâmetros de consulta, sejam atendidas a partir do cache. O cache Time to live (TTL) é controlado pelo `Cache-Control` cabeçalho. 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 uma imagem remota URL responder com `Cache-Control s-max-age=10`, mas o valor de `minimumCacheTTL` for 60, então 60 será usado.

Usando adaptadores de código aberto

Você pode usar qualquer adaptador de construção de SSR estrutura 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 [Implemente um SSR aplicativo no Amplify](#)

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 um nome como. URL `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 domínio personalizado URL, 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 serviço web do 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](#). 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, consulte [Usando TLS certificados SSL](#).

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, consulte [Começando a implantar um aplicativo no Amplify Hosting](#).

- Você tem um conhecimento básico de domínios e DNS terminologia.

Para obter mais informações sobre domínios e DNS, consulte [Compreender a DNS terminologia e os conceitos](#).

Tópicos

- [Compreender a DNS terminologia e os conceitos](#)

- [Usando TLS certificadosSSL/](#)
- [Adicione um domínio personalizado gerenciado pelo Amazon Route 53](#)
- [Adicione um domínio personalizado gerenciado por um DNS provedor terceirizado](#)
- [Atualizar DNS registros para um domínio gerenciado pelo GoDaddy](#)
- [Atualizar DNS registros de um domínio gerenciado pelo Google Domains](#)
- [Atualizar o TLS certificadoSSL/para um domínio](#)
- [Gerenciar subdomínios](#)
- [Subdomínios Wildcard](#)
- [Configure subdomínios automáticos para um domínio personalizado do Amazon Route 53](#)
- [Solucionar problemas de domínios personalizados](#)

Compreender a DNS terminologia e os conceitos

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 do DNS

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

CNAME

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

ANAME

Um ANAME registro é como um CNAME registro, mas no nível raiz. An ANAME aponta a raiz do seu domínio para um FQDN. Isso 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.

DNSverificação

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 DNS provedor para encontrar o endereço IP do servidor que hospeda o site.

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

O Amplify usa um CNAME registro para verificar se você é proprietário do seu 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 DNS as configurações do seu domínio e adicionar um novo CNAME registro 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.

SSL/TLScriação

Se você estiver usando um certificado gerenciado, AWS Amplify emite um TLS certificadoSSL/ para configurar um domínio personalizado seguro.

SSL/TLSconfiguração e verificação

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 DNS o registro de verificação. Para domínios gerenciados fora do Route 53, você deve adicionar

manualmente o registro de DNS verificação fornecido no console do Amplify ao seu domínio com um DNS provedor 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 CNAME os registros fornecidos no console do Amplify ao seu domínio com um DNS provedor terceirizado.

Usando TLS certificadosSSL/

Um TLS certificadoSSL/é um documento digital que permite que os navegadores da Web identifiquem e estabeleçam conexões de rede criptografadas com sites usando o TLS protocoloSSL/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 TLS certificadoSSL/para todos os domínios conectados ao seu aplicativo, para que todo o tráfego seja protegido por /2. HTTPS O certificado padrão gerado por 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 pode renovar o certificado se o registro CNAME de verificação tiver sido modificado ou excluído nas DNS configurações do 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. ACMé um serviço que permite provisionar, gerenciar e implantar com facilidade TLS certificados SSL públicos e privados para uso com Serviços da AWS seus recursos internos conectados. Certifique-se de solicitar ou importar o certificado na região Leste dos EUA (Norte da Virgínia) (us-east-1).

Certifique-se de que seu certificado personalizado cubra todos os subdomínios que você planeja adicionar. 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 `product.example.com` e `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 o 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, o ACM 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 TLS certificado SSL/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 TLS certificadoSSL/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 SSLCertificado personalizado.
 - b. Selecione o certificado a ser usado na lista.
9. Escolha Adicionar domínio.

 Note

Pode levar até 24 horas para que o DNS se propague e emita o certificado. Para obter ajuda na resolução de erros que ocorrem, consulte [Solucionar problemas de domínios personalizados](#).

Adicione um domínio personalizado gerenciado por um DNS provedor terceirizado

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

Se você estiver usando o GoDaddy Google Domains, consulte [the section called “Atualizar DNS registros para um domínio gerenciado pelo GoDaddy”](#) ou consulte [the section called “Atualizar DNS](#)

[registros de um domínio gerenciado pelo Google Domains](#)” os procedimentos específicos desses provedores.

Para adicionar um domínio personalizado gerenciado por um DNS provedor terceirizado

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 DNS provedor e adicione os servidores de nomes às suas DNS configurações.
 - d. Selecione Eu adicionei os servidores de nomes acima ao meu registro de domínio.
 - e. Prossiga para a etapa sete.
 - Para continuar com a configuração manual
 - a. Escolha a configuração manual
 - b. Escolha Configurar domínio.
 - c. Prossiga para a etapa sete.
7. Por padrão, o Amplify cria automaticamente duas entradas de subdomínio para seu domínio. Por exemplo, se seu nome de domínio for `exemplo.com`, você verá os subdomínios `https://www.exemplo.com` e `https://exemplo.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 TLS certificadoSSL/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 SSLCertificado 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 DNS registros com seu provedor de domínio terceirizado.

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

DNS Records

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

11. Execute um destes procedimentos:
 - Se você estiver usando GoDaddy, acesse [Atualizar DNS registros para um domínio gerenciado pelo GoDaddy](#).

- Se você estiver usando o Google Domains, acesse [Atualizar DNS registros de um domínio gerenciado pelo Google Domains](#).
 - Se você estiver usando um DNS provedor terceirizado diferente, vá para a próxima etapa deste procedimento.
12. Acesse o site do seu DNS provedor, faça login na sua conta e localize as configurações DNS de gerenciamento do seu domínio. Você configurará dois CNAME registros.
 13. Configure o primeiro CNAME registro para apontar seu subdomínio para o servidor de AWS validação.

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

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

DNS Records

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

Se o console do Amplify exibir um registro do servidor de ACM validação, como `_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`, insira o valor do registro. **`_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`** CNAME

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

DNS Records ×

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

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

Note

O certificado padrão do Amplify gerado por 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 CNAME verificação 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 CNAME registros 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 CNAME registro 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, recomendamos que você atualize esse CNAME registro depois que o status do seu domínio for exibido, como `AVAILABLE` no console do Amplify.

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

DNS Records ×

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

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

Note

A verificação da propriedade e DNS propagação do domínio 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 DNS registros para um domínio gerenciado pelo GoDaddy

Para adicionar um domínio personalizado gerenciado por GoDaddy

1. Antes de atualizar seus DNS registros com GoDaddy, conclua as etapas de um a nove do procedimento [the section called “Adicione um domínio personalizado gerenciado por um DNS provedor terceirizado”](#).
2. Faça login na sua GoDaddy conta.
3. Na sua lista de domínios, encontre o domínio a ser adicionado e escolha Gerenciar DNS.
4. Na DNS página, GoDaddy exibe uma lista de registros do seu domínio na seção DNSRegistros. Você precisa adicionar dois novos CNAME registros.
5. Crie o primeiro CNAME registro para direcionar seus subdomínios para o domínio Amplify.
 - a. Na seção DNSRegistros, escolha Adicionar novo registro.
 - b. Em Type (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 Valor, veja seus DNS registros no console do Amplify e insira o valor. Se o console do Amplify exibir o domínio do seu aplicativo como `d11111abcdef8.cloudfront.net`, insira Value. **d11111abcdef8.cloudfront.net**

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

DNS Records ×

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

- e. Escolha Salvar.
6. Crie o segundo CNAME registro para apontar para o servidor de validação AWS Certificate Manager (ACM). Um único certificado validado ACM gera um TLS certificado SSL/para seu domínio.
 - a. Em Type (Tipo), escolha CNAME.
 - b. Em Nome, insira o subdomínio.

Por exemplo, se o DNS registro 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.

DNS Records

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

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

Por exemplo, se o servidor de validação for `_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`, digite `_cjhvou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws` para Valor.

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

DNS Records

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

- d. Escolha Salvar.

Note

O certificado padrão do Amplify gerado por 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 CNAME verificação 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 ANAME ALIAS /records. Para DNS provedores que não têm ALIAS suporte ANAME /, é altamente recomendável migrar seu DNS para o Amazon Route 53. Para obter mais informações, consulte [Configurando o Amazon Route 53 como seu DNS serviço](#).

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

- a. Na DNSpágina, 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 Destino. URL
- d. Em Tipo de encaminhamento, escolha Temporário (302).
- e. Escolha Salvar.

Atualizar DNS registros de um domínio gerenciado pelo Google Domains

Para adicionar um domínio personalizado gerenciado pelo Google Domains

1. Antes de atualizar seus DNS registros com o Google Domains, conclua as etapas de um a nove do procedimento [Para adicionar um domínio personalizado gerenciado por um DNS provedor 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 esquerdo, escolha DNS. O Google exibe os Registros de recursos do seu domínio. Você precisa adicionar dois novos CNAME registros.
5. Crie o primeiro CNAME registro 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. Em Type (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.

DNS Records ×

Verify records in your domain registrar match these records.

Verification record

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

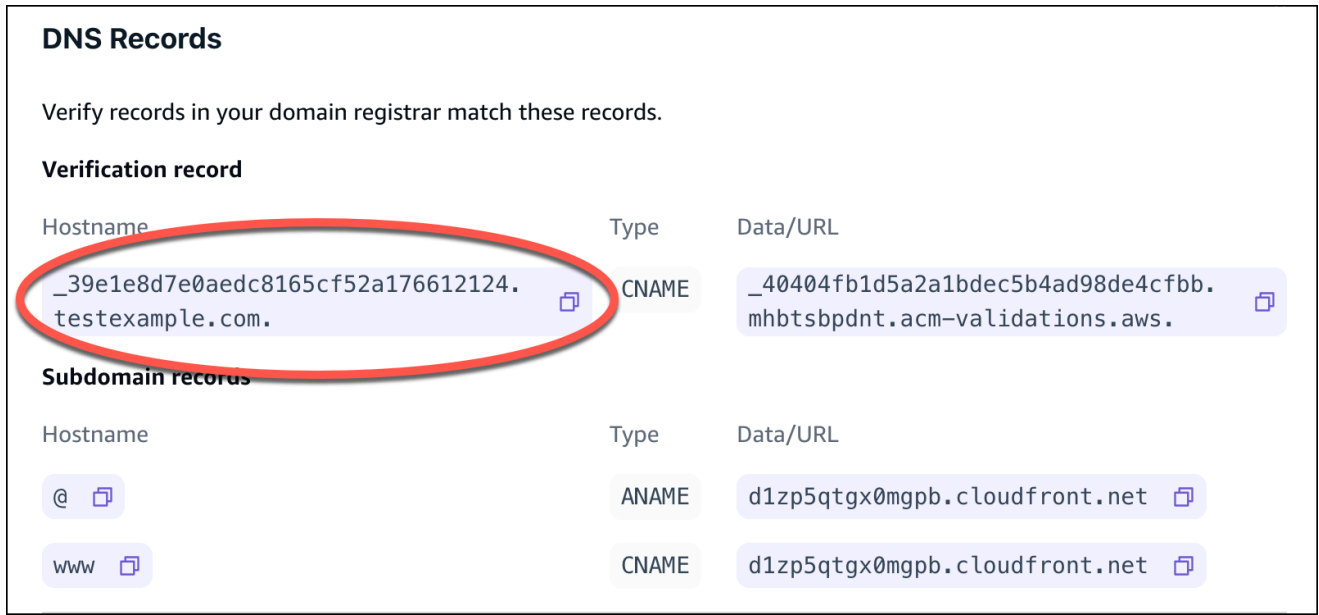
Subdomain records

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

6. Crie o segundo CNAME registro para apontar para o servidor de validação AWS Certificate Manager (ACM). Um único certificado validado ACM gera um TLS certificado SSL/para seu domínio.
 - a. Em Nome do host, insira o subdomínio.

