Benutzer-Leitfaden

AWS Amplify Hosten

AWS Amplify Hosten: Benutzer-Leitfaden

Table of Contents

Willkommen bei AWS Amplify Hosting

Amplify Hosting bietet einen Git-basierten Workflow für das Hosten serverloser Full-Stack-Webanwendungen mit kontinuierlicher Bereitstellung. Amplify stellt Ihre App im AWS globalen Content Delivery Network (CDN) bereit. Dieses Benutzerhandbuch enthält die Informationen, die Sie für den Einstieg in Amplify Hosting benötigen.

Unterstützte Frameworks

Amplify Hosting unterstützt viele gängige SSR-Frameworks, Single-Page-Application-Frameworks (SPA) und statische Site-Generatoren, einschließlich der folgenden.

SSR-Frameworks

- Next.js
- Nuxt
- Astromit einem Community-Adapter
- SvelteKitmit einem Community-Adapter
- Jedes SSR-Framework mit einem benutzerdefinierten Adapter

SPA-Frameworks

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

Generatoren für statische Websites

- Eleventy
- Gatsby
- Hugo
- Jekyll

- VuePress

Amplify Hosting-Funktionen

Feature-Zweige

Managen Sie Produktions- und Staging-Umgebungen für Ihr Frontend und Backend, indem Sie neue Filialen verbinden.

Benutzerdefinierte Domänen

Connect Sie Ihre Anwendung mit einer benutzerdefinierten Domain.

Vorschauen von Pull-Requests

Eine Vorschau der Änderungen während der Code-Reviews anzeigen.

End-to-end testen

Verbessern Sie die Qualität Ihrer App mit end-to-end Tests.

Passwortgeschützte Filialen

Schützen Sie Ihre Web-App mit einem Passwort, damit Sie an neuen Funktionen arbeiten können, ohne sie öffentlich zugänglich zu machen.

Leitet um und schreibt neu

Richten Sie Umschreibungen und Weiterleitungen ein, um das SEO-Ranking aufrechtzuerhalten und den Traffic auf der Grundlage der Anforderungen Ihrer Kunden-App weiterzuleiten.

Atomare Bereitstellungen

Atomare Bereitstellungen eliminieren Wartungsfenster, indem sie sicherstellen, dass Ihre Web-App erst aktualisiert wird, wenn die gesamte Bereitstellung abgeschlossen ist. Dadurch entfallen Szenarien, in denen Dateien nicht korrekt hochgeladen werden.

Erste Schritte mit Amplify Hosting

Informationen zu den ersten Schritten mit Amplify Hosting finden Sie im Erste Schritte mit der Bereitstellung einer App auf Amplify Hosting Tutorial. Nach Abschluss des Tutorials wissen Sie, wie Sie eine Web-App in einem Git-Repository (GitHub, BitBucket GitLab, oder AWS CodeCommit) verbinden und sie mit kontinuierlicher Bereitstellung auf Amplify Hosting bereitstellen.

Ein Backend erstellen

AWS Amplify Gen 2 bietet ein TypeScript basiertes Entwicklererlebnis, bei dem der Code an erster Stelle steht, um Backends zu definieren. Um zu erfahren, wie Sie Amplify Gen 2 verwenden, um ein Backend zu erstellen und mit Ihrer App zu verbinden, siehe Backend erstellen und verbinden in den Amplify-Dokumenten.

Um den Code-First-Ansatz Gen 2 von Amplify besser zu verstehen, schauen Sie sich den Amplify Gen 2 Workshop auf der AWS Workshop Studio-Website an. In diesem umfassenden Tutorial erstellen Sie eine serverlose Anwendung mit React und Next.js und lernen, wie Sie die Amplify Gen 2 Data- und Auth-Bibliotheken sowie die Amplify UI-Bibliothek verwenden, um der Anwendung Funktionen hinzuzufügen.

Wenn Sie nach der Dokumentation zum Erstellen von Backends für eine Gen 1-App mithilfe der CLI und Amplify Studio suchen, finden Sie in den Gen 1 Amplify-Dokumenten unter Build & Connect-Backend weitere Informationen.

Amplify Hosting-Preise

AWS Amplify berechnet Ihnen nur das, was Sie nutzen. Weitere Informationen finden Sie unter AWS Amplify - Preise.

Erste Schritte mit der Bereitstellung einer App auf Amplify Hosting

Um Ihnen zu helfen, zu verstehen, wie Amplify Hosting funktioniert, führen Sie die folgenden Tutorials durch die Erstellung und Bereitstellung von Anwendungen, die mit gängigen SSR-Frameworks erstellt wurden, die Amplify unterstützt.

Lernprogramme

- Stellen Sie eine Next.js App für Amplify Hosting bereit
- Stellen Sie eine Nuxt.js App für Amplify Hosting bereit
- Stellen Sie eine Astro.js App für Amplify Hosting bereit
- Stellen Sie eine SvelteKit App für Amplify Hosting bereit

Stellen Sie eine Next.js App für Amplify Hosting bereit

Dieses Tutorial führt Sie durch das Erstellen und Bereitstellen einer Next.js -Anwendung aus einem Git-Repository.

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Voraussetzungen erfüllen.

Melden Sie sich an für eine AWS-Konto

> Wenn Sie noch kein AWS Kunde sind, müssen Sie einen erstellen, AWS-Konto indem Sie den Online-Anweisungen folgen. Durch die Registrierung können Sie auf Amplify und andere AWS Dienste zugreifen, die Sie mit Ihrer Anwendung verwenden können.

Erstellen einer Anwendung

> Erstellen Sie anhand der create-next-appAnweisungen in der Dokumentation zu Next.js eine grundlegende Anwendung Next.js, die Sie für dieses Tutorial verwenden können.

Erstellen Sie ein Git-Repository

> Amplify unterstützt GitHub, Bitbucket GitLab, und. AWS CodeCommit Schieben Sie Ihre `create-next-app` Anwendung in Ihr Git-Repository.

Schritt 1: Ein Git-Repository Connect

In diesem Schritt verbinden Sie Ihre Anwendung Next.js in einem Git-Repository mit Amplify Hosting.

Um eine App in einem Git-Repository zu verbinden

1. Öffnen Sie die Amplify-Konsole.
2. Wenn Sie Ihre erste App in der aktuellen Region bereitstellen, beginnen Sie standardmäßig auf der AWS AmplifyServiceseite.

   Wählen Sie oben auf der Seite die Option App bereitstellen aus.
3. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.

   Für GitHub Repositorys verwendet Amplify die GitHub Apps-Funktion, um den Amplify-Zugriff zu autorisieren. Weitere Informationen zur Installation und Autorisierung der App finden Sie unter. GitHub Amplify-Zugriff auf Repositorys einrichten GitHub

   > **Note**
   >
   > Nachdem Sie die Amplify-Konsole mit Bitbucket,, oder AWS CodeCommit autorisiert haben GitLab, ruft Amplify ein Zugriffstoken vom Repository-Anbieter ab, speichert das Token jedoch nicht auf den Servern. AWS Amplify greift auf Ihr Repository nur mit Bereitstellungsschlüsseln zu, die in einem bestimmten Repository installiert sind.
4. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:

   a. Wählen Sie den Namen des Repositorys aus, zu dem Sie eine Verbindung herstellen möchten.

   b. Wählen Sie den Namen des Repository-Zweigs aus, zu dem eine Verbindung hergestellt werden soll.

   c. Wählen Sie Weiter aus.

Schritt 2: Bestätigen Sie die Build-Einstellungen

Amplify erkennt automatisch die Reihenfolge der Build-Befehle, die für den Zweig ausgeführt werden sollen, den Sie bereitstellen. In diesem Schritt überprüfen und bestätigen Sie Ihre Build-Einstellungen.

Um die Build-Einstellungen für eine App zu bestätigen

1. Suchen Sie auf der Seite mit den App-Einstellungen den Abschnitt Build-Einstellungen.

   Stellen Sie sicher, dass der Frontend-Build-Befehl und das Build-Ausgabeverzeichnis korrekt sind. Für diese Beispiel-App Next.js ist das Build-Ausgabeverzeichnis auf `.next` eingestellt.
2. Das Verfahren zum Hinzufügen einer Servicerolle hängt davon ab, ob Sie eine neue Rolle erstellen oder eine vorhandene verwenden möchten.
   - Um eine neue Rolle zu erstellen:
     - Wählen Sie Neue Servicerolle erstellen und verwenden aus.
   - Um eine bestehende Rolle zu verwenden:
     a. Wählen Sie Eine bestehende Rolle verwenden aus.
     b. Wählen Sie in der Liste der Servicerollen die zu verwendende Rolle aus.
3. Wählen Sie Weiter aus.

Schritt 3: Stellen Sie die Anwendung bereit

In diesem Schritt stellen Sie Ihre App im AWS globalen Content Delivery Network (CDN) bereit.

Um eine App zu speichern und bereitzustellen

1. Vergewissern Sie sich auf der Seite „Überprüfen“, dass Ihre Repository-Details und App-Einstellungen korrekt sind.
2. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus. Ihr Frontend-Build dauert in der Regel 1 bis 2 Minuten, kann aber je nach Größe der App variieren.
3. Wenn die Bereitstellung abgeschlossen ist, können Sie Ihre App über den Link zur `amplifyapp.com` Standarddomäne anzeigen.

> **Note**
>
> Um die Sicherheit Ihrer Amplify-Anwendungen zu erhöhen, ist die Domain amplifyapp.com in der Public Suffix List (PSL) registriert. Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, falls Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Amplify-Anwendungen einrichten müssen. Diese Vorgehensweise trägt dazu bei, Ihre Domain vor CSRF-Versuchen (Cross-Site Request Forgery Attempts,

Anforderungsfälschung zwischen Websites) zu schützen. Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

Schritt 4: (Optional) Ressourcen bereinigen

Wenn Sie die App, die Sie für das Tutorial bereitgestellt haben, nicht mehr benötigen, können Sie sie löschen. Mit diesem Schritt wird sichergestellt, dass Ihnen keine Ressourcen in Rechnung gestellt werden, die Sie nicht nutzen.

Um eine App zu löschen

1. Wählen Sie im Navigationsbereich im Menü App-Einstellungen die Option Allgemeine Einstellungen aus.
2. Wählen Sie auf der Seite Allgemeine Einstellungen die Option App löschen aus.
3. Geben Sie im Bestätigungsfenster ein**`delete`**. Wählen Sie dann App löschen.

Füge Funktionen zu deiner App hinzu

Nachdem Sie eine App für Amplify bereitgestellt haben, können Sie einige der folgenden Funktionen erkunden, die für Ihre gehostete Anwendung verfügbar sind.

Umgebungsvariablen

Anwendungen benötigen zur Laufzeit häufig Konfigurationsinformationen. Bei diesen Konfigurationen kann es sich um Datenbankverbindungsdetails, API-Schlüssel oder Parameter handeln. Umgebungsvariablen bieten eine Möglichkeit, diese Konfigurationen während der Erstellung verfügbar zu machen. Weitere Informationen finden Sie unter Umgebungsvariablen.

Benutzerdefinierte Domänen

In diesem Tutorial hostet Amplify Ihre App für Sie auf der `amplifyapp.com` Standarddomain mit einer URL wie`https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`. Wenn Sie Ihre App mit einer benutzerdefinierten Domain verbinden, sehen Benutzer, dass Ihre App auf einer benutzerdefinierten URL gehostet wird, z. `https://www.example.com` Weitere Informationen finden Sie unter Benutzerdefinierte Domains einrichten.

Vorschauen von Pull-Requests

Web-Pull-Request-Vorschauen bieten Teams die Möglichkeit, eine Vorschau der Änderungen an Pull-Requests (PRs) anzuzeigen, bevor sie den Code mit einem Produktions- oder Integrationszweig zusammenführen. Weitere Informationen finden Sie unter Webvorschauen für Pull-Requests.

Verwalten mehrerer Umgebungen

Um zu erfahren, wie Amplify mit Feature Branches und GitFlow Workflows arbeitet, um mehrere Bereitstellungen zu unterstützen, siehe Feature Branch-Bereitstellungen und Team-Workflows.