Por exemplo, se o DNS registro 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.



DNS Records

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

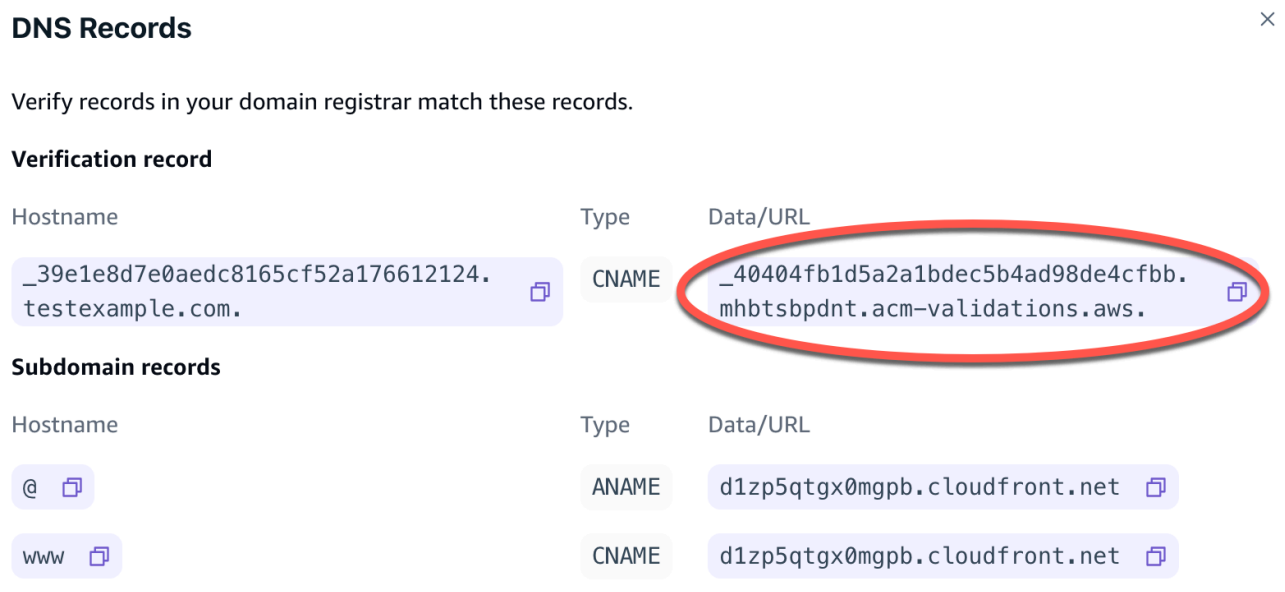
- b. Em Type (Tipo), escolha CNAME.
- c. Em Dados, insira o certificado de ACM validação.

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 ACM verificação a ser usado.



DNS Records

Verify records in your domain registrar match these records.

Verification record

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

Subdomain records

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

7. Escolha Salvar.

Note

O certificado padrão do Amplify; gerado por 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 CNAME verificação tiver sido modificado ou excluído. Você deve excluir e adicionar o domínio novamente no console do Amplify.

8. O suporte do Google Domains para ANAME ALIAS /records está em versão prévia. Para DNS provedores que não têm ALIAS suporte ANAME /, é altamente recomendável migrar seu DNS para o Amazon Route 53. Para obter mais informações, consulte [Configurando o Amazon Route 53 como seu DNS serviço](#). 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 DNS nas configurações de um domínio do Google podem levar até 48 horas para entrarem em vigor. Para obter ajuda na resolução de erros que ocorrem, consulte [Solução de problemas de domínios personalizados](#).

Atualizar o TLS certificadoSSL/para um domínio

Você pode alterar o TLS certificadoSSL/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 SSL TLS /certificates](#).

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 SSLCertificado 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 SSLCertificado personalizado.
 - b. Selecione o certificado a ser usado na lista.
 - Para alterar um certificado personalizado para um certificado personalizado diferente
 - Em SSLCertificado 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 SSL criação de um certificado gerenciado ou o processo de configuração de um certificado personalizado.

Gerenciar subdomínios

Um subdomínio é a parte do seu URL que aparece antes do seu nome de 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 um URL que termine com `example.com` será roteado 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 Remove 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 AWS Amplify console, consulte o capítulo de solução [Solucionar problemas de domínios personalizados](#) de problemas do Amplify. Se você não encontrar uma solução para o problema, entre em contato com AWS Support. Para obter mais informações, consulte [Criação de um caso de suporte](#) no Guia do usuário do AWS Support .

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
  configFile: *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. 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` com `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 ramificações do Git para criar uma nova implantação sempre que você conecta uma nova ramificação ao seu repositório. Depois de conectar sua primeira ramificação, você cria ramificações de 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

- [Fluxos de trabalho de equipe com aplicativos Amplify Gen 2 completos](#)
- [Fluxos de trabalho de equipe com aplicativos Amplify Gen 1 completos](#)
- [Implantações de ramificação de atributo com base em padrão](#)
- [Geração automática em tempo de construção da configuração do Amplify \(somente aplicativos de primeira geração\)](#)
- [Compilações condicionais de back-end \(somente aplicativos de primeira geração\)](#)

- [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 que prioriza a definição de 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.

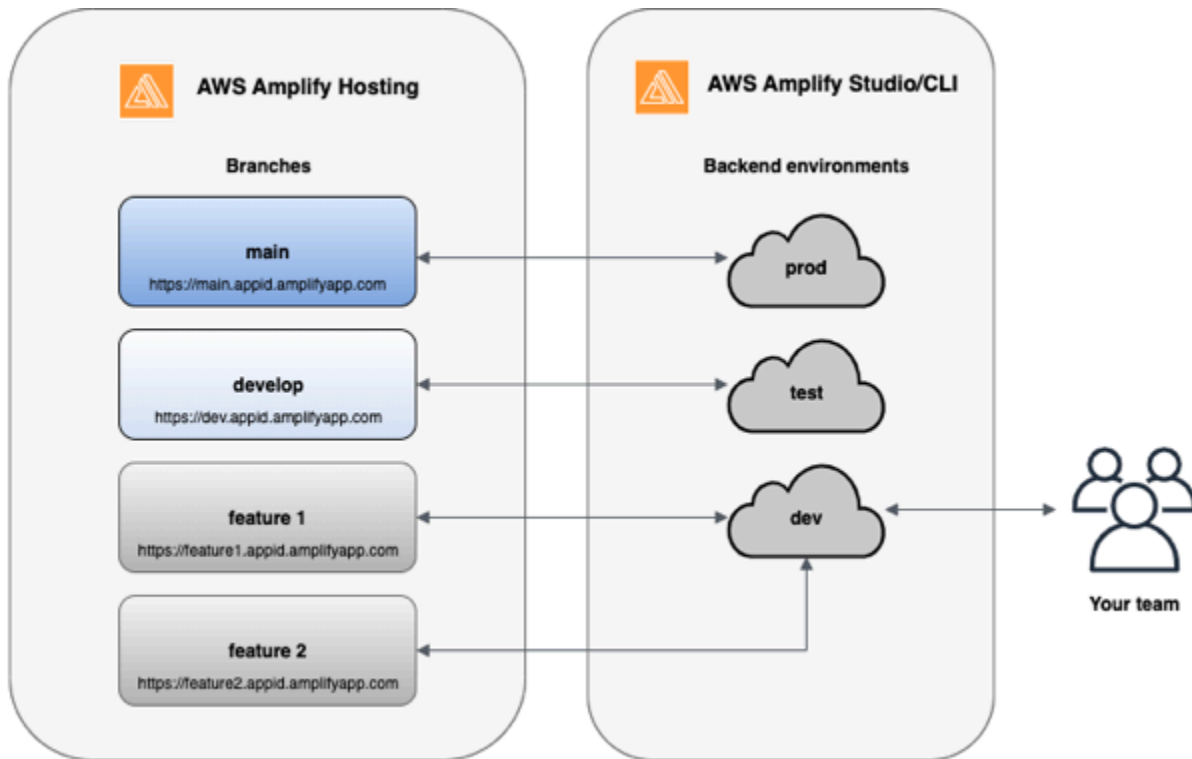
Fluxos de trabalho de equipe com aplicativos Amplify Gen 1 completos

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 pelo Amplify to. CLI AWS Para saber como configurar esse cenário de implantação, consulte [Construindo um back-end para um aplicativo](#).

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

Fluxo de trabalho de ramificação de atributo

- Crie ambientes de back-end de produção, teste e desenvolvimento com o Amplify Studio ou o 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 o 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.


quick-notes

The app homepage lists all deployed frontend and backend environments.

Frontend environments | **Backend environments**

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

prod



Categories added


- Authentication
- API

Deployment status

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

▶ Edit backend

test



Categories added


- Authentication
- API

Deployment status

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

▶ Edit backend

dev



Categories added

- Authentication
- API

Deployment status

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

▶ Edit backend

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

conceda a função de serviço ao Amplify. Escolha Salvar e implantar. Depois que a compilação for concluída, haverá uma implantação de ramificação principal disponível em <https://master.appid.amplifyapp.com>.

Configure build settings

App build settings

App name
Pick a name for your app.

Name cannot contain periods

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

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

Yes - choose an existing environment or create a new one

Create new environment

Select environment


- dev
- test
- prod

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.

Add repository branch

AWS CodeCommit

Repository service provider

 AWS CodeCommit

Branch
Select a branch from your repository.

develop

Backend environment
Select a backend environment for this branch.

test

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 no AmplifyCLI), você pode conectar uma ramificação diretamente do seu terminal. <region><region> Você pode encontrar seu aplicativo acessando Configurações do aplicativo > Geral > Aplicativo ARN: arn:aws:amplify: ::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').

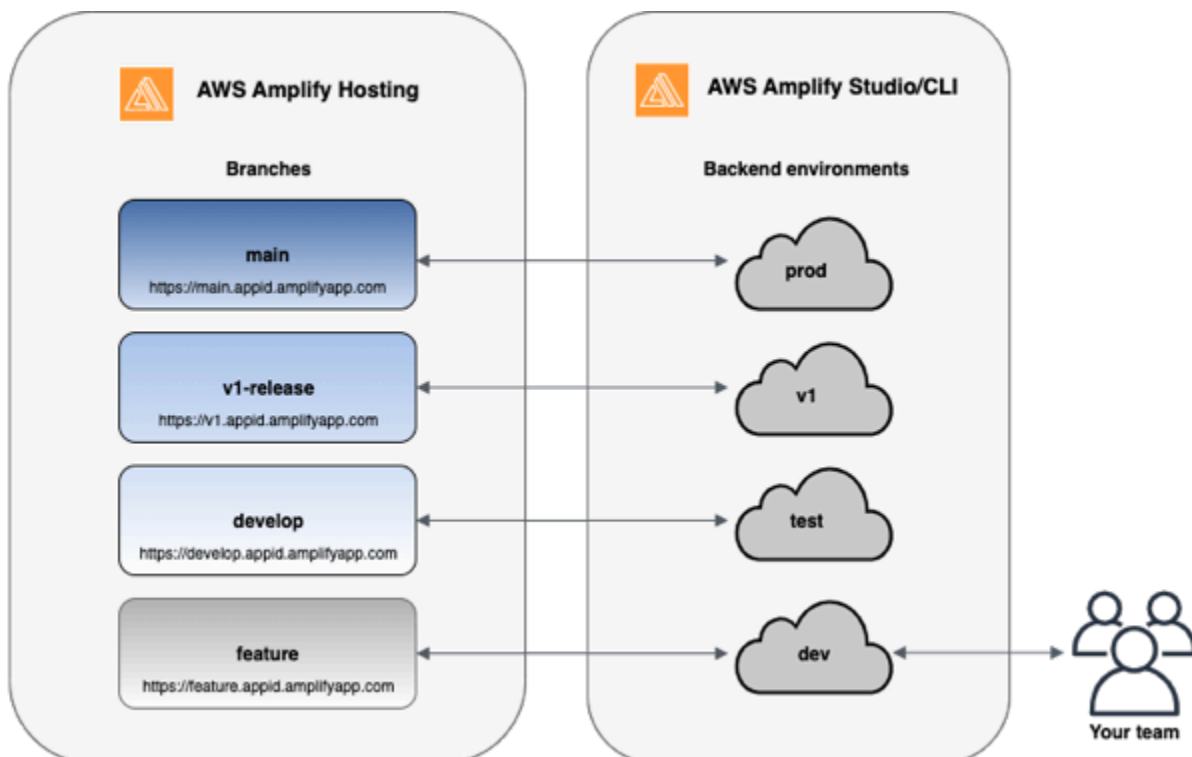
```
git push origin --delete newinternet
```