Stellen Sie eine Nuxt.js App für Amplify Hosting bereit

Verwenden Sie die folgenden Anweisungen, um eine Nuxt.js -Anwendung für Amplify Hosting bereitzustellen. Nuxt hat mithilfe des Nitro-Servers einen voreingestellten Adapter implementiert. Auf diese Weise können Sie ein Nuxt-Projekt ohne zusätzliche Konfiguration bereitstellen.

Um eine Nuxt-App für Amplify Hosting bereitzustellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite „Alle Apps“ die Option Neue App erstellen aus.
3. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.
4. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:
   a. Wählen Sie den Namen des Repositorys aus, zu dem Sie eine Verbindung herstellen möchten.
   b. Wählen Sie den Namen des Repository-Zweigs aus, zu dem eine Verbindung hergestellt werden soll.
   c. Wählen Sie Weiter aus.
5. Wenn Sie möchten, dass Amplify App-Logs an Amazon CloudWatch Logs liefern kann, müssen Sie dies explizit in der Konsole aktivieren. Öffnen Sie den Abschnitt Erweiterte Einstellungen und wählen Sie dann im Abschnitt Serverseitiges Rendering (SSR) -Bereitstellung die Option SSR-App-Logs aktivieren.
6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite Überprüfen die Option Speichern und bereitstellen aus.

Stellen Sie eine Astro.js App für Amplify Hosting bereit

Verwenden Sie die folgenden Anweisungen, um eine Astro.js -Anwendung für Amplify Hosting bereitzustellen. Sie können eine vorhandene Anwendung verwenden oder eine Starter-Anwendung anhand eines der offiziellen Beispiele von Astro erstellen. Informationen zum Erstellen einer Starter-Anwendung finden Sie in der Astro-Dokumentation unter Verwenden eines Themes oder einer Starter-Vorlage.

Um eine Astro-Site mit SSR für Amplify Hosting bereitzustellen, müssen Sie Ihrer Anwendung einen Adapter hinzufügen. Wir unterhalten keinen Amplify-eigenen Adapter für das Astro-Framework. In diesem Tutorial wird der `astro-aws-amplify` Adapter verwendet, der von einem Mitglied der Community erstellt wurde. Dieser Adapter ist auf Github verfügbar. com/alexnguyennz/astro-aws-amplify auf der Website. GitHub AWS verwaltet diesen Adapter nicht.

Um eine Astro-App für Amplify Hosting bereitzustellen

1. Navigieren Sie auf Ihrem lokalen Computer zu der Astro-Anwendung, die Sie bereitstellen möchten.
2. Um den Adapter zu installieren, öffnen Sie ein Terminalfenster und führen Sie den folgenden Befehl aus. In diesem Beispiel wird der Community-Adapter verwendet, der auf Github verfügbar ist. com/alexnguyennz/astro-aws-amplify. Sie können es durch den Namen des Adapters *`astro-aws-amplify`* ersetzen, den Sie verwenden.

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

3. Öffnen Sie die `astro.config.mjs` Datei im Projektordner für Ihre Astro-App. Aktualisieren Sie die Datei, um den Adapter hinzuzufügen. Die Datei sollte wie folgt aussehen.

   ```
   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. Übernehmen Sie die Änderung und übertragen Sie das Projekt in Ihr Git-Repository.

   Jetzt sind Sie bereit, Ihre Astro-App auf Amplify bereitzustellen.

5. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
6. Wählen Sie auf der Seite „Alle Apps“ die Option Neue App erstellen aus.
7. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.
8. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:

   a. Wählen Sie den Namen des Repositorys aus, zu dem Sie eine Verbindung herstellen möchten.

   b. Wählen Sie den Namen des Repository-Zweigs aus, zu dem eine Verbindung hergestellt werden soll.

   c. Wählen Sie Weiter aus.

9. Suchen Sie auf der Seite mit den App-Einstellungen den Abschnitt Build-Einstellungen. Geben Sie für Build-Ausgabeverzeichnis Folgendes ein **`.amplify-hosting`**.
10. Sie müssen auch die Frontend-Build-Befehle der App in der Build-Spezifikation aktualisieren. Um die Build-Spezifikation zu öffnen, wählen Sie YML-Datei bearbeiten.
11. Suchen Sie in der `amplify.yml` Datei den Abschnitt mit den Frontend-Build-Befehlen. Geben Sie **`mv node_modules ./.amplify-hosting/compute/default`** ein.

    Ihre Build-Einstellungsdatei sollte wie folgt aussehen.

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

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

12. Wählen Sie Speichern.
13. Wenn Sie möchten, dass Amplify App-Logs an Amazon CloudWatch Logs liefern kann, müssen Sie dies explizit in der Konsole aktivieren. Öffnen Sie den Abschnitt Erweiterte Einstellungen und wählen Sie dann im Abschnitt Serverseitiges Rendering (SSR) -Bereitstellung die Option SSR-App-Logs aktivieren.
14. Wählen Sie Weiter aus.
15. Wählen Sie auf der Seite Überprüfen die Option Speichern und bereitstellen aus.

Stellen Sie eine SvelteKit App für Amplify Hosting bereit

Verwenden Sie die folgenden Anweisungen, um eine SvelteKit Anwendung für Amplify Hosting bereitzustellen. Sie können Ihre eigene Anwendung verwenden oder eine Starter-App erstellen. Weitere Informationen finden Sie in der SvelteKit Dokumentation unter Projekt erstellen.

Um eine SvelteKit App mit SSR für Amplify Hosting bereitzustellen, müssen Sie Ihrem Projekt einen Adapter hinzufügen. Wir unterhalten keinen Amplify-eigenen Adapter für das SvelteKit Framework. In diesem Beispiel verwenden wir den, der von einem Mitglied der Community `amplify-adapter` erstellt wurde. Der Adapter ist auf Github erhältlich. com/gzimbron/amplify-Adapter auf der GitHub Website. AWS verwaltet diesen Adapter nicht.

Um eine SvelteKit App für Amplify Hosting bereitzustellen

1. Navigieren Sie auf Ihrem lokalen Computer zu der SvelteKit Anwendung, die Sie bereitstellen möchten.
2. Um den Adapter zu installieren, öffnen Sie ein Terminalfenster und führen Sie den folgenden Befehl aus. In diesem Beispiel wird der Community-Adapter verwendet, der auf Github verfügbar ist. com/gzimbron/amplify-Adapter. Wenn Sie einen anderen Community-Adapter verwenden, *`amplify-adapter`* ersetzen Sie ihn durch den Namen Ihres Adapters.

```
npm install amplify-adapter
```

3. Öffnen Sie die `svelte.config.js` Datei im Projektordner für Ihre SvelteKit App. Bearbeiten Sie die Datei so, dass sie den Namen Ihres Adapters verwendet `amplify-adapter` oder *`'amplify-adapter'`* durch den Namen Ihres Adapters ersetzt. Die Datei sollte wie folgt aussehen.

```
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. Übernehmen Sie die Änderung und übertragen Sie die Anwendung in Ihr Git-Repository.
5. Jetzt sind Sie bereit, Ihre SvelteKit App auf Amplify bereitzustellen.

   Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
6. Wählen Sie auf der Seite „Alle Apps“ die Option Neue App erstellen aus.
7. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.
8. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:

   a. Wählen Sie den Namen des Repositorys aus, zu dem Sie eine Verbindung herstellen möchten.

b. Wählen Sie den Namen des Repository-Zweigs aus, zu dem eine Verbindung hergestellt werden soll.

c. Wählen Sie Weiter aus.

9. Suchen Sie auf der Seite mit den App-Einstellungen den Abschnitt Build-Einstellungen. Geben Sie für Build-Ausgabeverzeichnis Folgendes ein**`build`**.

10. Sie müssen auch die Frontend-Build-Befehle der App in der Build-Spezifikation aktualisieren. Um die Build-Spezifikation zu öffnen, wählen Sie YML-Datei bearbeiten.

11. Suchen Sie in der `amplify.yml` Datei den Abschnitt mit den Frontend-Build-Befehlen. Geben Sie **`- cd build/compute/default/`** und ein. **`- npm i --production`**

    Ihre Build-Einstellungsdatei sollte wie folgt aussehen.

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

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

12. Wählen Sie Speichern.

13. Wenn Sie möchten, dass Amplify App-Logs an Amazon CloudWatch Logs liefern kann, müssen Sie dies explizit in der Konsole aktivieren. Öffnen Sie den Abschnitt Erweiterte Einstellungen und wählen Sie dann im Abschnitt Serverseitiges Rendering (SSR) -Bereitstellung die Option SSR-App-Logs aktivieren.

14. Wählen Sie Weiter aus.

15. Wählen Sie auf der Seite Überprüfen die Option Speichern und bereitstellen aus.

Bereitstellung serverseitig gerenderter Anwendungen mit Amplify Hosting

Sie können sie verwenden AWS Amplify , um Web-Apps bereitzustellen und zu hosten, die serverseitiges Rendern (SSR) verwenden. Amplify Hosting erkennt automatisch Anwendungen, die mit dem Framework Next.js erstellt wurden, und Sie müssen keine manuelle Konfiguration in der AWS-Managementkonsole vornehmen.

Amplify unterstützt auch jedes Javascript-basierte SSR-Framework mit einem Open-Source-Build-Adapter, der die Build-Ausgabe einer Anwendung in die Verzeichnisstruktur umwandelt, die Amplify Hosting erwartet. Sie können beispielsweise Apps bereitstellen, die mit Nuxt, Astro und den SvelteKit Frameworks erstellt wurden, indem Sie die verfügbaren Adapter installieren.

Fortgeschrittene Benutzer können die Bereitstellungsspezifikation verwenden, um einen Build-Adapter zu erstellen oder ein Post-Build-Skript zu konfigurieren.

Sie können die folgenden Frameworks mit minimaler Konfiguration für Amplify Hosting bereitstellen.

Next.js

- Amplify unterstützt Next.js 15-Anwendungen, ohne dass ein Adapter erforderlich ist. Um zu beginnen, sehen Sie sich Amplify Sie die Unterstützung für Next.js an.

Nuxt.js

- Amplify unterstützt Anwendungsbereitstellungen von Nuxt.js mit einem voreingestellten Adapter. Um zu beginnen, sehen Sie sich Amplify Sie die Unterstützung für Nuxt.js an.

Astro.js

- Amplify unterstützt Anwendungsbereitstellungen von Astro.js mit einem Community-Adapter. Um zu beginnen, sehen Sie sich Amplify Sie die Unterstützung für Astro.js an.

SvelteKit

- Amplify unterstützt SvelteKit Anwendungsbereitstellungen mit einem Community-Adapter. Um zu beginnen, sehen Sie sich Amplify Sie die Unterstützung für SvelteKit an.

Open-Source-Adapter

- Verwenden Sie einen Open-Source-Adapter — Anweisungen zur Verwendung von Adaptern, die nicht in der obigen Liste aufgeführt sind, finden Sie unterVerwendung von Open-Source-Adaptern für jedes SSR-Framework.

- Erstellen Sie einen Framework-Adapter — Framework-Autoren, die Funktionen integrieren möchten, die ein Framework bietet, können die Amplify Hosting-Bereitstellungsspezifikation verwenden, um Ihre Build-Ausgabe so zu konfigurieren, dass sie der von Amplify erwarteten Struktur entspricht. Weitere Informationen finden Sie unter Verwenden der Amplify Hosting-Bereitstellungsspezifikation zur Konfiguration der Build-Ausgabe.
- Konfigurieren Sie ein Post-Build-Skript — Sie können die Amplify Hosting-Bereitstellungsspezifikation verwenden, um Ihre Build-Ausgabe nach Bedarf für bestimmte Szenarien zu bearbeiten. Weitere Informationen finden Sie unter Verwenden der Amplify Hosting-Bereitstellungsspezifikation zur Konfiguration der Build-Ausgabe. Ein Beispiel finden Sie unter Bereitstellen eines Express-Servers mithilfe des Deployment-Manifests.

Topics

Amplify Sie die Unterstützung für Next.js

Amplify unterstützt die Bereitstellung und das Hosting von serverseitig gerenderten (SSR) Web-Apps, die mit Next.js erstellt wurden. Next.js ist ein React-Framework für die Entwicklung mit. SPAs JavaScript Sie können Apps bereitstellen, die mit den Versionen Next.js bis Next.js 15 erstellt wurden und Funktionen wie Bildoptimierung und Middleware bieten.

Entwickler können Next.js verwenden, um Static Site Generation (SSG) und SSR in einem einzigen Projekt zu kombinieren. SSG-Seiten werden zum Zeitpunkt der Erstellung vorgerendert, und SSR-Seiten werden zum Zeitpunkt der Anfrage vorgerendert.

Durch das Prerendern können die Leistung und die Suchmaschinenoptimierung verbessert werden. Da Next.js alle Seiten auf dem Server vorrendert, ist der HTML-Inhalt jeder Seite bereit, sobald er den Browser des Clients erreicht. Dieser Inhalt kann auch schneller geladen werden.

Schnellere Ladezeiten verbessern die Erfahrung des Endbenutzers mit einer Website und wirken sich positiv auf das SEO-Ranking der Website aus. Das Prerendering verbessert auch die Suchmaschinenoptimierung, indem es Suchmaschinen-Bots ermöglicht, den HTML-Inhalt einer Website einfach zu finden und zu crawlen.

Next.js bietet integrierte Analyseunterstützung für die Messung verschiedener Leistungskennzahlen wie Time to First Byte (TTFB) und First Contentful Paint (FCP). Weitere Informationen zu Next.js finden Sie unter Erste Schritte auf der Website Next.js.

Unterstützung für die Funktion Next.js

Amplify Hosting Compute verwaltet vollständig das serverseitige Rendering (SSR) für Apps, die mit den Versionen 12 bis 15 von Next.js erstellt wurden.

Wenn Sie vor der Veröffentlichung von Amplify Hosting Compute im November 2022 eine Next.js App für Amplify bereitgestellt haben, verwendet Ihre App den vorherigen SSR-Anbieter von Amplify, Classic (nur Next.js 11). Amplify Hosting Compute unterstützt keine Apps, die mit Next.js Version 11 oder früher erstellt wurden. Wir empfehlen dringend, dass Sie Ihre Next.js 11-Apps zum Compute Managed SSR-Anbieter von Amplify Hosting migrieren.

In der folgenden Liste werden die spezifischen Funktionen beschrieben, die der Amplify Hosting-Compute-SSR-Anbieter unterstützt.

Unterstützte Features

- Serverseitig gerenderte Seiten (SSR)
- Statische Seiten
- API-Routen
- Dynamische Routen
- Erfasse alle Routen
- SSG (Statische Generierung)
- Inkrementelle statische Regeneration (ISR)
- Internationalisiertes (i18n) Unterpfad-Routing
- Internationalisiertes (i18n) Domain-Routing
- Internationalisierte (i18n) automatische Erkennung von Gebietsschemas
- Middleware
- Umgebungsvariablen

- Bildoptimierung
- Next.js 13 App-Verzeichnis

Nicht unterstützte Funktionen

- Edge-API-Routen (Edge-Middleware wird nicht unterstützt)
- Inkrementelle statische Regeneration (ISR) auf Anfrage
- Next.js streamen
- Middleware auf statischen Assets und optimierten Bildern ausführen
- Ausführen von Code nach einer Antwort mit `unstable_after` (Experimentelle Funktion, veröffentlicht mit Next.js 15)

Next.js Bilder

Die maximale Ausgabegröße eines Bildes darf 4,3 MB nicht überschreiten. Sie können eine größere Bilddatei irgendwo speichern und sie mit der Image-Komponente Next.js skalieren und für ein Webp- oder AVIF-Format optimieren und sie dann in einer kleineren Größe bereitstellen.

Beachten Sie, dass in der Dokumentation zu Next.js empfohlen wird, das Sharp-Bildverarbeitungsmodul zu installieren, damit die Bildoptimierung in der Produktion ordnungsgemäß funktioniert. Dies ist jedoch für Amplify-Bereitstellungen nicht erforderlich. Amplify stellt Sharp automatisch für Sie bereit.

Bereitstellen einer Next.js SSR-Anwendung für Amplify

Standardmäßig stellt Amplify neue SSR-Apps mithilfe des Compute-Service von Amplify Hosting bereit, der die Versionen 12 bis 15 von Next.js unterstützt. Amplify Hosting Compute verwaltet die Ressourcen, die für die Bereitstellung einer SSR-App erforderlich sind, vollständig. SSR-Apps in Ihrem Amplify-Konto, die Sie vor dem 17. November 2022 bereitgestellt haben, verwenden den klassischen SSR-Anbieter (nur Next.js 11).

Wir empfehlen dringend, Apps, die Classic (nur Next.js 11) SSR verwenden, zum Amplify Hosting Compute SSR-Anbieter zu migrieren. Amplify führt keine automatischen Migrationen für Sie durch. Sie müssen Ihre App manuell migrieren und dann einen neuen Build initiieren, um das Update abzuschließen. Detaillierte Anweisungen finden Sie unter Migrieren einer Next.js 11 SSR-App auf Amplify Hosting Compute.

Verwenden Sie die folgenden Anweisungen, um eine neue Next.js SSR-App bereitzustellen.

So stellen Sie mithilfe des Amplify Hosting-Compute-SSR-Anbieters eine SSR-App für Amplify bereit

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite „Alle Apps“ die Option Neue App erstellen aus.
3. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.
4. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:
   a. Wählen Sie in der Liste „Kürzlich aktualisierte Repositorys“ den Namen des Repositorys aus, mit dem Sie eine Verbindung herstellen möchten.
   b. Wählen Sie in der Branch-Liste den Namen des Repository-Branches aus, zu dem Sie eine Verbindung herstellen möchten.
   c. Wählen Sie Weiter aus.
5. Die App erfordert eine IAM-Servicerolle, die Amplify übernimmt, wenn es in Ihrem Namen andere Dienste anruft. Sie können entweder Amplify Hosting Compute erlauben, automatisch eine Servicerolle für Sie zu erstellen, oder Sie können eine Rolle angeben, die Sie erstellt haben.
   - Damit Amplify automatisch eine Rolle erstellen und an Ihre App anhängen kann:
     - Wählen Sie Neue Servicerolle erstellen und verwenden aus.
   - Um eine Servicerolle anzuhängen, die Sie zuvor erstellt haben:
     a. Wählen Sie Bestehende Servicerolle verwenden aus.
     b. Wählen Sie die zu verwendende Rolle aus der Liste aus.
6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite Überprüfen die Option Speichern und bereitstellen aus.

Einstellungen für die Datei package.json

Wenn Sie eine Next.js App bereitstellen, überprüft Amplify das Build-Skript der App in der `package.json` Datei, um den Anwendungstyp zu bestimmen.

Das Folgende ist ein Beispiel für das Build-Skript für eine Next.js App. Das Build-Skript `"next build"` gibt an, dass die App sowohl SSG- als auch SSR-Seiten unterstützt. Dieses Build-Skript wird auch für SSG-Apps von Next.js 14 oder höher verwendet.

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

Im Folgenden finden Sie ein Beispiel für das Build-Skript für eine SSG-App vom Typ Next.js 13 oder früher. Das Build-Skript `"next build && next export"` gibt an, dass die App nur SSG-Seiten unterstützt.

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

Amplify Sie die Build-Einstellungen für eine Next.js SSR-Anwendung

Nach der Überprüfung der `package.json` Datei Ihrer App überprüft Amplify die Build-Einstellungen für die App. Sie können die Build-Einstellungen in der Amplify-Konsole oder in einer `amplify.yml` Datei im Stammverzeichnis Ihres Repositorys speichern. Weitere Informationen finden Sie unter Konfiguration der Build-Einstellungen für eine Amplify-Anwendung.

Wenn Amplify feststellt, dass Sie eine Next.js SSR-App bereitstellen und keine `amplify.yml` Datei vorhanden ist, generiert es eine Buildspec für die App und setzt auf. `baseDirectory .next` Wenn Sie eine App bereitstellen, in der eine `amplify.yml` Datei vorhanden ist, überschreiben die Build-Einstellungen in der Datei alle Build-Einstellungen in der Konsole. Daher müssen Sie den Wert `.next` in der Datei manuell `baseDirectory` auf setzen.

Im Folgenden finden Sie ein Beispiel für die Build-Einstellungen für eine App, für die der Wert auf eingestellt `baseDirectory` ist`.next`. Dies weist darauf hin, dass die Build-Artefakte für eine Next.js -App bestimmt sind, die SSG- und SSR-Seiten unterstützt.

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

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

Amplify Sie die Build-Einstellungen für eine SSG-Anwendung von Next.js 13 oder früher

Wenn Amplify feststellt, dass Sie eine SSG-App von Next.js 13 oder früher bereitstellen, generiert es eine Build-Spezifikation für die App und setzt `baseDirectory` sie auf. `out` Wenn Sie eine App bereitstellen, in der eine `amplify.yml` Datei vorhanden ist, müssen Sie den Wert `out` in der Datei manuell `baseDirectory` auf setzen. Das `out` Verzeichnis ist der Standardordner, den Next.js zum Speichern exportierter statischer Elemente erstellt. Wenn Sie die Build-Spezifikationseinstellungen Ihrer App konfigurieren, ändern Sie den Namen des `baseDirectory` Ordners so, dass er der Konfiguration Ihrer App entspricht.

Im Folgenden finden Sie ein Beispiel für die Build-Einstellungen für eine App, bei der auf festgelegt `baseDirectory` ist, `out` um anzuzeigen, dass die Build-Artefakte für eine App von Next.js 13 oder früher bestimmt sind, die nur SSG-Seiten unterstützt.

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

Amplify Sie die Build-Einstellungen für eine SSG-Anwendung von Next.js 14 oder höher

In Next.js, Version 14, war der `next export` Befehl veraltet und wurde durch den Befehl `output: 'export'` in der `next.config.js` Datei ersetzt, um statische Exporte zu ermöglichen. Wenn Sie eine reine Next.js 14 SSG-Anwendung in der Konsole bereitstellen, generiert Amplify eine Buildspec für die App und setzt sie auf. `baseDirectory .next` Wenn Sie eine App bereitstellen, in der eine `amplify.yml` Datei vorhanden ist, müssen Sie das auf in der `baseDirectory` Datei manuell setzen. `.next` Dies ist dieselbe `baseDirectory` Einstellung, die Amplify für Next.js `WEB_COMPUTE` - Anwendungen verwendet, die sowohl SSG- als auch SSR-Seiten unterstützen.

Das Folgende ist ein Beispiel für die Build-Einstellungen für eine App nur Next.js 14 SSG mit der Einstellung auf. `baseDirectory .next`

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

Migrieren einer Next.js 11 SSR-App auf Amplify Hosting Compute

Wenn Sie eine neue Next.js App bereitstellen, verwendet Amplify standardmäßig die neueste unterstützte Version von Next.js. Derzeit unterstützt der Compute SSR-Anbieter Amplify Hosting Next.js Version 15.

Die Amplify-Konsole erkennt Apps in Ihrem Konto, die vor der Version des Amplify Hosting-Rechendienstes vom November 2022 mit voller Unterstützung für die Versionen 12 bis 15 von Next.js bereitgestellt wurden. Auf der Konsole wird ein Informationsbanner angezeigt, das Apps mit

Filialen identifiziert, die über den vorherigen SSR-Anbieter von Amplify, Classic, bereitgestellt wurden (nur Next.js 11). Wir empfehlen dringend, dass Sie Ihre Apps zum Amplify Hosting-Compute-SSR-Anbieter migrieren.

Wenn Sie Ihre gehostete Next.js 11-Anwendung auf Next.js 12 oder höher aktualisieren, wird möglicherweise eine `"target" property is no longer supported` Fehlermeldung angezeigt, wenn eine Bereitstellung ausgelöst wird. In diesem Fall müssen Sie zu Amplify Hosting Compute migrieren.

Sie müssen die App und alle ihre Produktionszweige gleichzeitig manuell migrieren. Eine App kann nicht sowohl die Zweige Classic (nur Next.js 11) als auch Next.js 12 oder höher enthalten.

Verwenden Sie die folgenden Anweisungen, um eine App zum Amplify Hosting-Compute-SSR-Anbieter zu migrieren.

Um eine App zum Amplify Hosting-Compute-SSR-Anbieter zu migrieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App Next.js aus, die Sie migrieren möchten.