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

GitFlow fluxo de trabalho

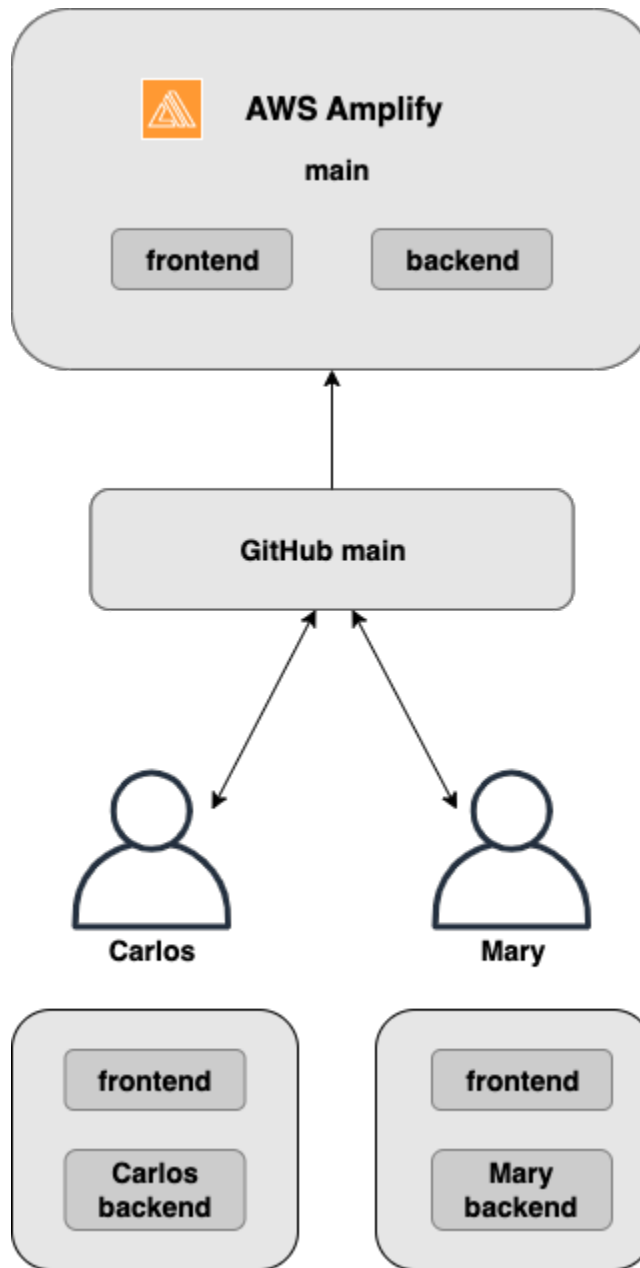
GitFlow usa duas ramificações para registrar o histórico do projeto. A ramificação principal rastreia somente o código de lançamento, e a ramificação de desenvolvimento é usada como uma ramificação de integração para novos recursos. GitFlow simplifica o desenvolvimento paralelo ao isolar o novo desenvolvimento 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 o 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 detectará ambientes de back-end criados pelo 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 um compartilhável. URL [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 a 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)

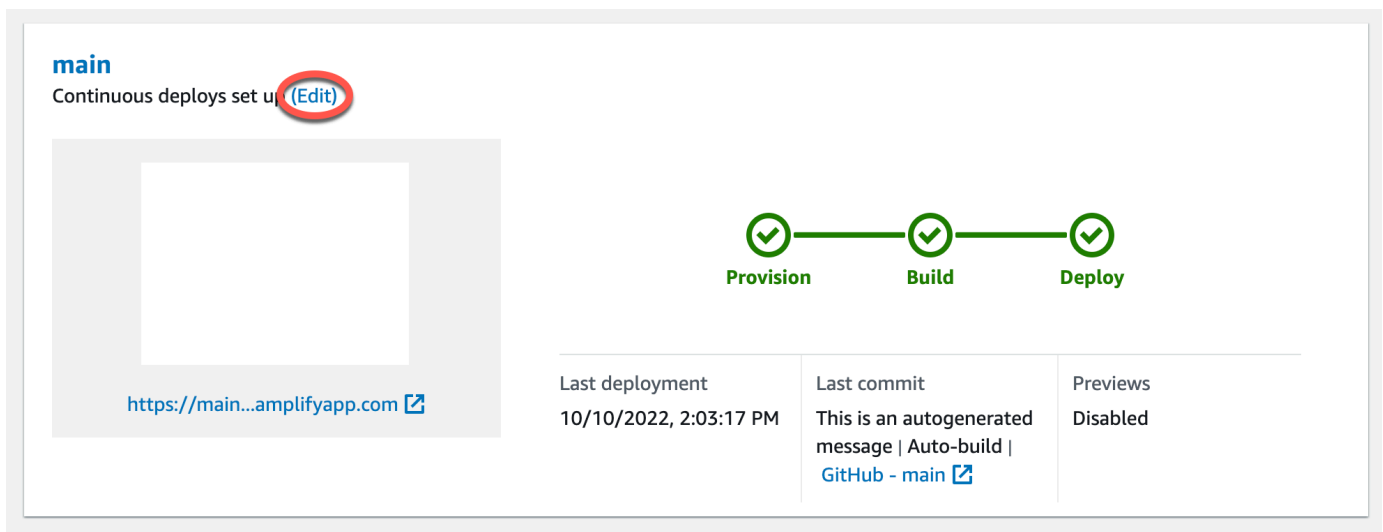
Note

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 no 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.



Last deployment 10/10/2022, 2:03:17 PM	Last commit This is an autogenerated message Auto-build GitHub - main	Previews Disabled
---	--	----------------------

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.

Edit target backend

Select a backend environment to use with this branch

App name

Example-Amplify-App (this app) ▼

Environment

dev ▼

Enable full-stack continuous deployments (CI/CD)

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

6. 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 e que prioriza o código. Portanto, esse recurso não é necessário para back-ends de 2ª 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 no 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 no 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.

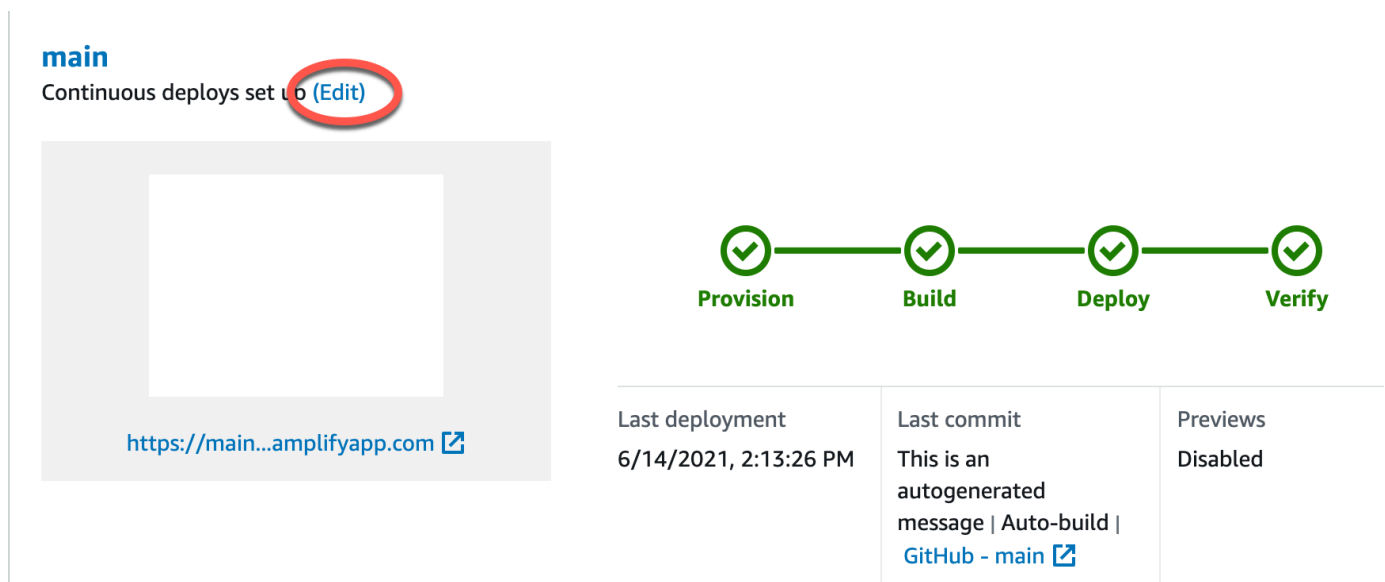
10. Escolha Próximo.

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 no 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.



main
Continuous deploys set up (Edit)

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

Provision Build Deploy Verify

Last deployment 6/14/2021, 2:13:26 PM	Last commit This is an autogenerated message Auto-build GitHub - main	Previews Disabled
--	---	----------------------

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](#)
- [Crie um back-end para um aplicativo de primeira geração](#)

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, consulte [Configurar 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](#).

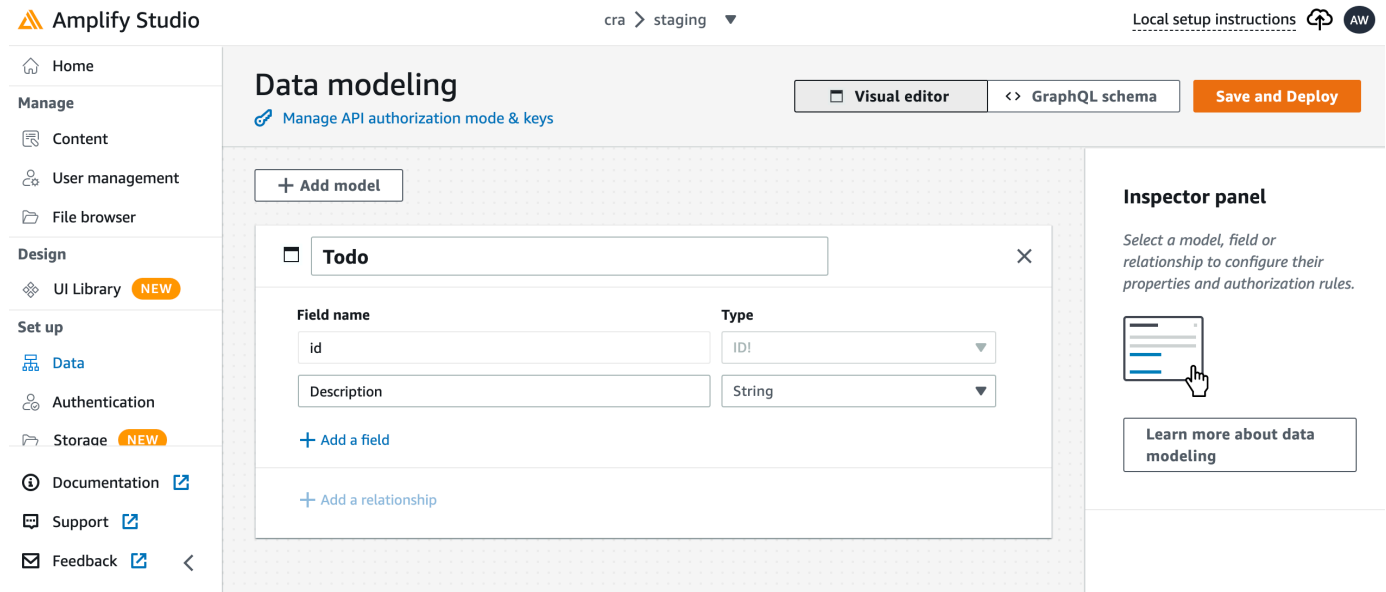
Use as instruções a seguir para criar um banco de dados simples usando a interface do construtor visual do Amplify Studio.