   > **Note**
   >
   > Bevor Sie eine App in der Amplify-Konsole migrieren, müssen Sie zunächst die package.json-Datei der App aktualisieren, um Next.js Version 12 oder höher zu verwenden.

3. Wählen Sie im Navigationsbereich App-Einstellungen, Allgemein aus.
4. Auf der Startseite der App zeigt die Konsole ein Banner an, wenn die App über Zweige verfügt, die über den SSR-Anbieter Classic (nur Next.js 11) bereitgestellt wurden. Wählen Sie auf dem Banner die Option Migrieren aus.
5. Wählen Sie im Bestätigungsfenster für die Migration die drei Anweisungen aus und wählen Sie Migrieren aus.
6. Amplify erstellt Ihre App und stellt sie erneut bereit, um die Migration abzuschließen.

Eine SSR-Migration rückgängig machen

Wenn Sie eine Next.js App bereitstellen, erkennt Amplify Hosting die Einstellungen in Ihrer App und legt den internen Plattformwert für die App fest. Es gibt drei gültige Plattformwerte. Eine SSG-App ist

auf den Plattformwert `WEB` eingestellt. Eine SSR-App, die Next.js Version 11 verwendet, ist auf den Plattformwert gesetzt. `WEB_DYNAMIC` Eine SSR-App Next.js 12 oder höher ist auf den Plattformwert eingestellt. `WEB_COMPUTE`

Wenn Sie eine App mithilfe der Anweisungen im vorherigen Abschnitt migrieren, ändert Amplify den Plattformwert Ihrer App von `WEB_DYNAMIC` auf`WEB_COMPUTE`. Nachdem die Migration zu Amplify Hosting Compute abgeschlossen ist, können Sie die Migration in der Konsole nicht mehr rückgängig machen. Um die Migration rückgängig zu machen, müssen Sie den verwenden, AWS Command Line Interface um die Plattform der App wieder zu ändern. `WEB_DYNAMIC` Öffnen Sie ein Terminalfenster und geben Sie den folgenden Befehl ein, um die App-ID und die Region mit Ihren eindeutigen Informationen zu aktualisieren.

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

SSR-Funktionalität zu einer statischen Next.js App hinzufügen

Sie können SSR-Funktionalität zu einer vorhandenen statischen (SSG) Next.js App hinzufügen, die mit Amplify bereitgestellt wird. Bevor Sie mit der Konvertierung Ihrer SSG-App in SSR beginnen, aktualisieren Sie die App so, dass sie Next.js Version 12 oder höher verwendet, und fügen Sie SSR-Funktionen hinzu. Dann müssen Sie die folgenden Schritte ausführen.

1. Verwenden Sie die AWS Command Line Interface , um den Plattformtyp der App zu ändern.
2. Fügen Sie der App eine Servicerolle hinzu.
3. Aktualisieren Sie das Ausgabeverzeichnis in den Build-Einstellungen der App.
4. Aktualisieren Sie die `package.json` Datei der App, um anzugeben, dass die App SSR verwendet.

Aktualisierung der Plattform

Es gibt drei gültige Werte für den Plattformtyp. Eine SSG-App ist auf den Plattformtyp `WEB` eingestellt. Eine SSR-App, die Next.js Version 11 verwendet, ist auf den Plattformtyp eingestellt. `WEB_DYNAMIC` Für Apps, die auf Next.js 12 oder höher mithilfe von SSR bereitgestellt werden, das von Amplify Hosting Compute verwaltet wird, ist der Plattformtyp auf eingestellt. `WEB_COMPUTE`

Wenn Sie Ihre App als SSG-App bereitgestellt haben, hat Amplify den Plattformtyp auf gesetzt. `WEB` Verwenden Sie die AWS CLI , um die Plattform für Ihre App zu ändern. `WEB_COMPUTE` Öffnen Sie ein

Terminalfenster und geben Sie den folgenden Befehl ein. Aktualisieren Sie dabei den roten Text mit Ihrer eindeutigen App-ID und Region.

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

Eine Servicerolle hinzufügen

Eine Servicerolle ist die AWS Identity and Access Management (IAM) -Rolle, die Amplify übernimmt, wenn es in Ihrem Namen andere Dienste anruft. Folgen Sie diesen Schritten, um einer SSG-App, die bereits mit Amplify bereitgestellt wurde, eine Servicerolle hinzuzufügen.

Um eine Servicerolle hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wenn Sie in Ihrem Amplify-Konto noch keine Servicerolle erstellt haben, finden Sie Informationen unter Hinzufügen einer Servicerolle, um diesen vorausgesetzten Schritt abzuschließen.
3. Wählen Sie die statische App Next.js aus, zu der Sie eine Servicerolle hinzufügen möchten.
4. Wählen Sie im Navigationsbereich App-Einstellungen, Allgemein aus.
5. Wählen Sie auf der Seite mit den App-Details die Option Bearbeiten
6. Wählen Sie unter Servicerolle den Namen einer vorhandenen Servicerolle oder den Namen der Servicerolle, die Sie in Schritt 2 erstellt haben.
7. Wählen Sie Speichern.

Aktualisierung der Build-Einstellungen

Bevor Sie Ihre App erneut mit SSR-Funktionalität bereitstellen, müssen Sie die Build-Einstellungen für die App aktualisieren, um das Ausgabeverzeichnis auf festzulegen. `.next` Sie können die Build-Einstellungen in der Amplify-Konsole oder in einer in Ihrem Repo gespeicherten `amplify.yml` Datei bearbeiten. Weitere Informationen finden Sie unter Konfiguration der Build-Einstellungen für eine Amplify-Anwendung.

Das Folgende ist ein Beispiel für die Build-Einstellungen für eine App, bei der auf eingestellt `baseDirectory` ist. `.next`

```
version: 1
frontend:
```

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

Die Datei package.json wird aktualisiert

Nachdem Sie eine Servicerolle hinzugefügt und die Build-Einstellungen aktualisiert haben, aktualisieren Sie die Datei der App. `package.json` Stellen Sie wie im folgenden Beispiel das Build-Skript so ein, dass `"next build"` es angibt, dass die App Next.js sowohl SSG- als auch SSR-Seiten unterstützt.

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

Amplify erkennt die Änderung an der `package.json` Datei in Ihrem Repo und stellt die App mit SSR-Funktionalität erneut bereit.

Umgebungsvariablen für serverseitige Laufzeiten zugänglich machen

Amplify Hosting unterstützt das Hinzufügen von Umgebungsvariablen zu den Builds Ihrer Anwendung, indem Sie sie in der Projektkonfiguration in der Amplify-Konsole festlegen.

Eine Serverkomponente von Next.js hat jedoch standardmäßig keinen Zugriff auf diese Umgebungsvariablen. Dieses Verhalten dient dazu, alle Geheimnisse zu schützen, die in Umgebungsvariablen gespeichert sind, die Ihre Anwendung während der Erstellungsphase verwendet.

Um bestimmte Umgebungsvariablen für Next.js zugänglich zu machen, können Sie die Amplify-Build-Spezifikationsdatei so ändern, dass sie in den Umgebungsdateien festgelegt werden, die Next.js erkennt. Dadurch kann Amplify diese Umgebungsvariablen laden, bevor die Anwendung erstellt wird.

> **Important**
>
> Wir empfehlen dringend, keine Anmeldeinformationen, Geheimnisse oder vertraulichen Informationen in Ihren Umgebungsvariablen zu speichern, da jeder Benutzer mit Zugriff auf Bereitstellungsartefakte diese lesen kann.
> Um Ihrer SSR-Rechenfunktion Zugriff auf AWS Ressourcen zu gewähren, empfehlen wir die Verwendung von IAM-Rollen.

Das folgende Beispiel für eine Build-Spezifikation zeigt, wie Umgebungsvariablen im Abschnitt mit den Build-Befehlen hinzugefügt werden.

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

In diesem Beispiel enthält der Abschnitt mit den Build-Befehlen zwei Befehle, die Umgebungsvariablen in die `.env.production` Datei schreiben, bevor der Build der Anwendung ausgeführt wird. Amplify Hosting ermöglicht Ihrer Anwendung den Zugriff auf diese Variablen, wenn die Anwendung Datenverkehr empfängt.

Die folgende Zeile aus dem Abschnitt mit den Build-Befehlen im vorherigen Beispiel zeigt, wie Sie eine bestimmte Variable aus der Build-Umgebung nehmen und sie der `.env.production` Datei hinzufügen.

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

Wenn die Variablen in Ihrer Build-Umgebung existieren, enthält die `.env.production` Datei die folgenden Umgebungsvariablen.

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

Die folgende Zeile aus dem Abschnitt mit den Build-Befehlen im vorherigen Beispiel zeigt, wie Sie der `.env.production` Datei eine Umgebungsvariable mit einem bestimmten Präfix hinzufügen. In diesem Beispiel `NEXT_PUBLIC_` werden alle Variablen mit dem Präfix hinzugefügt.

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

Wenn in der Build-Umgebung mehrere Variablen mit dem `NEXT_PUBLIC_` Präfix existieren, sieht die `.env.production` Datei wie folgt aus.

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

SSR-Umgebungsvariablen für Monorepos

Wenn Sie eine SSR-App in einem Monorepo bereitstellen und bestimmte Umgebungsvariablen für Next.js zugänglich machen möchten, müssen Sie der `.env.production` Datei Ihr Anwendungsstammverzeichnis voranstellen. Die folgende Beispiel-Build-Spezifikation für eine Next.js -App in einem Nx-Monorepo zeigt, wie Umgebungsvariablen im Abschnitt mit den Build-Befehlen hinzugefügt werden.

```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
```

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

Die folgenden Zeilen aus dem Abschnitt mit den Build-Befehlen im vorherigen Beispiel zeigen, wie bestimmte Variablen aus der Build-Umgebung entnommen und zur `.env.production` Datei für eine App in einem Monorepo mit dem Anwendungsstamm hinzugefügt werden. `apps/app`

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

Bereitstellen einer Next.js -App in einem Monorepo

Amplify unterstützt Apps in generischen Monorepos sowie Apps in Monorepos, die mit npm workspace, pnpm workspace, Yarn workspace, Nx und Turborepo erstellt wurden. Wenn Sie Ihre App bereitstellen, erkennt Amplify automatisch das von Ihnen verwendete Monorepo-Build-Framework. Amplify wendet automatisch Build-Einstellungen für Apps in einem NPM-Workspace, Yarn-Workspace oder Nx an. Turborepo- und pnpm-Apps erfordern eine zusätzliche Konfiguration. Weitere Informationen finden Sie unter Konfiguration der Monorepo-Build-Einstellungen.

Ein detailliertes Nx-Beispiel finden Sie im Blogbeitrag Share code between Next.js apps with Nx on AWS Amplify Hosting.

Amplify Sie die Unterstützung für Nuxt.js

Nuxt ist ein Framework für die Erstellung von Full-Stack-Webanwendungen mit Vue.js.

Adapter

Sie können eine Nuxt.js -Anwendung für Amplify bereitstellen, indem Sie einen voreingestellten Adapter ohne Konfiguration verwenden. Weitere Informationen zum Adapter finden Sie in der Nuxt-Dokumentation.

Tutorial

Informationen zum Bereitstellen einer Nuxt.js App auf Amplify finden Sie unterStellen Sie eine Nuxt.js App für Amplify Hosting bereit.

Demo

Eine Videodemonstration finden Sie unter Nuxt Hosting With ZERO Configuration In Minutes (With AWS) on. YouTube

Amplify Sie die Unterstützung für Astro.js

Astro ist ein Web-Framework zur Erstellung inhaltsorientierter Webanwendungen.

Adapter

Sie können eine Astro.js -Anwendung mithilfe eines Community-Adapters für Amplify bereitstellen. Wir unterhalten keinen Amplify-eigenen Adapter für das Astro-Framework. Ein Adapter ist jedoch auf Github verfügbar. com/alexnguyennz/astro-aws-amplify auf der Website. GitHub Dieser Adapter wurde von einem Mitglied der Community erstellt und wird nicht von verwaltet. AWS

Tutorial

Informationen zum Bereitstellen einer Astro-App auf Amplify finden Sie unter. Stellen Sie eine Astro.js App für Amplify Hosting bereit

Demo

Eine Videodemonstration finden Sie unter So stellen Sie eine Astro-Website AWS auf dem Amazon Web Services YouTube Services-Kanal bereit.

Amplify Sie die Unterstützung für SvelteKit

SvelteKit ist ein Framework für die Erstellung von Full-Stack-Webanwendungen mit Svelte.

Adapter

Sie können eine SvelteKit Anwendung mithilfe eines Community-Adapters für Amplify bereitstellen. Wir unterhalten keinen Amplify-eigenen Adapter für das SvelteKit Framework. Ein Adapter ist jedoch auf Github verfügbar. com/gzimbron/amplify-Adapter auf der GitHub Website. Dieser Adapter wurde von einem Mitglied der Community erstellt und wird nicht von AWS verwaltet.

Tutorial

Informationen zum Bereitstellen einer SvelteKit App auf Amplify finden Sie unterStellen Sie eine SvelteKit App für Amplify Hosting bereit.

Demo

Eine Videodemonstration finden Sie unter So stellen Sie eine SvelteKit Website (mit API) AWS auf dem Amazon Web Services YouTube Services-Kanal bereit.

Bereitstellung einer SSR-App für Amplify

Sie können diese Anweisungen verwenden, um eine App bereitzustellen, die mit einem beliebigen Framework erstellt wurde, mit einem Bereitstellungspaket, das der von Amplify erwarteten Build-Ausgabe entspricht. Wenn Sie eine Next.js -Anwendung bereitstellen, ist kein Adapter erforderlich.

Wenn Sie eine SSR-App bereitstellen, die einen Framework-Adapter verwendet, müssen Sie zuerst den Adapter installieren und konfigurieren. Detaillierte Anweisungen finden Sie unter Verwendung von Open-Source-Adaptern für jedes SSR-Framework.

Um eine SSR-App für Amplify Hosting bereitzustellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite „Alle Apps“ die Option Neue App erstellen aus.
3. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.
4. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:

    a. Wählen Sie den Namen des Repositorys aus, zu dem Sie eine Verbindung herstellen möchten.

   b. Wählen Sie den Namen des Repository-Zweigs aus, zu dem eine Verbindung hergestellt werden soll.
   c. Wählen Sie Weiter aus.
5. Auf der Seite mit den App-Einstellungen erkennt Amplify automatisch Next.js SSR-Apps.

   Wenn Sie eine SSR-App bereitstellen, die einen Adapter für ein anderes Framework verwendet, müssen Sie Amazon CloudWatch Logs explizit aktivieren. Öffnen Sie den Abschnitt Erweiterte Einstellungen und wählen Sie dann im Abschnitt Serverseitiges Rendering (SSR) -Bereitstellung die Option SSR-App-Logs aktivieren aus.
6. Für die App ist eine IAM-Servicerolle erforderlich, von der Amplify übernimmt, um Protokolle an Ihre zu senden. AWS-Konto

   Das Verfahren zum Hinzufügen einer Servicerolle hängt davon ab, ob Sie eine neue Rolle erstellen oder eine bestehende verwenden möchten.
   - Um eine neue Rolle zu erstellen:
     - Wählen Sie Neue Servicerolle erstellen und verwenden aus.
   - Um eine bestehende Rolle zu verwenden:
     a. Wählen Sie Eine bestehende Rolle verwenden aus.
     b. Wählen Sie in der Liste der Servicerollen die zu verwendende Rolle aus.
7. Wählen Sie Weiter aus.
8. Wählen Sie auf der Seite Überprüfen die Option Speichern und bereitstellen aus.

Von SSR unterstützte Funktionen

Dieser Abschnitt enthält Informationen zur Unterstützung von SSR-Funktionen durch Amplify.

Amplify bietet Versionsunterstützung für Node.js, die der Version von Node.js entspricht, die zum Erstellen Ihrer App verwendet wurde.

Amplify bietet eine integrierte Bildoptimierungsfunktion, die alle SSR-Apps unterstützt. Wenn Sie die Standard-Bildoptimierungsfunktion nicht verwenden möchten, können Sie einen benutzerdefinierten Bildoptimierungslader implementieren.

Topics

- Bildoptimierung für SSR-Apps
- Amazon CloudWatch Logs für SSR-Apps
- Amplify Sie die SSR-Unterstützung von Next.js 11

Unterstützung der Version Node.js für Apps von Next.js

Wenn Amplify eine Next.js Compute-App erstellt und bereitstellt, verwendet es die Node.js Laufzeitversion, die der Hauptversion entsprichtNode.js, die zum Erstellen der App verwendet wurde.

> **Note**
>
> Ab dem 15. September 2025 unterstützt Amplify-Hosting die Laufzeiten Node.js 14, Node.js 16 und Node.js 18 nicht mehr. Zu den unterstützten Laufzeiten gehören Node.js 20 und Node.js 22.

Sie können die Node.js Version angeben, die in der Funktion zum Überschreiben von Live-Paketen in der Amplify-Konsole verwendet werden soll. Weitere Informationen zur Konfiguration von Live-Paket-Updates finden Sie unterVerwenden bestimmter Paket- und Abhängigkeitsversionen im Build-Image. Sie können die Node.js Version auch mithilfe anderer Mechanismen angeben, z. B. mit nvm Befehlen. Wenn Sie keine Version angeben, verwendet Amplify standardmäßig die aktuelle Version, die vom Amplify-Build-Container verwendet wird.

Bildoptimierung für SSR-Apps

Amplify Hosting bietet eine integrierte Bildoptimierungsfunktion, die alle SSR-Apps unterstützt. Mit der Bildoptimierung von Amplify können Sie qualitativ hochwertige Bilder im richtigen Format, der richtigen Größe und Auflösung für das Gerät, das auf sie zugreift, liefern und gleichzeitig die kleinstmögliche Dateigröße beibehalten.

Derzeit können Sie entweder die Bildkomponente Next.js verwenden, um Bilder bei Bedarf zu optimieren, oder Sie können einen benutzerdefinierten Bildlader implementieren. Wenn Sie Next.js 13 oder höher verwenden, müssen Sie keine weiteren Maßnahmen ergreifen, um die Bildoptimierungsfunktion von Amplify zu verwenden. Wenn Sie einen benutzerdefinierten Loader implementieren, lesen Sie das folgende Thema Verwenden eines benutzerdefinierten Image-Loaders.

Verwenden eines benutzerdefinierten Image-Loaders

Wenn Sie einen benutzerdefinierten Bildlader verwenden, erkennt Amplify den Loader in der `next.config.js` Datei Ihrer Anwendung und nutzt die integrierte Bildoptimierungsfunktion nicht. Weitere Informationen zu den benutzerdefinierten Loadern, die Next.js unterstützt, finden Sie in der Bilddokumentation zu Next.js.

Amazon CloudWatch Logs für SSR-Apps

Amplify sendet Informationen über Ihre SSR-Laufzeit an Amazon CloudWatch Logs in Ihrem. AWS-Konto Wenn Sie eine SSR-App bereitstellen, benötigt die App eine IAM-Dienstrolle, die Amplify übernimmt, wenn es andere Dienste in Ihrem Namen aufruft. Sie können entweder Amplify Hosting Compute erlauben, automatisch eine Servicerolle für Sie zu erstellen, oder Sie können eine Rolle angeben, die Sie erstellt haben.

Wenn Sie Amplify erlauben, eine IAM-Rolle für Sie zu erstellen, verfügt die Rolle bereits über die Berechtigungen zum Erstellen CloudWatch von Protokollen. Wenn Sie Ihre eigene IAM-Rolle erstellen, müssen Sie Ihrer Richtlinie die folgenden Berechtigungen hinzufügen, damit Amplify auf Amazon CloudWatch Logs zugreifen kann.

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

Weitere Informationen zu Servicerollen finden Sie unter Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen.

Amplify Sie die SSR-Unterstützung von Next.js 11

Wenn Sie vor der Veröffentlichung von Amplify Hosting Compute am 17. November 2022 eine Next.js App für Amplify bereitgestellt haben, verwendet Ihre App den vorherigen SSR-Anbieter von Amplify, Classic (nur Next.js 11). Die Dokumentation in diesem Abschnitt gilt nur für Apps, die mit dem SSR-Anbieter Classic (Next.js 11 nur) bereitgestellt wurden.

> **Note**
>
> Wir empfehlen dringend, dass Sie Ihre Next.js 11-Apps zum Compute Managed SSR-Anbieter von Amplify Hosting migrieren. Weitere Informationen finden Sie unter Migrieren einer Next.js 11 SSR-App auf Amplify Hosting Compute.

In der folgenden Liste werden die spezifischen Funktionen beschrieben, die der SSR-Anbieter Amplify Classic (nur Next.js 11) unterstützt.

Unterstützte Funktionen

- Serverseitig gerenderte Seiten (SSR)
- Statische Seiten
- API-Routen
- Dynamische Routen
- Erfasse alle Routen
- SSG (Statische Generierung)
- Inkrementelle statische Regeneration (ISR)
- Internationalisiertes (i18n) Unterpfad-Routing
- Umgebungsvariablen

Nicht unterstützte Funktionen

- Bildoptimierung
- Inkrementelle statische Regeneration (ISR) auf Anfrage
- Internationalisiertes (i18n) Domain-Routing
- Internationalisierte (i18n) automatische Erkennung von Gebietsschemas
- Middleware
- Edge-Middleware
- Edge-API-Routen

Preise für Next.js 11 SSR-Apps

Bei der Bereitstellung Ihrer Next.js 11 SSR-App erstellt Amplify zusätzliche Backend-Ressourcen in Ihrem AWS Konto, darunter:

- Ein Amazon Simple Storage Service (Amazon S3) -Bucket, der die Ressourcen für die statischen Ressourcen Ihrer App speichert. Informationen zu den Amazon S3 S3-Gebühren finden Sie unter Amazon S3 S3-Preise.
- Eine CloudFront Amazon-Distribution zur Bereitstellung der App. Informationen zu CloudFront Gebühren finden Sie unter CloudFront Amazon-Preise.
- Vier Lambda @Edge -Funktionen zur Anpassung der bereitgestellten Inhalte CloudFront .

AWS Identity and Access Management Berechtigungen für Next.js 11 SSR-Apps

Amplify benötigt AWS Identity and Access Management (IAM-) Berechtigungen, um eine SSR-App bereitzustellen. Für SSR-Apps stellt Amplify Ressourcen wie einen Amazon S3 S3-Bucket, eine CloudFront Distribution, Lambda@Edge Funktionen, eine Amazon SQS SQS-Warteschlange (falls ISR verwendet wird) und IAM-Rollen bereit. Ohne die erforderlichen Mindestberechtigungen wird beim Versuch, Ihre SSR-App bereitzustellen, eine `Access Denied` Fehlermeldung angezeigt. Um Amplify die erforderlichen Berechtigungen zu gewähren, müssen Sie eine Servicerolle angeben.

Informationen zum Erstellen einer IAM-Servicerolle, die Amplify übernimmt, wenn Sie andere Dienste in Ihrem Namen aufrufen, finden Sie unter. Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen Diese Anweisungen zeigen, wie Sie eine Rolle erstellen, die die verwaltete Richtlinie anhängt. `AdministratorAccess-Amplify`

Die `AdministratorAccess-Amplify` verwaltete Richtlinie bietet Zugriff auf mehrere AWS Dienste, einschließlich IAM-Aktionen. Sie sollte als ebenso leistungsfähig wie die Richtlinie angesehen werden. `AdministratorAccess` Diese Richtlinie bietet mehr Berechtigungen, als für die Bereitstellung Ihrer SSR-App erforderlich sind.