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.

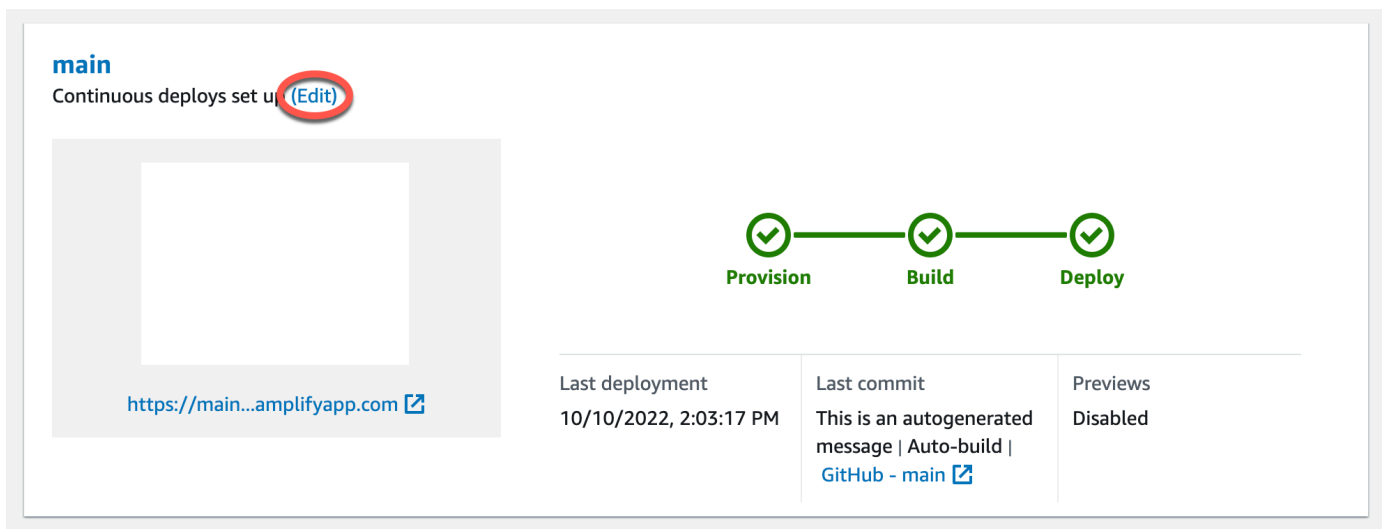
```
amplify pull --appId abcd1234 --envName staging
```