Es wird empfohlen, die bewährte Methode der Gewährung der geringsten Rechte zu befolgen und die der Servicerolle gewährten Berechtigungen zu reduzieren. Anstatt Administratorzugriffsberechtigungen für Ihre Servicerolle zu gewähren, können Sie Ihre eigene, vom Kunden verwaltete IAM-Richtlinie erstellen, die nur die für die Bereitstellung Ihrer SSR-App erforderlichen Berechtigungen gewährt. Anweisungen zur Erstellung einer vom Kunden verwalteten Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Wenn Sie Ihre eigene Richtlinie erstellen, finden Sie in der folgenden Liste die Mindestberechtigungen, die für die Bereitstellung einer SSR-App erforderlich sind.

```
acm:DescribeCertificate
acm:DescribeCertificate
acm:ListCertificates
acm:RequestCertificate
cloudfront:CreateCloudFrontOriginAccessIdentity
cloudfront:CreateDistribution
cloudfront:CreateInvalidation
cloudfront:GetDistribution
cloudfront:GetDistributionConfig
cloudfront:ListCloudFrontOriginAccessIdentities
cloudfront:ListDistributions
cloudfront:ListDistributionsByLambdaFunction
cloudfront:ListDistributionsByWebACLId
cloudfront:ListFieldLevelEncryptionConfigs
cloudfront:ListFieldLevelEncryptionProfiles
cloudfront:ListInvalidations
cloudfront:ListPublicKeys
cloudfront:ListStreamingDistributions
cloudfront:UpdateDistribution
cloudfront:TagResource
cloudfront:UntagResource
cloudfront:ListTagsForResource
iam:AttachRolePolicy
iam:CreateRole
iam:CreateServiceLinkedRole
iam:GetRole
iam:PutRolePolicy
iam:PassRole
lambda:CreateFunction
lambda:EnableReplication
lambda:DeleteFunction
lambda:GetFunction
lambda:GetFunctionConfiguration
lambda:PublishVersion
lambda:UpdateFunctionCode
lambda:UpdateFunctionConfiguration
lambda:ListTags
lambda:TagResource
lambda:UntagResource
route53:ChangeResourceRecordSets
```

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

Problembehandlung bei Next.js 11 SSR-Bereitstellungen

Wenn bei der Bereitstellung einer Classic (nur Next.js 11) SSR-App mit Amplify unerwartete Probleme auftreten, lesen Sie sich die folgenden Themen zur Fehlerbehebung durch.

Topics

- Meine Anwendung Next.js verwendet Funktionen, die nicht unterstützt werden
- Bilder in meiner Next.js -Anwendung werden nicht geladen
- Nicht unterstützte Regionen

Das Ausgabeverzeichnis meiner Anwendung wurde überschrieben

Das Ausgabeverzeichnis für eine mit Amplify bereitgestellte App Next.js muss auf `.next` eingestellt sein. Wenn das Ausgabeverzeichnis Ihrer App überschrieben wird, überprüfen Sie die Datei. `next.config.js` Wenn das Build-Ausgabeverzeichnis standardmäßig den Wert haben soll`.next`, entfernen Sie die folgende Zeile aus der Datei:

```
distDir: 'build'
```

Stellen Sie sicher, dass das Ausgabeverzeichnis `.next` in Ihren Build-Einstellungen auf eingestellt ist. Informationen zum Anzeigen der Build-Einstellungen Ihrer App finden Sie unterKonfiguration der Build-Einstellungen für eine Amplify-Anwendung.

Im Folgenden finden Sie ein Beispiel für die Build-Einstellungen für eine App, für die die Einstellung auf festgelegt `baseDirectory` ist`.next`.

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

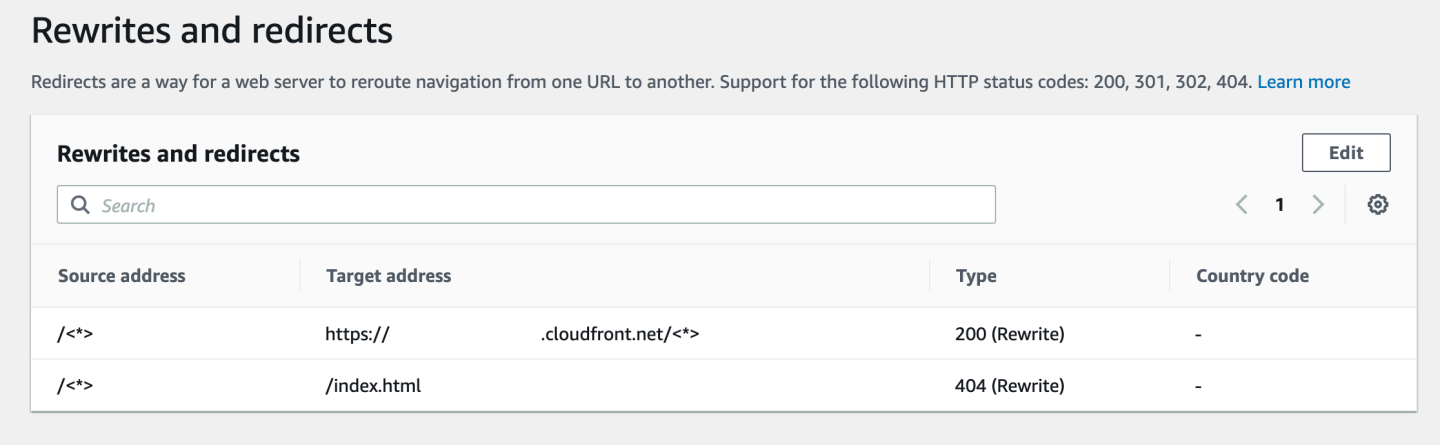
Nach der Bereitstellung meiner SSR-Site erhalte ich eine 404-Fehlermeldung

Wenn Sie nach der Bereitstellung Ihrer Site einen 404-Fehler erhalten, könnte das Problem daran liegen, dass Ihr Ausgabeverzeichnis überschrieben wurde. Gehen Sie wie im vorherigen Thema beschrieben vor, um Ihre `next.config.js` Datei und das richtige Build-Ausgabeverzeichnis in der Build-Spezifikation Ihrer App zu überprüfen. Das Ausgabeverzeichnis meiner Anwendung wurde überschrieben

In meiner Anwendung fehlt die Rewrite-Regel für CloudFront SSR-Distributionen

Wenn Sie eine SSR-App bereitstellen, erstellt Amplify eine Rewrite-Regel für Ihre CloudFront SSR-Distributionen. Wenn Sie in einem Webbrowser nicht auf Ihre App zugreifen können, stellen Sie sicher, dass die CloudFront Rewrite-Regel für Ihre App in der Amplify-Konsole existiert. Wenn sie fehlt, können Sie sie entweder manuell hinzufügen oder Ihre App erneut bereitstellen.

Um die Umschreib- und Umleitungsregeln einer App in der Amplify-Konsole anzuzeigen oder zu bearbeiten, wählen Sie im Navigationsbereich App-Einstellungen und dann Umschreibungen und Weiterleitungen aus. Der folgende Screenshot zeigt ein Beispiel für die Rewrite-Regeln, die Amplify für Sie erstellt, wenn Sie eine SSR-App bereitstellen. Beachten Sie, dass in diesem Beispiel eine CloudFront Rewrite-Regel existiert.



Meine Anwendung ist zu groß für die Bereitstellung

Amplify begrenzt die Größe einer SSR-Bereitstellung auf 50 MB. Wenn Sie versuchen, eine Next.js SSR-App für Amplify bereitzustellen und eine `RequestEntityTooLargeException` Fehlermeldung erhalten, ist Ihre App zu groß für die Bereitstellung. Sie können versuchen, dieses Problem zu umgehen, indem Sie Ihrer Datei Code zur Cache-Bereinigung hinzufügen. `next.config.js`

Im Folgenden finden Sie ein Beispiel für Code in der `next.config.js` Datei, der die Cache-Bereinigung durchführt.

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

Mein Build schlägt mit einem Fehler fehl, dass nicht genügend Arbeitsspeicher zur Verfügung steht

Next.js ermöglicht es Ihnen, Build-Artefakte zwischenzuspeichern, um die Leistung bei nachfolgenden Builds zu verbessern. Darüber hinaus komprimiert der AWS CodeBuild Container von Amplify diesen Cache und lädt ihn in Ihrem Namen auf Amazon S3 hoch, um die nachfolgende Build-Leistung zu verbessern. Dies könnte dazu führen, dass Ihr Build mit einem Fehler wegen unzureichenden Speichers fehlschlägt.

Führen Sie die folgenden Aktionen aus, um zu verhindern, dass Ihre App das Speicherlimit während der Erstellungsphase überschreitet. Entfernen Sie es zunächst `.next/cache/**/*` aus dem Abschnitt cache.paths Ihrer Build-Einstellungen. Als Nächstes entfernen Sie die `NODE_OPTIONS` Umgebungsvariable aus Ihrer Build-Einstellungsdatei. Stellen Sie stattdessen die `NODE_OPTIONS` Umgebungsvariable in der Amplify-Konsole ein, um das maximale Speicherlimit für den Knoten zu definieren. Weitere Informationen zum Setzen von Umgebungsvariablen mit der Amplify-Konsole finden Sie unterUmgebungsvariablen setzen.

Nachdem Sie diese Änderungen vorgenommen haben, versuchen Sie es erneut mit Ihrem Build. Wenn dies erfolgreich ist, fügen Sie `.next/cache/**/*` dem Abschnitt cache.paths Ihrer Build-Einstellungsdatei etwas hinzu.

Weitere Informationen zur Cache-Konfiguration von Next.js zur Verbesserung der Build-Leistung finden Sie unter AWS CodeBuild auf der Website Next.js.

Meine Anwendung hat sowohl SSR- als auch SSG-Zweige

Sie können keine App bereitstellen, die sowohl über SSR- als auch über SSG-Zweige verfügt. Wenn Sie sowohl SSR- als auch SSG-Branches bereitstellen müssen, müssen Sie eine App bereitstellen, die nur SSR-Branches verwendet, und eine andere App, die nur SSG-Zweige verwendet.

Meine Anwendung speichert statische Dateien in einem Ordner mit einem reservierten Pfad

Next.js kann statische Dateien aus einem Ordner mit dem Namen `public` bereitstellen, der im Stammverzeichnis des Projekts gespeichert ist. Wenn Sie eine Next.js App mit Amplify bereitstellen und hosten, darf Ihr Projekt keine Ordner mit dem Pfad `public/static` enthalten. Amplify behält sich den `public/static` Pfad für die Verwendung bei der Verteilung der App vor. Wenn Ihre App diesen Pfad enthält, müssen Sie den `static` Ordner umbenennen, bevor Sie ihn mit Amplify bereitstellen.

Meine Anwendung hat ein CloudFront Limit erreicht

CloudFront Servicekontingenten beschränken Ihr AWS Konto auf 25 Distributionen mit angehängten Lambda @Edge -Funktionen. Wenn Sie dieses Kontingent überschreiten, können Sie entweder alle ungenutzten CloudFront Distributionen aus Ihrem Konto löschen oder eine Erhöhung des Kontingents beantragen. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Lambda @Edge -Funktionen werden in der Region USA Ost (Nord-Virginia) erstellt

Wenn Sie eine Next.js App bereitstellen, erstellt Amplify Lambda @Edge -Funktionen, um den bereitgestellten Inhalt anzupassen. CloudFront Lambda @Edge -Funktionen werden in der Region USA Ost (Nord-Virginia) erstellt, nicht in der Region, in der Ihre App bereitgestellt wird. Dies ist eine Lambda @Edge -Beschränkung. Weitere Informationen zu Lambda @Edge -Funktionen finden Sie unter Einschränkungen für Edge-Funktionen im Amazon CloudFront Developer Guide.

Meine Anwendung Next.js verwendet Funktionen, die nicht unterstützt werden

Mit Amplify bereitgestellte Apps unterstützen die Hauptversionen von Next.js bis Version 11. Eine ausführliche Liste der Funktionen von Next.js, die von Amplify unterstützt und nicht unterstützt werden, finden Sie unter. supported features

Wenn Sie eine neue Next.js App bereitstellen, verwendet Amplify standardmäßig die neueste unterstützte Version von Next.js. Wenn Sie über eine bestehende Next.js App verfügen, die Sie mit einer älteren Version von Next.js auf Amplify bereitgestellt haben, können Sie die App zum Amplify Hosting Compute SSR-Anbieter migrieren. Detaillierte Anweisungen finden Sie unter Migrieren einer Next.js 11 SSR-App auf Amplify Hosting Compute.

Bilder in meiner Next.js -Anwendung werden nicht geladen

Wenn Sie Ihrer App Next.js mithilfe der `next/image` Komponente Bilder hinzufügen, darf die Größe des Bilds 1 MB nicht überschreiten. Wenn Sie die App auf Amplify bereitstellen, geben Bilder,

die größer als 1 MB sind, einen 503-Fehler zurück. Dies wird durch ein Lambda @Edge -Limit verursacht, das die Größe einer Antwort, die von einer Lambda-Funktion generiert wird, einschließlich Header und Hauptteil, auf 1 MB beschränkt.

Das Limit von 1 MB gilt für andere Artefakte in Ihrer App, wie PDF- und Dokumentdateien.

Nicht unterstützte Regionen

Amplify unterstützt die klassische (nur Next.js 11) SSR-App-Bereitstellung nicht in jeder AWS Region, in der Amplify verfügbar ist. Classic (nur Next.js 11) SSR wird in den folgenden Regionen nicht unterstützt: Europa (Mailand) eu-south-1, Naher Osten (Bahrain) me-south-1 und Asien-Pazifik (Hongkong) ap-east-1.

Problembehandlung bei SSR-Bereitstellungen

Wenn bei der Bereitstellung einer SSR-App mit Amplify Hosting Compute unerwartete Probleme auftreten, finden Sie weitere Informationen Fehlerbehebung bei serverseitig gerenderten Anwendungen im Kapitel zur Fehlerbehebung bei Amplify.

Fortgeschritten: Open-Source-Adapter

Framework-Autoren können die Dateisystem-basierte Bereitstellungsspezifikation verwenden, um Open-Source-Build-Adapter zu entwickeln, die auf ihre spezifischen Frameworks zugeschnitten sind. Diese Adapter verwandeln die Build-Ausgabe einer App in ein Bereitstellungspaket, das der erwarteten Verzeichnisstruktur von Amplify Hosting entspricht. Dieses Bereitstellungspaket enthält alle zum Hosten einer App erforderlichen Dateien und Ressourcen, einschließlich der Laufzeitkonfiguration, z. B. Routing-Regeln.

Wenn Sie kein Framework verwenden, können Sie Ihre eigene Lösung entwickeln, um eine Build-Ausgabe zu generieren, die Amplify erwartet.

Themen

- Verwenden der Amplify Hosting-Bereitstellungsspezifikation zur Konfiguration der Build-Ausgabe
- Bereitstellen eines Express-Servers mithilfe des Deployment-Manifests
- Integration der Bildoptimierung für Framework-Autoren
- Verwendung von Open-Source-Adaptern für jedes SSR-Framework

Verwenden der Amplify Hosting-Bereitstellungsspezifikation zur Konfiguration der Build-Ausgabe

Die Amplify Hosting-Bereitstellungsspezifikation ist eine dateisystembasierte Spezifikation, die die Verzeichnisstruktur definiert, die Bereitstellungen für Amplify Hosting erleichtert. Ein Framework kann diese erwartete Verzeichnisstruktur als Ausgabe seines Build-Befehls generieren, sodass das Framework die Service Primitives von Amplify Hosting nutzen kann. Amplify Hosting versteht die Struktur des Bereitstellungspakets und stellt es entsprechend bereit.

Eine Videodemonstration, in der erklärt wird, wie die Bereitstellungsspezifikation verwendet wird, finden Sie unter How to host any website using AWS Amplify on the Amazon Web Services YouTube channel.

Das Folgende ist ein Beispiel für die Ordnerstruktur, die Amplify für das Bereitstellungspaket erwartet. Auf einer höheren Ebene hat es einen Ordner mit dem Namen`static`, einen Ordner mit dem Namen `compute` und eine Bereitstellungsmanifestdatei mit dem Namen`deploy-manifest.json`.

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

Amplify Sie die primitive SSR-Unterstützung

Die Amplify Hosting-Bereitstellungsspezifikation definiert einen Vertrag, der den folgenden Primitiven genau entspricht.

Statische Vermögenswerte

Bietet Frameworks die Möglichkeit, statische Dateien zu hosten.

Datenverarbeitung

Bietet Frameworks die Möglichkeit, einen Node.js HTTP-Server auf Port 3000 auszuführen.

Bildoptimierung

Stellt Frameworks einen Dienst zur Optimierung von Bildern zur Laufzeit zur Verfügung.

Routing-Regeln

Stellt Frameworks einen Mechanismus zur Verfügung, mit dem eingehende Anforderungspfade bestimmten Zielen zugeordnet werden können.

Das .amplify-hosting/static Verzeichnis

Sie müssen alle öffentlich zugänglichen statischen Dateien, die über die Anwendungs-URL bereitgestellt werden sollen, im `.amplify-hosting/static` Verzeichnis ablegen. Die Dateien in diesem Verzeichnis werden über das statische Asset-Primitiv bereitgestellt.

Auf statische Dateien kann im Stammverzeichnis (/) der Anwendungs-URL zugegriffen werden, ohne dass ihr Inhalt, ihr Dateiname oder ihre Erweiterung geändert werden. Darüber hinaus werden Unterverzeichnisse in der URL-Struktur beibehalten und vor dem Dateinamen angezeigt. Als Beispiel `.amplify-hosting/static/favicon.ico` wird von `https://myAppId.amplify-hostingapp.com/favicon.ico` und `.amplify-hosting/static/_nuxt/main.js` wird bedient von `https://myAppId.amplify-hostingapp.com/_nuxt/main.js`

Wenn ein Framework die Möglichkeit unterstützt, den Basispfad der Anwendung zu ändern, muss es den Basispfad den statischen Assets innerhalb des `.amplify-hosting/static` Verzeichnisses voranstellen. Wenn der Basispfad beispielsweise lautet`/folder1/folder2`, dann `main.css` wird die Build-Ausgabe für ein statisches Asset aufgerufen. `.amplify-hosting/static/folder1/folder2/main.css`

Das .amplify-hosting/compute Verzeichnis

Eine einzelne Rechenressource wird durch ein einzelnes Unterverzeichnis mit dem Namen repräsentiert, das innerhalb des `.amplify-hosting/compute` Verzeichnisses `default` enthalten ist. Der Pfad ist`.amplify-hosting/compute/default`. Diese Rechenressource ist dem Compute-Primitiv von Amplify Hosting zugeordnet.

Der Inhalt des `default` Unterverzeichnisses muss den folgenden Regeln entsprechen.

- Im Stammverzeichnis des `default` Unterverzeichnisses muss eine Datei vorhanden sein, die als Einstiegspunkt zur Rechenressource dient.
- Bei der Einstiegspunktdatei muss es sich um ein Modul Node.js handeln und sie muss einen HTTP-Server starten, der auf Port 3000 lauscht.
- Sie können andere Dateien im `default` Unterverzeichnis platzieren und über den Code in der Einstiegspunktdatei auf sie verweisen.
- Der Inhalt des Unterverzeichnisses muss eigenständig sein. Der Code im Einstiegspunktmodul kann auf keine Module außerhalb des Unterverzeichnisses verweisen. Beachten Sie, dass Frameworks ihren HTTP-Server nach Belieben bündeln können. Wenn der Rechenvorgang mit dem `node server.js` Befehl, wo sich der Name der Eintragsdatei `server.js is` befindet, aus dem Unterverzeichnis heraus initiiert werden kann, geht Amplify davon aus, dass die Verzeichnisstruktur der Deployment-Spezifikation entspricht.

Amplify Hosting bündelt und stellt alle Dateien im `default` Unterverzeichnis auf einer bereitgestellten Rechenressource bereit. Jeder Rechenressource werden 512 MB flüchtiger Speicher zugewiesen. Dieser Speicher wird nicht von allen Ausführungsinstanzen gemeinsam genutzt, sondern von nachfolgenden Aufrufen innerhalb derselben Ausführungsinstanz gemeinsam genutzt. Ausführungsinstanzen sind auf eine maximale Ausführungszeit von 15 Minuten begrenzt, und der einzige beschreibbare Pfad innerhalb der Ausführungsinstanz ist das Verzeichnis. `/tmp` Die unkomprimierte Größe jedes Rechenressourcenpakets darf 220 MB nicht überschreiten. Beispielsweise darf das `.amplify/compute/default` Unterverzeichnis 220 MB nicht überschreiten, wenn es unkomprimiert ist.

Die Datei .amplify-hosting/deploy-manifest.json

Verwenden Sie die `deploy-manifest.json` Datei, um die Konfigurationsdetails und Metadaten für eine Bereitstellung zu speichern. Eine `deploy-manifest.json` Datei muss mindestens ein

`version` Attribut, das `routes` Attribut mit einer angegebenen Catch-All-Route und das Attribut mit den `framework` angegebenen Framework-Metadaten enthalten.

Die folgende Objektdefinition veranschaulicht die Konfiguration für ein Bereitstellungsmanifest.

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

In den folgenden Themen werden die Details und die Verwendung der einzelnen Attribute im Bereitstellungsmanifest beschrieben.

Verwenden des Versionsattributs

Das `version` Attribut definiert die Version der Bereitstellungsspezifikation, die Sie implementieren. Derzeit ist Version 1 die einzige Version für die Amplify Hosting-Bereitstellungsspezifikation. Das folgende JSON-Beispiel demonstriert die Verwendung des `version` Attributs.

```
"version": 1
```

Verwenden des Route-Attributs

Das `routes` Attribut ermöglicht es Frameworks, das Primitiv der Amplify Hosting-Routing-Regeln zu nutzen. Routing-Regeln bieten einen Mechanismus zum Weiterleiten eingehender Anforderungspfade an ein bestimmtes Ziel im Bereitstellungspaket. Routing-Regeln geben nur das Ziel einer eingehenden Anfrage vor und werden angewendet, nachdem die Anfrage durch Umschreib- und Umleitungsregeln transformiert wurde. Weitere Informationen darüber, wie Amplify Hosting mit Umschreibungen und Weiterleitungen umgeht, finden Sie unter. Umleitungen und Umschreibungen für eine Amplify-Anwendung einrichten

Routing-Regeln schreiben oder transformieren die Anfrage nicht. Wenn eine eingehende Anfrage dem Pfadmuster für eine Route entspricht, wird die Anfrage unverändert an das Ziel der Route weitergeleitet.

Die im `routes` Array angegebenen Routing-Regeln müssen den folgenden Regeln entsprechen.

- Es muss eine Sammelroute angegeben werden. Eine Catch-All-Route hat das `/*` Muster, das allen eingehenden Anfragen entspricht.
- Das `routes` Array kann maximal 25 Elemente enthalten.
- Sie müssen entweder eine `Static` Route oder eine `Compute` Route angeben.
- Wenn Sie eine `Static` Route angeben, muss das `.amplify-hosting/static` Verzeichnis existieren.
- Wenn Sie eine `Compute` Route angeben, muss das `.amplify-hosting/compute` Verzeichnis existieren.
- Wenn Sie eine `ImageOptimization` Route angeben, müssen Sie auch eine `Compute` Route angeben. Dies ist erforderlich, da die Bildoptimierung für rein statische Anwendungen noch nicht unterstützt wird.

Die folgende Objektdefinition veranschaulicht die Konfiguration für das `Route` Objekt.

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

In der folgenden Tabelle werden die Eigenschaften des `Route` Objekts beschrieben.

Key (Schlüssel)	Typ	Erforderlich	Description
Pfad	Zeichenfolge	Ja	Definiert ein Muster, das den Pfaden eingehender Anfragen entspricht (ohne Querystring). Die maximale Pfadlänge beträgt 255 Zeichen. Ein Pfad muss mit einem Schrägstrich `/` beginnen.

Key (Schlüssel)	Typ	Erforderlich	Description
			Ein Pfad kann jedes der folgenden Zeichen enthalten: [A-Z], [a-z], [0-9], [_-.*$/~"'@: +]. Für den Musterver gleich werden nur die folgenden Platzhalt erzeichen unterstützt: • *(entspricht 0 oder mehr Zeichen) • Das /* Muster wird als Catch-All-Muster bezeichne t und entspricht allen eingehenden Anfragen.

Key (Schlüssel)	Typ	Erforderlich	Description
Ziel	Target	Ja	Ein Objekt, das das Ziel definiert, an das die übereinstimmende Anfrage weitergeleitet werden soll. Wenn eine `Compute` Route angegeben ist, `ComputeResource` muss eine entsprech ende vorhanden sein. Wenn eine `ImageOpti mization` Route angegeben ist, `imageSettings` muss diese ebenfalls angegeben werden.