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 [Implantação de 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">  
  
```


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.

ℹ 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
  configFile: '**/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 ambiente `USER_DISABLE_TESTS` 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.

Environment Variables Manage variables

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

Branch	Variable	Value
All branches	USER_DISABLE_TESTS	True

Rows per page: 15 << < 1 > >>

Para desativar os testes para uma ramificação específica, adicione a variável de ambiente `USER_DISABLE_TESTS` 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.

Environment Variables

Manage variables

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

Branch	Variable	Value
All branches	USER_DISABLE_TESTS	False
main	USER_DISABLE_TESTS	True

Rows per page 15

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
<code>/original.html</code>	<code>/destination.html</code>	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
<code>/docs/<*></code>	<code>/documents/<*></code>	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
<code>/<*></code>	<code>/index.html</code>	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
<code></^[^.]�+\$ \.(?!(css gif ico jpg js png txt svg woff woff2 ttf map json webp))\$)([^\.]�+\$)/></code>	<code>/index.html</code>	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
<code>/images/<*></code>	<code>https://images.otherdomain.com/<*></code>	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
<code>/about</code>	<code>/about</code>	<code>/about.html</code>
<code>/about</code> (when <code>about.html</code> returns 404)	<code>/about/</code>	<code>/about/index.html</code>
<code>/about/</code>	<code>/about/</code>	<code>/about/index.html</code>

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
<code>/docs/<year>/<month>/<date>/<itemid></code>	<code>/documents/<year>/<month>/<date>/<itemid></code>	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 redirecionamento	Código do país
/docs/<*>/content/<*>	/documents/<*>/content/<*>	permanent redirect (301)	

Restringindo o acesso às filiais de um aplicativo Amplify

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 URL a 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).

O Amplify acelera solicitações com falha que estão tentando acessar recursos protegidos por senha. Esse comportamento protege os aplicativos contra ataques de dicionário ou outras tentativas de ler dados por trás dos controles de acesso.

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 (SSR) do lado do servidor, 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
<code>_BUILD_TIMEOUT</code>	A duração do tempo limite de compilação em minutos	30
<code>_LIVE_UPDATES</code>	A ferramenta será atualizada para a versão mais recente.	<code>[{"name": "Amplify CLI", "pkg": "@aws-amplify/cli", "type"</code>

Nome da variável	Descrição	Valor de exemplo
		<code>:"npm", "version": "latest"]]</code>
USER_DISABLE_TESTS	<p>A etapa de teste é ignorada durante uma compilação. É possível desativar os testes para todas as ramificações ou ramificações específicas em um aplicativo.</p> <p>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.</p>	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 de recurso da Amazon (ARN) da filial da compilação atual	aws:arn:amplify:us-west-2:123456789012:appname/branch/...
AWS_CLONE_URL	O clone URL usado para buscar o conteúdo do repositório git	git@github.com:<user-name>/<repo-name>.git
AWS_COMMIT_ID	<p>O ID de confirmação da compilação atual</p> <p>“HEAD” para reconstruções</p>	abcd1234

Nome da variável	Descrição	Valor de exemplo
AWS_JOB_ID	<p>O ID de trabalho da compilação atual.</p> <p>Isso inclui algum preenchimento com "0" para que sempre tenha o mesmo comprimento.</p>	0000000001
AWS_PULL_REQUEST_ID	<p>O ID do pull request da compilação de pré-visualização da web do pull request.</p> <p>Essa variável de ambiente não está disponível ao ser usada AWS CodeCommit como seu provedor de repositório.</p>	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 ter 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 ter 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 ter 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 ou 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	<p>O ID do cliente do aplicativo a ser usado por aplicativos da web</p> <p>O cliente do aplicativo deve ser configurado com acesso ao grupo de usuários do Amazon Cognito especificado pela variável de ambiente <code>AMPLIFY_USERPOOL_ID</code>.</p>	123456
AMPLIFY_NATIVECLIENT_ID	<p>O ID do cliente do aplicativo a ser usado por aplicativos nativos</p> <p>O cliente do aplicativo deve ser configurado com acesso ao grupo de usuários do Amazon Cognito especificado pela variável de ambiente <code>AMPLIFY_USERPOOL_ID</code>.</p>	123456
AMPLIFY_IDENTITYPOOL_ID	O ID do banco de identidades do Amazon Cognito	example-identitypool-id

Nome da variável	Descrição	Valor de exemplo
AMPLIFY_PERMISSION S_BOUNDARY_ARN	O ARN para que a IAM política seja usada como um limite de permissões que se aplica a todas as IAM funções criadas pelo Amplify. Para obter mais informações, consulte Limite de IAM permissões para funções geradas pelo Amplify .	arn:aws:iam::123456789012:policy/example-policy
AMPLIFY_DESTRUCTIV E_UPDATES	Defina essa variável de ambiente como true para permitir que um GraphQL API seja atualizado com operações de esquema que podem potencialmente causar perda de dados.	true

Note

As variáveis de AMPLIFY_AMAZON_CLIENT_SECRET ambiente AMPLIFY_AMAZON_CLIENT_ID e são OAuth tokens, não uma chave de AWS acesso e uma chave secreta.

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 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 no Manual](#) do GNU Bash.

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

1. Faça login 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 API token (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 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.

Select a backend environment to use with this branch

App name

Environment

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

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

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

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

- 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 tempo de execução 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 API chaves, 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 Armazenamento de parâmetros do AWS Systems Manager, 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. Em Fonte da KMS chave, 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 JSON string.

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	<code>com.yourapp.auth</code>

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](#)
- [Logs de acesso](#)

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	<p>O número total de solicitações de visualizações recebidas pelo seu aplicativo.</p> <p>A estatística mais relevante é Sum. Use a estatística Sum para obter o número total de solicitações.</p>

Metrics	Descrição
BytesDownloaded	<p>A quantidade total de dados transferidos do seu aplicativo (baixados) em bytes pelos visualizadores para as solicitações GET, HEAD e OPTIONS.</p> <p>A estatística mais relevante é Sum.</p>
BytesUploaded	<p>A quantidade total de dados transferidos para seu aplicativo (enviados) em bytes usando as solicitações POST e PUT.</p> <p>A estatística mais relevante é Sum.</p>
4XXErrors	<p>O número de solicitações que retornaram um erro no intervalo do código de status HTTP 400-499.</p> <p>A estatística mais relevante é Sum. Use a estatística Sum para obter o total de ocorrências desses erros.</p>
5XXErrors	<p>O número de solicitações que retornaram um erro no intervalo do código de status HTTP 500-599.</p> <p>A estatística mais relevante é Sum. Use a estatística Sum para obter o total de ocorrências desses erros.</p>

Metrics	Descrição
Latência	<p>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.</p> <p>As estatísticas mais relevantes são Average, Maximum, Minimum, p10, p50, p90, p95 e p100.</p> <p>Use a estatística Average para avaliar as latências esperadas.</p>

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 [Implantação de 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 logs 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](#)
- [Atualizações de pacote ao vivo](#)

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-exemplerepo`, você digitaria **`public.ecr.aws/xxxxxxxx/ecr-exemplerepo`**.
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.

Gerenciando a configuração de cache de um aplicativo

O Amplify usa CloudFront a Amazon para gerenciar a configuração de cache de seus aplicativos hospedados. Uma configuração de cache é aplicada a cada aplicativo para otimizar o desempenho.

Em 13 de agosto de 2024, o Amplify lançou melhorias na eficiência do cache para aplicativos. Para obter mais informações, consulte [Melhorias no CDN cache para melhorar o desempenho do aplicativo com AWS Amplify hospedagem](#).

A tabela a seguir resume o suporte do Amplify para comportamentos específicos de armazenamento em cache antes e depois do lançamento dos aprimoramentos de cache.

Comportamento de armazenamento em cache	Suporte anterior	Com melhorias no cache
Você pode adicionar cabeçalhos personalizados para um aplicativo no console do Amplify ou em <code>customHeaders.yaml</code> um arquivo. Um dos cabeçalhos que você pode substituir é <code>Cache-Control</code> . Para obter mais informações, consulte Cabeçalhos personalizados .	Sim	Sim
O Amplify respeita os <code>Cache-Control</code> cabeçalhos que você define em um <code>customHeaders.yaml</code> arquivo e eles têm precedência sobre as configurações de cache padrão do Amplify.	Sim	Sim
O Amplify respeita os <code>Cache-Control</code> cabeçalhos definidos na estrutura de	Sim	Sim

Comportamento de armazenamento em cache	Suporte anterior	Com melhorias no cache
um aplicativo para rotas dinâmicas (por exemplo, rotas Next.js). SSR Se um <code>Cache-Control</code> cabeçalho for definido no <code>customHeaders.yaml</code> arquivo do aplicativo, isso terá precedência sobre as configurações no <code>next.config.js</code> arquivo.		
Cada nova implantação do aplicativo CI/CD limpa o cache.	Sim	Sim
Você pode ativar o modo de desempenho para um aplicativo.	Sim	Não A configuração do modo de desempenho não está mais disponível no console do Amplify. No entanto, você pode criar um <code>Cache-Control</code> cabeçalho que defina a <code>s-maxage</code> diretiva. Para obter instruções, consulte Gerenciar desempenho do aplicativo .

A tabela a seguir lista as alterações nos valores padrão para configurações específicas de cache.

Configuração de cache	Valor padrão anterior	Valor padrão com melhorias no cache
Duração do cache para ativos estáticos	Dois segundos	Um ano

Configuração de cache	Valor padrão anterior	Valor padrão com melhorias no cache
Duração do cache para respostas de proxy reverso	Dois segundos	Zero segundos (sem cache)
Tempo máximo de vida (TTL)	Dez minutos	Um ano

Para obter mais informações sobre como o Amplify determina a configuração de cache a ser aplicada a um aplicativo e instruções sobre como gerenciar a configuração da chave de cache, consulte os tópicos a seguir.

Tópicos

- [Como o Amplify aplica a configuração de cache a um aplicativo](#)
- [Gerenciando cookies de chave de cache](#)

Como o Amplify aplica a configuração de cache a um aplicativo

Para gerenciar o armazenamento em cache do seu aplicativo, o Amplify determina o tipo de conteúdo que está sendo veiculado examinando o tipo de plataforma do aplicativo e as regras de reescrita. Para Compute aplicativos, o Amplify também examina as regras de roteamento no manifesto de implantação.

Note

O tipo de plataforma do aplicativo é definido pela Amplify Hosting durante a implantação. Um aplicativo SSG (estático) é definido para o tipo de plataforma `WEB`. Um SSR aplicativo (Next.js 12 ou posterior) está configurado para o tipo de plataforma `WEB_COMPUTE`.

O Amplify identifica os quatro tipos de conteúdo a seguir e aplica a política de cache gerenciado especificada.

Estático

O conteúdo veiculado por aplicativos com a `WEB` plataforma ou as rotas estáticas em um `WEB_COMPUTE` aplicativo.

Esse conteúdo usa a política de Amplify-StaticContent cache.

Otimização de imagem

As imagens veiculadas pelas ImageOptimization rotas em um WEB_COMPUTE aplicativo.

Esse conteúdo usa a política de Amplify-ImageOptimization cache.

Computação

O conteúdo veiculado pelas Compute rotas em um WEB_COMPUTE aplicativo. Isso inclui todo o conteúdo renderizado () do lado do servidor. SSR

Esse conteúdo usa a política Amplify-Default ou a política de Amplify-DefaultNoCookies cache, dependendo do cacheConfig.type valor definido em seu AmplifyApp.

Proxy reverso

O conteúdo veiculado por caminhos que correspondem a uma regra personalizada de reescrita por proxy reverso. Para obter mais informações sobre como criar essa regra personalizada, consulte [Regravação de proxy reverso](#) o capítulo Usando redirecionamentos.

Esse conteúdo usa a política Amplify-Default ou a política de Amplify-DefaultNoCookies cache, dependendo do cacheConfig.type valor definido em seu AmplifyApp.

Entendendo as políticas de cache gerenciado do Amplify

O Amplify usa as seguintes políticas predefinidas de cache gerenciado para otimizar a configuração de cache padrão para seus aplicativos hospedados.

- Amplify-Default
- Amplify-DefaultNoCookies
- Amplify-ImageOptimization
- Amplify-StaticContent

Configurações padrão da política de cache gerenciado do Amplify

[Exibir essa política no CloudFront console](#)

Esta política foi projetada para uso com uma origem que é uma aplicação Web do [AWS Amplify](#).

Essa política tem as seguintes configurações:

- MínimoTTL: 0 segundos
- MáximoTTL: 31536000 segundos (um ano)
- PadrãoTTL: 0 segundos
- Cabeçalhos incluídos na chave de cache:
 - Authorization
 - Accept
 - CloudFront-Viewer-Country
 - Host
- Cookies included in cache key (Cookies incluídos na chave de cache): todos os cookies serão incluídos.
- Query strings included in cache key (Strings de consulta incluídas na chave de cache): todas as strings de consulta serão incluídas.
- Configuração de objetos compactados em cache: Gzip e Brotli ativados.

Amplify - configurações de política de cache DefaultNoCookies gerenciado

[Exibir essa política no CloudFront console](#)

Esta política foi projetada para uso com uma origem que é uma aplicação Web do [AWS Amplify](#).

Essa política tem as seguintes configurações:

- MínimoTTL: 0 segundos
- MáximoTTL: 31536000 segundos (um ano)
- PadrãoTTL: 0 segundos
- Cabeçalhos incluídos na chave de cache:
 - Authorization
 - Accept
 - CloudFront-Viewer-Country
 - Host
- Cookies incluídos na chave de cache: Nenhum cookie está incluído.

- Query strings included in cache key (Strings de consulta incluídas na chave de cache): todas as strings de consulta serão incluídas.
- Configuração de objetos compactados em cache: Gzip e Brotli ativados.

Amplify - configurações de política de cache ImageOptimization gerenciado

[Exibir essa política no CloudFront console](#)

Esta política foi projetada para uso com uma origem que é uma aplicação Web do [AWS Amplify](#).

Essa política tem as seguintes configurações:

- MínimoTTL: 0 segundos
- MáximoTTL: 31536000 segundos (um ano)
- PadrãoTTL: 0 segundos
- Cabeçalhos incluídos na chave de cache:
 - Authorization
 - Accept
 - Host
- Cookies incluídos na chave de cache: Nenhum cookie está incluído.
- Query strings included in cache key (Strings de consulta incluídas na chave de cache): todas as strings de consulta serão incluídas.
- Configuração de objetos compactados em cache: Gzip e Brotli ativados.

Amplify - configurações de política de cache StaticContent gerenciado

[Exibir essa política no CloudFront console](#)

Esta política foi projetada para uso com uma origem que é uma aplicação Web do [AWS Amplify](#).

Essa política tem as seguintes configurações:

- MínimoTTL: 0 segundos
- MáximoTTL: 31536000 segundos (um ano)
- PadrãoTTL: 0 segundos

- Cabeçalhos incluídos na chave de cache:
 - `Authorization`
 - `Host`
- Cookies incluídos na chave de cache: Nenhum cookie está incluído.
- Cadeias de caracteres de consulta incluídas na chave de cache: nenhuma sequência de caracteres de consulta está incluída.
- Configuração de objetos compactados em cache: Gzip e Brotli ativados.

Gerenciando cookies de chave de cache

Ao implantar seu aplicativo no Amplify, você pode escolher se deseja incluir ou excluir cookies na chave de cache. No console do Amplify, essa configuração é especificada na página de cabeçalhos e cache personalizados usando o botão Configurações da chave de cache. Para obter instruções, consulte [Incluindo ou excluindo cookies da chave de cache](#).

Incluir cookies na chave de cache

Essa é a configuração de cache padrão. Com essa configuração, o Amplify escolhe automaticamente uma configuração de cache ideal para seu aplicativo com base no tipo de conteúdo que está sendo veiculado.

Se você estiver usando o SDKs ou o AWS CLI, essa configuração corresponde `cacheConfig.type` à configuração `AMPLIFY_MANAGED` com o `CreateApp` ou `UpdateAppAPIs`.

Excluir cookies da chave de cache

Essa configuração de cache é semelhante à configuração padrão, exceto pelo fato de excluir todos os cookies da chave de cache. Você deve escolher explicitamente esse tipo de configuração de cache.

Optar por excluir cookies da chave de cache pode resultar em melhor desempenho do cache. No entanto, antes de escolher essa configuração de cache, é importante considerar se seu aplicativo usa cookies para veicular conteúdo dinâmico.

Se você estiver usando o SDKs ou o AWS CLI, essa configuração corresponde à configuração do `cacheConfig.type` para `AMPLIFY_MANAGED_NO_COOKIES` com o `CreateApp` ou `UpdateAppAPIs`.

Para obter mais informações sobre a chave de cache, consulte [Entenda a chave de cache](#) no Amazon CloudFront Developer Guide;

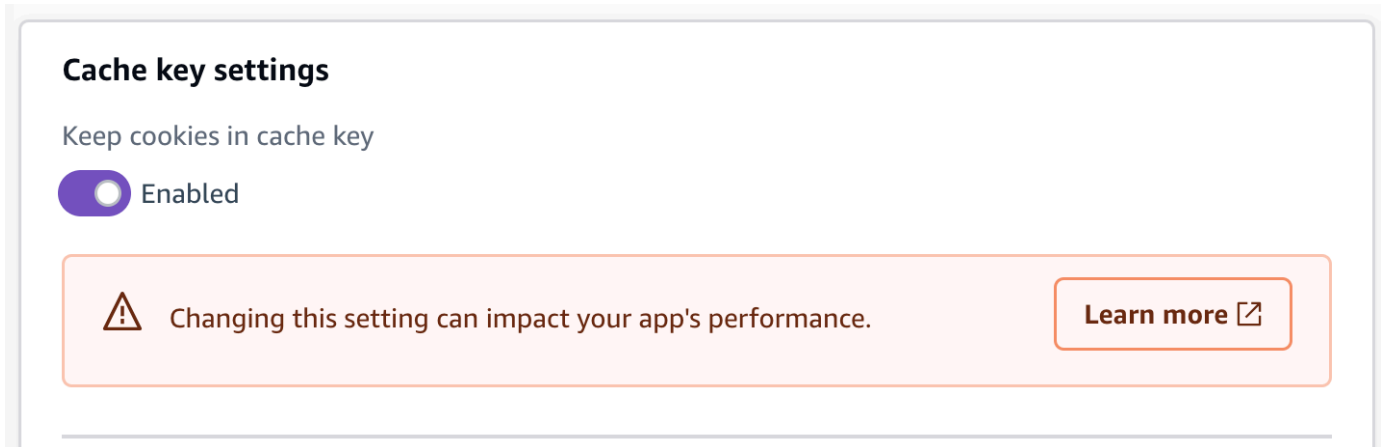
Incluindo ou excluindo cookies da chave de cache

Você pode definir a configuração do cookie da chave de cache para um aplicativo no console do Amplify ou no SDKs AWS CLI

Use o procedimento a seguir para especificar se deseja incluir ou excluir cookies da chave de cache ao implantar um novo aplicativo usando o console do Amplify.

Para definir a configuração do cookie da chave de cache ao implantar um aplicativo no Amplify

1. Faça login no 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. Se o aplicativo exigir uma função de IAM serviço, você pode permitir que a computação do Amplify Hosting crie automaticamente uma função de serviço para você ou pode especificar uma função que você 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 Configurações avançadas e, em seguida, localize a seção Configurações da chave de cache.
7. Escolha Manter cookies na chave de cache ou Remover cookies da chave de cache. A captura de tela a seguir mostra a opção de configurações da tecla Cache no console.



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

Alterando a configuração do cookie da chave de cache para um aplicativo

Você pode alterar a configuração do cookie da chave de cache para um aplicativo que já está implantado no Amplify. Use o procedimento a seguir para alterar a inclusão ou exclusão de cookies da chave de cache de um aplicativo usando o console do Amplify.

Para alterar a configuração do cookie da chave de cache para um aplicativo implantado

1. Faça login no AWS Management Console e abra o console do [Amplify](#).
2. Na página Todos os aplicativos, escolha o aplicativo que você deseja atualizar.
3. No painel de navegação, escolha Hospedagem e, em seguida, escolha Cabeçalhos e cache personalizados.
4. Na página Cabeçalhos e cache personalizados, localize a seção Configurações da chave de cache e escolha Editar.
5. Escolha Manter cookies na chave de cache ou Remover cookies da chave de cache. A captura de tela a seguir mostra a opção de configurações da tecla Cache no console.

Cache key settings

Keep cookies in cache key

Enabled



Changing this setting can impact your app's performance.

[Learn more](#) 

6. Escolha Salvar.

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 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 obtê-lo do servidor de origem antes de 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 contém 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](#)
- [Proteção de dados no Amplify](#)
- [Validação de conformidade para AWS Amplify](#)
- [Segurança de infraestrutura em AWS Amplify](#)
- [Registro e monitoramento de eventos de segurança no Amplify](#)
- [Prevenção do problema do substituto confuso entre serviços](#)
- [Práticas recomendadas de segurança para o Amplify](#)

Gerenciamento de identidade e acesso para o Amplify

AWS Identity and Access Management (IAM) é uma ferramenta Serviço da AWS que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar os recursos do Amplify. IAMé um Serviço da AWS que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciando acesso usando políticas](#)
- [Como o Amplify funciona com IAM](#)
- [Exemplos de políticas baseadas em identidade para o Amplify](#)
- [AWS políticas gerenciadas para AWS Amplify](#)
- [Solução de problemas de identidade e acesso do Amplify](#)

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. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta página para entender os conceitos básicos doIAM. Para saber mais sobre como sua empresa pode usar o IAM Amplify, consulte. [Como o Amplify funciona com IAM](#)

IAMadministrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso ao Amplify. Para ver exemplos de políticas baseadas em identidade

do Amplify que você pode usar, consulte. IAM [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 IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do 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 entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. 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 você mesmo assinar solicitações, consulte [Assinar AWS API solicitações](#) no Guia IAM do usuário.

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 AWS IAM Identity Center do usuário e [Uso da autenticação multifator \(MFA\) AWS no Guia do IAMusuário](#).

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 ver a lista

completa de tarefas que exigem que você faça login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do IAM usuário.

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 informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham 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 IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do IAMusuário.

Um [IAMgrupo](#) é uma identidade que especifica uma coleção de IAM usuários. 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 IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

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 IAM usuário \(em vez de uma função\)](#) no Guia do IAM usuário.

IAMfunções

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console [trocando de funções](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Usando IAM funções](#) no Guia IAM do usuário.

IAMfunções 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 informações sobre funções para federação, consulte [Criação de uma função para um provedor de identidade terceirizado](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em. IAM Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Manual do Usuário do AWS IAM Identity Center .
- **Permissões temporárias IAM de IAM usuário** — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas** — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente 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 funções e políticas baseadas em recursos para acesso entre contas, consulte [Acesso a recursos entre contas IAM no Guia](#) do IAM usuário.
- **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 na 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 IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um Serviço da AWS, combinadas com a solicitação Serviço da AWS para fazer solicitações aos serviços posteriores. FAS solicitações 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 FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- **Função de serviço** — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma Serviço da AWS](#) no Guia do IAM usuário.
- **Função vinculada ao serviço** — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um 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 IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.
- **Aplicativos em execução na Amazon EC2** — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

Para saber se usar IAM funções ou IAM usuários, consulte [Quando criar uma IAM função \(em vez de um usuário\)](#) no Guia do IAM usuário.

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 JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas 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 permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar 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 do AWS Management Console AWS CLI, do ou do AWS API.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. 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 [Criação de IAM políticas no Guia](#) do IAMusuário.

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 [Escolha entre políticas gerenciadas e políticas em linha no Guia](#) do IAMusuário.

Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e 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 de uma política baseada IAM em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [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 recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAM usuário ou função). 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 IAM entidades](#) no Guia IAM do usuário.
- Políticas de controle de serviço (SCPs) — SCPs são JSON políticas 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 várias Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo

cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations eSCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.

- 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 IAM do usuário.

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 política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.

Como o Amplify funciona com IAM

Antes de usar IAM para gerenciar o acesso ao Amplify, saiba quais IAM recursos estão disponíveis para uso com o Amplify.

IAMrecursos que você pode usar com o Amplify

IAMrecurso	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

IAMrecurso	Suporte do Amplify
ABAC(tags nas 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 IAM maioria dos recursos, [AWS consulte os serviços que funcionam IAM](#) com no Guia IAMdo usuário.

Políticas baseadas em identidade do Amplify

Compatível com políticas baseadas em identidade: Sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. 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 [Criação de IAM políticas no Guia](#) do IAMusuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem 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 você pode usar em uma JSON política, consulte a [referência IAM JSON de elementos de política](#) no Guia IAM do usuário.

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

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e 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 habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. 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 IAM administrador 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 IAM no](#) Guia do IAM usuário.

Ações políticas para Amplify

Compatível com ações de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O `Action` elemento de uma JSON política 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 AWS API operação associada. Há algumas exceções, como ações somente de permissão que não têm uma operação correspondente. API 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

Compatível com recursos de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento Resource JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento Resource ou NotResource. Como prática recomendada, especifique um recurso usando seu [Amazon Resource Name \(ARN\)](#). 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 por AWS Amplify na Referência](#) de Autorização de Serviço. Para saber com quais ações você pode especificar cada recurso, consulte [Ações definidas por AWS Amplify](#). ARN

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

Compatível com chaves de condição de política específicas de serviço: Sim

Os administradores podem usar AWS JSON políticas 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, você pode conceder permissão a um IAM usuário para acessar um recurso somente se ele estiver marcado com o nome de IAM usuário. Para obter mais informações, consulte [elementos de IAM política: variáveis e tags](#) no Guia IAM do usuário.

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 IAM do usuário.

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

SuportesACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Controle de acesso baseado em atributos (ABAC) com Amplify

Suportes ABAC (tags nas políticas): Parciais

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

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 ABAC, consulte [O que é ABAC?](#) no Guia do IAM usuário. Para ver um tutorial com etapas de configuração ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\) no Guia](#) do IAM usuário.

Usar credenciais temporárias com o Amplify

Compatível 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 nesse trabalho IAM](#) no Guia do IAM usuário.

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 a troca de funções, consulte [Alternando para uma função \(console\)](#) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. 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 obter mais informações, consulte [Credenciais de segurança temporárias em IAM](#).

Sessões de acesso direto para o Amplify

Suporta sessões de acesso direto (FAS): Sim

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um Serviço da AWS, combinadas com a solicitação Serviço da AWS para fazer solicitações aos serviços posteriores. FAS as solicitações 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 FAS solicitações, consulte [Encaminhar sessões de acesso](#).

Perfis de serviço do Amplify

Compatível com perfis de serviço: Sim

Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma Serviço da AWS](#) no Guia do IAM usuário.

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

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um 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 IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Para obter detalhes sobre a criação ou o gerenciamento de funções vinculadas a serviços, consulte [AWS serviços que funcionam com IAM](#) no Guia do IAM usuário. 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 o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte [Criação de IAM políticas no Guia](#) do IAM usuário.

Para obter detalhes sobre ações e tipos de recursos definidos pelo Amplify, incluindo o formato de cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição AWS Amplify na Referência de Autorização de Serviço](#). ARNs

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 sua 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 AWS gerenciadas](#) ou [políticas AWS gerenciadas para funções de trabalho](#) no Guia IAM do usuário.

- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar 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 IAM como usar para aplicar permissões, consulte [Políticas e permissões IAM no](#) Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica Serviço da AWS, como AWS CloudFormation. Para obter mais informações, consulte [Elementos IAM JSON da política: Condição](#) no Guia IAM do usuário.
- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAMO Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação da política do IAM Access Analyzer](#) no Guia do IAM Usuário.
- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte [Configurando o API acesso MFA protegido](#) no Guia do IAM usuário.

Para obter mais informações sobre as melhores práticas em IAM, consulte [as melhores práticas de segurança IAM no](#) Guia IAM do usuário.

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 AWS CLI ou AWS API. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Com o lançamento do Amplify Studio, a exclusão de um aplicativo ou back-end requer ambas as permissões `amplify` e `amplifybackend`. Se uma IAM política fornecer somente `amplify` permissões, 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 [Adicionar permissões a um usuário](#) no Guia do IAM usuário.

Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou AWS API

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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. 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 elas 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 um novo Serviço da AWS é lançado ou novas API operações são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte [políticas AWS gerenciadas](#) no Guia IAM do usuário.

AWS política gerenciada: AdministratorAccess -Amplify

Você pode anexar a AdministratorAccess-Amplify política às suas IAM identidades. 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 IAM anexa a política

`AdministratorAccess-Amplify` gerenciada à função `Amplify-Backend Deployment` de serviço.

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 IAM ações. 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`

Você pode anexar a `AmplifyBackendDeployFullAccess` política às suas IAM identidades.

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 de parâmetros `String` e `SecureString` os SSM parâmetros gerenciados pelo Amplify.
- `AWS AppSync`— Atualize e recupere recursos de AWS AppSync esquema, resolvedor e função. O objetivo é oferecer suporte à funcionalidade de hotswapping do sandbox 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.

Recupere as tags de uma função Lambda. O objetivo é oferecer suporte às funções do Lambda definidas pelos clientes.

- Amazon S3— Recupere os ativos de implantação do Amplify.
- AWS Security Token Service— Permite que AWS Cloud Development Kit (AWS CDK) CLI o 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.
- CloudWatch Logs— Recupere os registros da função Lambda de um cliente. O objetivo é permitir que um ambiente sandbox de desenvolvimento em nuvem da Amplify transmita os registros de uma função Lambda para o terminal do cliente.

Para ver as permissões dessa política, consulte [AmplifyBackendDeployFullAccess](#) na 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 receber alertas automáticos sobre alterações nessa página, assine o RSS feed na [Histórico do documento para AWS Amplify](#) página.

Alteração	Descrição	Data
AmplifyBackendDeployFullAccess : atualizar para uma política existente	Adicione acesso de leitura ao <code>lambda:ListTags</code> e aos <code>logs:FilterLogEvents</code> recursos para dar suporte às funções do Lambda definidas pelos clientes. Essas permissões permitem que um ambiente sandbox de desenvolvimento em nuvem do Amplify transmita os registros de uma função	18 de julho de 2024

Alteração	Descrição	Data
	Lambda para o terminal do cliente.	
AmplifyBackendDeployFullAccess : atualizar para uma política existente	Adicione acesso de leitura ao <code>arn:aws:ssm:*:*:parameter/cdk-bootstrap/*</code> recurso para permitir que o Amplify detecte a versão CDK bootstrap na conta de um cliente.	31 de maio de 2024

Alteração	Descrição	Data
<p>AmplifyBackendDeployFullAccess: atualizar para uma política existente</p>	<p>Adicione uma nova declaração o <code>AmplifyDiscoverRDSVpcConfig</code> de política com as permissões de EC2 somente leitura da Amazon RDS e da Amazon, definidas de acordo com as condições do recurso e da conta. Essas permissões suportam o <code>npx amplify generate schema-from-database</code> comando Amplify Gen 2, que permite aos clientes gerar um esquema de dados Typescript a partir de um banco de dados existente. SQL</p> <p>Adicione as <code>ec2:DescribeSubnets</code> permissões <code>rds:DescribeDBProxies</code> <code>rds:DescribeDBInstances</code> <code>rds:DescribeDBClusters</code> <code>rds:DescribeDBSubnetGroups</code> , e. O <code>npx amplify generate schema-from-database</code> comando exige essas permissões para verificar se um host de banco de dados especificado está hospedado na Amazon RDS e gerar automaticamente a VPC configuração da Amazon</p>	<p>17 de abril de 2024</p>

Alteração	Descrição	Data
	necessária para provisionar os outros recursos necessários para configurar e AWS AppSync API apoiar um SQL banco de dados.	
AmplifyBackendDeployFullAccess : atualizar para uma política existente	<p>Adicione a ação <code>cloudformation:DeleteStack</code> de política para oferecer suporte à exclusão da pilha quando a <code>DeleteBranch</code> API for chamada.</p> <p>Adicione a ação <code>lambda:GetFunction</code> de política para oferecer suporte às funções de hotswapping.</p> <p>Adicione a ação <code>lambda:UpdateFunctionConfiguration</code> política para oferecer suporte às atualizações da função Lambda.</p>	5 de abril de 2024
AdministratorAccess-Amplify — Atualização de uma política existente	Adicione as <code>cloudformation:UntagResource</code> e <code>cloudformation:TagResource</code> e as chamadas para o suporte AWS CloudFormation APIs.	4 de abril de 2024

Alteração	Descrição	Data
<p>AmplifyBackendDeployFullAccess: atualizar para uma política existente</p>	<p>Adicione a ação <code>lambda:InvokeFunction</code> política para oferecer suporte ao AWS Cloud Development Kit (AWS CDK) hotswapping. Ele AWS CDK faz chamadas diretas para uma função Lambda para realizar a troca de ativos do Amazon S3.</p> <p>Adicione a ação <code>lambda:UpdateFunctionCode</code> de política para oferecer suporte às funções de hotswapping.</p>	<p>02 de janeiro de 2024</p>
<p>AmplifyBackendDeployFullAccess: atualizar para uma política existente</p>	<p>Adicione ações de política para viabilizar a operação <code>UpdateApiKey</code>. Isso é necessário para permitir uma implantação bem-sucedida da aplicação após sair e reiniciar o sandbox sem excluir recursos.</p>	<p>17 de novembro de 2023</p>
<p>AmplifyBackendDeployFullAccess: atualizar para uma política existente</p>	<p>Adicione a permissão <code>amplify:GetBackendEnvironment</code> para apoiar a implantação do aplicativo Amplify.</p>	<p>6 de novembro de 2023</p>
<p>AmplifyBackendDeployFullAccess— Nova política</p>	<p>O Amplify adicionou uma nova política com as permissões mínimas necessárias para implantar os recursos de back-end do Amplify.</p>	<p>8 de outubro de 2023</p>

Alteração	Descrição	Data
AdministratorAccess-Amplifi car — Atualização de uma política existente	Adicione a <code>ecr:DescribeRepositories</code> permissão exigida pela interface de linha de comando do Amplify ()CLI.	1.º de junho de 2023

Alteração	Descrição	Data
<p>AdministratorAccess-Amplify — Atualização de uma política existente</p>	<p>Adicione uma ação de política para apoiar a remoção de tags de um recurso AWS AppSync .</p> <p>Adicione uma ação política para apoiar o recurso Amazon Polly.</p> <p>Adicione uma ação de política para oferecer suporte à atualização da configuração do OpenSearch domínio.</p> <p>Adicione uma ação de política para apoiar a remoção de tags de um perfil AWS Identity and Access Management .</p> <p>Adicione uma ação de política para apoiar a remoção de tags de um recurso Amazon DynamoDB.</p> <p>Adicione as permissões <code>cloudfront:GetCloudFrontOriginAccessIdentity</code> e <code>cloudfront:GetCloudFrontOriginAccessIdentityConfig</code> ao bloco de instruções para apoiar os fluxos de trabalho <code>CLISDKCalls</code> de publicação e hospedagem do Amplify.</p>	<p>24 de fevereiro de 2023</p>

Alteração	Descrição	Data
	<p>Adicione a permissão <code>s3:PutBucketPublicAccessBlock</code> ao bloco de instruções para permitir que o <code>CLIManageviaCFNPolicy</code> 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.</p> <p>Adicione a <code>cloudformation:DescribeStacks</code> permissão ao bloco de instruções <code>CLISDKCalls</code> 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.</p> <p>Adicione a permissão <code>cloudformation:ListStacks</code> ao bloco de instruções <code>CLICloudformationPolicy</code>. Essa permissão é necessária para apoiar totalmente a <code>CloudFormation DescribeStacks</code> ação.</p>	

Alteração	Descrição	Data
AdministratorAccess-Amplify car — Atualização de uma política existente	<p>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</p>	<p>30 de agosto de 2022</p>
AdministratorAccess-Amplify car — Atualização de uma política existente	<p>Adicione ações de política para bloquear o acesso público ao bucket do Amazon S3 de implantação do Amplify.</p>	<p>27 de abril de 2022</p>
AdministratorAccess-Amplify car — Atualização de uma política existente	<p>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.</p> <p>Adicione uma ação para permitir que os clientes especifiquem uma função Lambda diferente para lidar com eventos de uma fonte de eventos existente usando o Amplify. CLI Com essas mudanças, AWS Lambda será capaz de realizar a UpdateEventSourceMapping ação.</p>	<p>17 de abril de 2022</p>

Alteração	Descrição	Data
AdministratorAccess-Amplify car — 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-Amplify car — 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-Amplify — Atualização de uma política existente	<p>Adicione ações do Amazon Lex para apoiar a categoria Amplify Interactions.</p> <p>Adicione ações do Amazon Rekognition para apoiar a categoria Amplify Predictions.</p> <p>Adicione uma ação do Amazon Cognito para oferecer suporte à MFA configuração nos grupos de usuários do Amazon Cognito.</p> <p>Adicione CloudFormation ações ao suporte AWS CloudFormation StackSets.</p> <p>Adicione ações do Amazon Location Service para apoiar a categoria Amplify Geo.</p> <p>Adicione uma ação do Lambda para oferecer suporte às camadas do Lambda no Amplify.</p> <p>Adicione ações de CloudWatch registros para apoiar CloudWatch eventos.</p> <p>Adicione ações do Amazon S3 para oferecer suporte à categoria Amplify Storage.</p> <p>Adicione ações de política para oferecer suporte a</p>	27 de setembro de 2021

Alteração	Descrição	Data
	aplicativos renderizados () do lado do servidor. SSR	

Alteração	Descrição	Data
AdministratorAccess-Amplify — Atualização de uma política existente	<p>Consolide todas as ações do Amplify em uma única ação <code>amplify:*</code>.</p> <p>Adicione uma ação do Amazon S3 para oferecer suporte à criptografia de buckets Amazon S3 do cliente.</p> <p>Adicione ações de limite de IAM permissão para oferecer suporte aos aplicativos do Amplify que têm limites de permissão ativados.</p> <p>Adicione SNS ações da Amazon para oferecer suporte à visualização de números de telefone de origem e à visualização, criação, verificação e exclusão de números de telefone de destino.</p> <p>Amplify Studio: adicione ações políticas AWS Lambda, IAM, e do Amazon Cognito para permitir o gerenciamento de back-ends no console do Amplify AWS CloudFormation e no Amplify Studio.</p> <p>Adicione uma declaração de política AWS Systems Manager (SSM) para</p>	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 informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com o Amplify e 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, é preciso atualizar suas políticas para permitir que você realize a ação.

O exemplo de erro a seguir ocorre quando o mateojackson IAM usuário tenta usar o console para ver detalhes sobre um *my-example-widget* recurso fictício, mas não tem as permissões fictícias `amplify:GetWidget`.

```
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 IAM política que fornece somente `amplify` permissões, você receberá um erro de permissões ao tentar excluir um aplicativo.

O exemplo de erro a seguir ocorre quando o `mateojackson` IAM usuário tenta usar o console para excluir um `example-amplify-app` recurso fictício, mas não tem as `amplifybackend:RemoveAllBackends` permissões.

```
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 exemplo de erro a seguir ocorre quando um IAM usuário chamado `marymajor` tenta usar o console para realizar 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 suporte a 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 IAM](#).
- Para saber como fornecer acesso aos seus recursos em todas as Contas da AWS que você possui, consulte [Fornecer acesso a um IAM usuário em outra Conta da AWS de sua propriedade](#) no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Fornecer Contas da AWS acesso a terceiros](#) no Guia do IAM usuário.
- Para saber como fornecer acesso por meio da federação de identidades, consulte [Fornecendo acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do IAM usuário.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.

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 APN parceiros, atuando como controladores ou processadores de dados, são responsáveis por quaisquer dados pessoais que coloquem 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 a autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos.
- Configure API e registre as 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,, API AWS CLI, ou. AWS 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 sua solicitação para esse servidor.

Para obter mais informações sobre proteção de dados, consulte o [Modelo de Responsabilidade AWS Compartilhada e GDPR](#) a postagem no blog AWS de segurança.

Criptografia em repouso

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 usos SSDs que são criptografados para pontos de presença de edge location (POPs) e EBS volumes criptografados para caches regionais de borda (RECs). O código e a configuração da função no CloudFront Functions são sempre armazenados em um formato criptografado SSDs no local POPs criptografado na 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. Toda comunicação entre clientes e o Amplify e entre o Amplify e suas dependências posteriores é protegida por meio de TLS conexões assinadas usando o processo de assinatura Signature Version 4. Todos os endpoints do Amplify Hosting usam SHA-256 certificados que são gerenciados pela AWS Certificate Manager Autoridade de Certificação Privada. Para obter mais informações, consulte [Processo de assinatura da versão 4](#) e [O que é 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

Audidores terceirizados avaliam a segurança e a conformidade AWS Amplify como parte de vários programas de AWS conformidade. Isso inclui SOC, PCI, ISO, HIPAA, MTCS, C5, K-ISMS, ENS High, OSPAR, HITRUSTCSF, e FINMA.


Para saber se um 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 HIPAA segurança e conformidade na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

 Note

Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a [Referência de serviços HIPAA elegíveis](#).

- 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 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 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, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso 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 API chamadas AWS publicadas para acessar o Amplify pela rede. Os clientes devem oferecer suporte para:

- Segurança da camada de transporte (TLS). Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Suítes de criptografia com sigilo direto perfeito (), como (Ephemeral PFS Diffie-Hellman) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. 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 CPU uso ou outras métricas de suas instâncias do Amazon Elastic Compute Cloud (AmazonEC2) 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 de EC2 instâncias da Amazon 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 CloudTrail captura API chamadas e eventos relacionados feitos por ou em nome de sua AWS conta e entrega os arquivos de log para um bucket do Amazon Simple Storage Service (Amazon

S3) especificado por você. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte [Log de chamadas de API do Amplify usando o AWS CloudTrail](#).

- EventBridge da 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 SaaS AWS e serviços, e encaminha esses dados para destinos como AWS Lambda. Isso permite monitorar eventos que ocorrem em serviços e criar 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 concede 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 a ramificação ARN do aplicativo Amplify. Especifique esse valor no formato `arn:Partition:amplify:Region:Account:apps/AppId/branches/BranchName`.

A maneira mais eficaz de se proteger contra o confuso problema do deputado é usar a chave de contexto ARN de condição `aws:SourceArn` global com todo o recurso. Se você não souber a totalidade ARN do recurso ou se estiver especificando vários recursos, use a chave de condição de contexto `aws:SourceArn` global 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 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	Ajusté	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 o 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 o é implantado pelo AWS Amplify Console após uma construção.

Nome	Padrão	Ajuste	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	Ajuste	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 processamento 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](#)
- [Solução de problemas de imagem de compilação do Amazon Linux 2023](#)
- [Solucionar problemas de domínios personalizados](#)
- [Solução de problemas de aplicativos renderizados do lado do servidor](#)

Solução de problemas gerais do Amplify

As informações a seguir podem ajudá-lo a solucionar problemas gerais com o Amplify Hosting.

Tópicos

- [HTTPCódigo de status 429 \(muitas solicitações\)](#)

HTTPCódigo de status 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 atribuídos 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 a do RPS aplicativo. Por exemplo, com os mesmos 20.000 tokens, uma SSR página altamente otimizada que responde em 100 milissegundos pode suportar mais do que uma página com latência superior RPS 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. AL20 023 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 AL2 023 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 na [Amplify](#) Hosting. GitHub FAQ

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 AL2 023 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 os tópicos desta seção para obter ajuda.

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

- [Preciso verificar se meu CNAME problema está 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 CNAMEAlreadyExistsException erro](#)
- [Eu recebo um erro de verificação adicional necessária](#)
- [Eu recebo um erro 404 no CloudFront URL](#)
- [Recebo SSL certificados ou HTTPS erros ao visitar meu domínio](#)




Preciso verificar se meu CNAME problema está resolvido

1. Depois de atualizar seus DNS registros com seu provedor de domínio terceirizado, você pode usar uma ferramenta como [dig](#) ou um site gratuito, como <https://www.whatsmydns.net/>, para verificar se o CNAME registro está sendo resolvido corretamente. A captura de tela a seguir demonstra como usar whatsmydns.net para verificar seu registro do domínio www.example.com.

CNAME



2. Escolha Pesquisar e whatsmydns.net exibirá os resultados para o seu CNAME. A captura de tela a seguir é um exemplo de uma lista de resultados que verifica se o CNAME problema foi resolvido corretamente em cloudfront.net. URL


 Dallas TX, United States Speakeasy	d1e0xkpcedddpz.cloudfront.net ✓
 Reston VA, United States Sprint	d1e0xkpcedddpz.cloudfront.net ✓
 Atlanta GA, United States Speakeasy	d1e0xkpcedddpz.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 CNAME entrada existe nas suas DNS configurações com seu provedor de domínio.

 Important

É importante atualizar seus registros CNAME assim que você criar seu domínio personalizado. Depois que o aplicativo é criado no console do Amplify, o registro CNAME é verificado em intervalos de alguns minutos para determinar se ocorre solução. Se não ocorrer depois de uma hora, a verificação será feita em intervalos de algumas horas, o que poderá causar um atraso na disponibilidade do domínio. Se você adicionou ou atualizou seus registros CNAME algumas horas depois de criar seu aplicativo, essa é a causa mais provável de seu aplicativo ficar preso no estado de verificação pendente.

3. Se você verificou que o CNAME registro existe, pode haver um problema com seu DNS provedor. Você pode entrar em contato com o DNS provedor para diagnosticar por que a DNS verificação não CNAME está sendo resolvida ou migrá-la DNS para o Route 53. Para obter mais informações, consulte [Tornando o Amazon Route 53 o DNS serviço 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.

Search all fields X All Types

Displaying 1 to 2 out of 2 Hosted Zones

Domain Name	Type	Record Set Count	Comment
local.	Private	2	Created by Route 53 Auto Nam.

Hosted Zone Details

Domain Name: [Redacted]

Type: Public Hosted Zone

Hosted Zone ID: Z1NMQLEEGTLCM3

Record Set Count: 2

Comment: [Edit]

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

- 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.

Registered domains > designaws.com

Edit contacts Manage DNS Delete domain

Name servers ⓘ ns-294.awsdns-36.com
ns-1886.awsdns-43.co.uk
ns-953.awsdns-55.net
ns-1192.awsdns-21.org
[Add or edit name servers](#)

DNSSEC status ⓘ Not available ⓘ

Modify this to match NameServers in your hosted zone.

- 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 CNAMEAlreadyExistsException erro

Se você receber um CNAMEAlreadyExistsException erro, 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.

- 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 de 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 AWS Certificate Manager (ACM) requer 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 tiver um plano de suporte, publique um novo tópico no [Fórum de ACM Discussão](#).

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 no CloudFront URL

Para atender ao tráfego, o Amplify Hosting aponta para um registro CloudFront URL via a. CNAME No processo de conectar um aplicativo a um domínio personalizado, o console do Amplify exibe o

CloudFront URL para o aplicativo. No entanto, você não pode acessar seu aplicativo diretamente usando isso CloudFrontURL. Retorna um erro 404. Seu aplicativo resolve apenas usando o URL 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 SSL certificados ou HTTPS erros ao visitar meu domínio

Se você tiver DNS registros de Autorização de Autoridade Certificadora (CAA) configurados com seu DNS provedor terceirizado, AWS Certificate Manager (ACM) talvez não consiga atualizar ou reemitir certificados intermediários para seu SSL certificado de domínio personalizado. Para resolver isso, você precisa adicionar um CAA registro 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 CAA registro para confiar em uma autoridade de certificação da Amazon

1. Configure um CAA registro 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 CAA registro, consulte [Problemas de autorização da autoridade de certificação \(CAA\)](#) no Guia do AWS Certificate Manager usuário.
2. Use um dos métodos a seguir para atualizar seu SSL certificado:
 - Atualize manualmente usando o console do Amplify.

Note

Esse método causará tempo de inatividade para seu domínio personalizado.

- a. Faça login AWS Management Console e abra o console do [Amplify](#).
- b. Escolha o aplicativo ao qual você deseja adicionar um CAA registro.
- 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 SSL certificado 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 DNS provedor terceirizado.](#)
 - [Atualizar DNS registros para um domínio gerenciado pelo GoDaddy.](#)
 - [Atualizar DNS registros de um domínio gerenciado pelo Google Domains.](#)
- Entre em contato AWS Support para que seu SSL certificado seja reemitido.

Solução de problemas de aplicativos renderizados do lado do servidor

Se você tiver problemas inesperados ao implantar um SSR aplicativo 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 SSR web](#) no repositório Amplify GitHub Hosting Issues.

Tópicos

- [Você está usando um adaptador de framework](#)
- [APIs rotas 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 HTTP resposta é muito grande](#)

Você está usando um adaptador de framework

Se você estiver tendo problemas para implantar um SSR aplicativo que usa um adaptador de estrutura, consulte [Usando adaptadores de código aberto](#).

APIs rotas Edge fazem com que sua compilação do Next.js falhe

Atualmente, o Amplify não oferece suporte ao Next.js Edge API Routes. Você deve usar o non-edge APIs e o middleware 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 ao On-Demand. ISR 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 SSR aplicativos é de 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 o 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 os artefatos pré-assinados URL 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 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 o [AWS CodeBuild](#) site do Next.js.

O tamanho da HTTP resposta é muito grande

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

AWS Amplify Referência de host

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

Tópicos

- [Suporte a AWS CloudFormation](#)
- [Suporte a AWS Command Line Interface](#)
- [Suporte para marcação de recursos](#)
- [Amplify Hosting API](#)

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: 13 de agosto de 2024

Alteração	Descrição	Data
Novo capítulo sobre gerenciamento de configuração de cache	Foi adicionado o Gerenciamento da configuração de cache de um aplicativo capítulo para descrever o comportamento padrão de armazenamento em cache do Amplify e como ele aplica políticas de cache gerenciado ao conteúdo.	13 de agosto de 2024
Tópico de políticas gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	18 de julho 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.	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	17 de abril de 2024

Alteração	Descrição	Data
	políticas AWS gerenciadas do Amplify.	
Capítulo de introdução atualizado	O Começando a implantar um aplicativo no 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
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 sobre 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

Alteração	Descrição	Data
Novo suporte para SSL TLS certificados personalizados	O Usando TLS certifica dosSSL/ tópico foi adicionado ao Configurar domínios personalizados capítulo para descrever o suporte do Amplify para TLS certificados SSL /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
Novo suporte para SSR estruturas	O Implantação de aplicativos renderizados do lado do servidor com o Amplify Hosting tópico foi atualizado para descrever o suporte do Amplify para qualquer estrutura baseada em JavaScript com um adaptador de código aberto SSR.	19 de novembro de 2023
Nova compatibilidade com o lançamento do recurso de otimização de imagem	Adição do tópico Otimização de imagem para SSR aplicativos para descrever a compatibilidade integrada com otimização de imagem para aplicações renderizadas no lado do servidor.	19 de novembro de 2023

Alteração	Descrição	Data
Tópico de políticas gerenciadas atualizado	O 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
Novo tópico sobre subdomínios 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 gerenciadas para AWS Amplify tópico foi atualizado para descrever a nova política AmplifyBackendDeployFullAccess AWS gerenciada do Amplify.	8 de outubro de 2023

Alteração	Descrição	Data
Novo suporte para lançamento de atributos de estruturas monorepo	O tópico 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 gerenciadas atualizado	O AWS políticas gerenciadas para AWS Amplify tópico foi atualizado para descrever as mudanças recentes nas políticas AWS gerenciadas do Amplify.	1.º de junho 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.	24 de fevereiro de 2023
Capítulo de renderização no lado do servidor atualizado	O capítulo Implantação de 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

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.	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
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

Alteração	Descrição	Data
Lançamento do novo atributo Amplify Studio	O tópico Bem-vindo ao AWS Amplify Hosting 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
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

Alteração	Descrição	Data
Novo tópico de políticas gerenciadas	Foi adicionado o AWS políticas gerenciadas para AWS Amplify tópico para descrever as políticas AWS gerenciadas do Amplify e as mudanças recentes nessas políticas.	28 de julho de 2021
Capítulo de renderização do lado do servidor atualizado	O capítulo Implantação de 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
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 <code>AMPLIFY_MONOREPO_APP_ROOT</code> ao implantar um aplicativo monorepo com o Amplify.	20 de julho de 2021

Alteração	Descrição	Data
Capítulo atualizado sobre implantações de ramificação de atributo	<p>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 <code>aws-exports.js</code> 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.</p>	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 lançamento SSR de recursos	Foi adicionado o Implantação de aplicativos renderizados do lado do servidor com o Amplify Hosting capítulo para descrever o suporte do Amplify para aplicativos web que usam renderização do lado do servidor (SSR) e são criados com Next.js.	18 de maio de 2021
Novo capítulo de segurança	Foi adicionado o capítulo Segurança no Amplify para descrever como aplicar o modelo de responsabilidade compartilhada ao usar o Amplify e como configurar o Amplify para atender aos seus objetivos de segurança e conformidade.	26 de março de 2021

Alteração	Descrição	Data
Tópico de compilações personalizadas atualizado	Atualizou o tópico Imagens de compilação personalizadas e atualizações de pacotes ao vivo para descrever como configurar uma imagem de compilação personalizada hospedada no Amazon Elastic Container Registry Public.	12 de março de 2021
Tópico de monitoramento atualizado	Atualizou o tópico Monitoramento para descrever como acessar dados de CloudWatch métricas da Amazon e definir alarmes.	2 de fevereiro de 2021
Novo tópico de CloudTrail registro	Foi adicionado o AWS CloudTrail tópico Logging Amplify API calls using para descrever como AWS CloudTrail captura e registra todas as API ações da Referência do AWS Amplify Console e da API Referência da UI do AWS Amplify Administrador. API	2 de fevereiro de 2021

Alteração	Descrição	Data
Lançamento do novo atributo Admin UI	O tópico Bem-vindo ao AWS Amplify Hosting 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	<p>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.</p>	20 de junho de 2020
Novo tópico de notificações	<p>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.</p>	20 de junho de 2020

Alteração	Descrição	Data
Atualizou o tópico de domínios personalizados	Atualizou o Configurar domínios personalizados tópico para melhorar os procedimentos para adicionar 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.