Key (Schlüssel)	Typ	Erforderlich	Description
Reserve	Target	Nein	Ein Objekt, das das Ziel definiert, auf das zurückgegriffen werden soll, falls das ursprüngliche Ziel einen 404-Fehler zurückgibt. Die `target` Art und die `fallback` Art können für eine angegebene Route nicht identisch sein. Beispielsweise ist ein Fallback von `Static` bis nicht `Static` zulässig. Fallbacks werden nur für GET-Anfragen unterstützt, die keinen Hauptteil haben. Wenn in der Anfrage ein Hauptteil vorhanden ist, wird er während des Fallbacks gelöscht.

Die folgende Objektdefinition veranschaulicht die Konfiguration für das `Target` Objekt.

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

In der folgenden Tabelle werden die Eigenschaften des `Target` Objekts beschrieben.

Key (Schlüssel)	Typ	Erforderlich	Description
nett	Art der Zielperson	Ja	Und `enum` das definiert den Zieltyp. Gültige Werte sind `Static`, `Compute` und `ImageOpti mization` .
src	Zeichenfolge	Ja für `Compute` Nein für andere Primitive	Eine Zeichenfolge, die den Namen des Unterverzeichnisses im Bereitstellungspak et angibt, das den ausführbaren Code des Primitivs enthält. Gültig und nur für das Compute-Primitiv erforderlich. Der Wert muss auf eine der Rechenressourcen verweisen, die im Bereitstellungspak et vorhanden sind. Derzeit ist der einzige unterstützte Wert für dieses Feld`default`.
CacheControl	Zeichenfolge	Nein	Eine Zeichenfolge, die den Wert des Cache-Control-Head ers angibt, der auf die Antwort angewende t werden soll. Gilt nur für Static und

Key (Schlüssel)	Typ	Erforderlich	Description
			Primitive. ImageOptimization Der angegebene Wert wird durch benutzerdefinierte Header überschrieben. Weitere Informationen zu den Kunden-Headern von Amplify Hosting finden Sie unter. Benutzerdefinierte Header für eine Amplify-App einrichten **Note** Dieser Cache-Control-Header wird nur auf erfolgreiche Antworten angewendet, deren Statuscode auf 200 (OK) gesetzt ist.

Die folgende Objektdefinition veranschaulicht die Verwendung der Aufzählung`TargetKind`.

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

```
}
```

Die folgende Liste spezifiziert die gültigen Werte für die `TargetKind` Aufzählung.

Statisch

Leitet Anfragen an das statische Asset-Primitiv weiter.

Datenverarbeitung

Leitet Anfragen an das Compute-Primitiv weiter.

ImageOptimization

Leitet Anfragen an das Bildoptimierungs-Primitiv weiter.

Das folgende JSON-Beispiel demonstriert die Verwendung des `routes` Attributs mit mehreren angegebenen Routing-Regeln.

```
"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"
      }
    }
  ]
```

Weitere Informationen zur Angabe von Routing-Regeln in Ihrem Bereitstellungsmanifest finden Sie unter Bewährte Methoden für die Konfiguration von Routing-Regeln

Verwenden des ComputeResources-Attributs

Das `computeResources` Attribut ermöglicht es Frameworks, Metadaten zu den bereitgestellten Rechenressourcen bereitzustellen. Jeder Rechenressource muss eine entsprechende Route zugeordnet sein.

Die folgende Objektdefinition demonstriert die Verwendung für das `ComputeResource` Objekt.

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

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

In der folgenden Tabelle werden die Eigenschaften des `ComputeResource` Objekts beschrieben.

Key (Schlüssel)	Typ	Erforderlich	Description
name	Zeichenfolge	Ja	Gibt den Namen der Rechenressource an. Der Name muss mit dem Namen des Unterverz eichnisses innerhalb von übereinst immen. `.amplify-hosting/compute directory` Für Version 1 der Bereitstellungsspe zifikation ist `default` der einzig gültige Wert.
runtime	ComputeRuntime	Ja	Definiert die Laufzeit für die bereitgestellte Rechenressource. Gültige Werte sind `nodejs20.x` und `nodejs22.x` .
Einstiegspunkt	Zeichenfolge	Ja	Gibt den Namen der Startdatei an, von der aus Code für die angegeben e Rechenressource ausgeführt wird. Die Datei muss sich in dem Unterverzeichnis

Key (Schlüssel)	Typ	Erforderlich	Description
			befinden, das eine Rechenressource darstellt.

Wenn Sie eine Verzeichnisstruktur haben, die wie folgt aussieht.

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

Das JSON für das `computeResource` Attribut wird wie folgt aussehen.

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

Verwenden des ImageSettings-Attributs

Das `imageSettings` Attribut ermöglicht es Frameworks, das Verhalten des Bildoptimierungsprimitivs anzupassen, das eine On-Demand-Optimierung von Bildern zur Laufzeit ermöglicht.

Die folgende Objektdefinition veranschaulicht die Verwendung des `ImageSettings` Objekts.

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

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

In der folgenden Tabelle werden die Eigenschaften des `ImageSettings` Objekts beschrieben.

Key (Schlüssel)	Typ	Erforderlich	Description
Größen	Nummer []	Ja	Eine Reihe unterstützter Bildbreiten.
domains	Zeichenfolge []	Ja	Eine Reihe zulässiger externer Domänen, die die Bildoptimierung verwenden können. Lassen Sie das Array leer, damit nur die Bereitstellungsdomäne die Bildoptimierung verwenden kann.
RemotePatterns	RemotePattern[]	Ja	Eine Reihe zulässiger externer Muster, für die Bildoptimierung verwendet werden kann. Ähnlich wie Domänen, bietet jedoch mehr Kontrolle mit regulären Ausdrücken (Regex).
Formate	ImageFormat[]	Ja	Eine Reihe von zulässigen Ausgabebildformaten.
Minimaler CacheTTL	Zahl	Ja	Die Cache-Dauer in Sekunden für die optimierten Bilder.

Key (Schlüssel)	Typ	Erforderlich	Description
SVG ist gefährlich erlaubt	Boolesch	Ja	Erlaubt SVG-Eingabebild. URLs Dies ist aus Sicherheitsgründen standardmäßig deaktiviert.

Die folgende Objektdefinition demonstriert die Verwendung für das `RemotePattern` Objekt.

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

In der folgenden Tabelle werden die Eigenschaften des `RemotePattern` Objekts beschrieben.

Key (Schlüssel)	Typ	Erforderlich	Description
Protokoll	Zeichenfolge	Nein	Das Protokoll des zulässigen Remote-Patterns. Der einzige gültige Wert ist `https`.
hostname	Zeichenfolge	Ja	Der Hostname des zulässigen Remote-Patterns. Sie können ein Literal oder einen Platzhalter angeben. Ein einzelnes `*` entspricht einer einzelnen Subdomain. Ein doppeltes `**`

Key (Schlüssel)	Typ	Erforderlich	Description
			entspricht einer beliebigen Anzahl von Subdomains. Amplify erlaubt keine pauschalen Platzhalter, wenn nur `**` angegeben ist.
port	Zeichenfolge	Nein	Der Port des erlaubten Remote-Musters.
Pfadname	Zeichenfolge	Nein	Der Pfadname des zulässigen Remote-Patterns.

Das folgende Beispiel demonstriert das `imageSettings` Attribut.

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

```
    "dangerouslyAllowSVG": false
  }
```

Verwenden des Framework-Attributs

Verwenden Sie das `framework` Attribut, um Framework-Metadaten anzugeben.

Die folgende Objektdefinition veranschaulicht die Konfiguration für das `FrameworkMetadata` Objekt.

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

In der folgenden Tabelle werden die Eigenschaften des `FrameworkMetadata` Objekts beschrieben.

Key (Schlüssel)	Typ	Erforderlich	Description
name	Zeichenfolge	Ja	Der Name des Frameworks.
version	Zeichenfolge	Ja	Die Version des Frameworks. Es muss sich um eine gültige semantische Versionierungszeichenfolge (Semver) handeln.

Bewährte Methoden für die Konfiguration von Routing-Regeln

Routingregeln bieten einen Mechanismus zum Weiterleiten eingehender Anforderungspfade an bestimmte Ziele im Bereitstellungspaket. In einem Bereitstellungspaket können Framework-Autoren Dateien an die Build-Ausgabe ausgeben, die für eines der folgenden Ziele bereitgestellt werden:

- Primitiv für statische Assets — Dateien sind im `.amplify-hosting/static` Verzeichnis enthalten.

- Compute Primitive — Dateien sind im `.amplify-hosting/compute/default` Verzeichnis enthalten.

Framework-Autoren stellen in der Deploy-Manifestdatei auch eine Reihe von Routing-Regeln bereit. Jede Regel im Array wird mit der eingehenden Anfrage in sequentieller Reihenfolge abgeglichen, bis eine Übereinstimmung gefunden wird. Wenn es eine passende Regel gibt, wird die Anfrage an das in der Abgleichsregel angegebene Ziel weitergeleitet. Optional kann für jede Regel ein Fallback-Ziel angegeben werden. Wenn das ursprüngliche Ziel einen 404-Fehler zurückgibt, wird die Anfrage an das Ausweichziel weitergeleitet.

Gemäß der Bereitstellungsspezifikation muss es sich bei der letzten Regel in der Durchlaufreihenfolge um eine Sammelregel handeln. Eine Catch-All-Regel wird mit dem `/*` Pfad angegeben. Wenn die eingehende Anfrage mit keiner der vorherigen Routen im Routing-Regel-Array übereinstimmt, wird die Anfrage an das Ziel der Catch-All-Regel weitergeleitet.

Für SSR-Frameworks wie Nuxt.js muss das Ziel der Catch-All-Regel das Compute-Primitiv sein. Das liegt daran, dass SSR-Anwendungen serverseitig gerenderte Seiten mit Routen haben, die zum Zeitpunkt der Erstellung nicht vorhersehbar sind. Zum Beispiel, wenn eine Nuxt.js Anwendung eine Seite hat, auf der `/blog/[slug]` sich ein dynamischer Routenparameter `[slug]` befindet. Das Ziel der Catch-All-Regel ist die einzige Möglichkeit, Anfragen an diese Seiten weiterzuleiten.

Im Gegensatz dazu können spezifische Pfadmuster verwendet werden, um auf Routen zu zielen, die zum Zeitpunkt der Erstellung bekannt sind. Stellt beispielsweise statische Nuxt.js Elemente aus dem `/_nuxt` Pfad bereit. Das bedeutet, dass auf den `/_nuxt/*` Pfad eine bestimmte Routing-Regel angewendet werden kann, die Anfragen an das statische Asset-Primitiv weiterleitet.

Routing öffentlicher Ordner

Die meisten SSR-Frameworks bieten die Möglichkeit, veränderbare statische Assets aus einem `public` Ordner bereitzustellen. Dateien wie `favicon.ico` und `robots.txt` werden normalerweise im `public` Ordner gespeichert und über die Stamm-URL der Anwendung bereitgestellt. Die `favicon.ico` Datei wird beispielsweise von bereitgestellt`https://example.com/favicon.ico`. Beachten Sie, dass es für diese Dateien kein vorhersehbares Pfadmuster gibt. Sie werden fast ausschließlich durch den Dateinamen bestimmt. Die einzige Möglichkeit, Dateien innerhalb des `public` Ordners als Ziel zu verwenden, besteht darin, die Catch-All-Route zu verwenden. Das Ziel der Catch-All-Route muss jedoch das Compute-Primitiv sein.

Wir empfehlen einen der folgenden Ansätze für die Verwaltung Ihres `public` Ordners.

1. Verwenden Sie ein Pfadmuster, um auf Anforderungspfade zu zielen, die Dateierweiterungen enthalten. Sie können es beispielsweise als Ziel für alle Anforderungspfade verwenden `/*.*`, die eine Dateierweiterung enthalten.

   Beachten Sie, dass dieser Ansatz unzuverlässig sein kann. Wenn der `public` Ordner beispielsweise Dateien ohne Dateierweiterungen enthält, fallen sie nicht unter diese Regel. Ein weiteres Problem, das Sie bei diesem Ansatz beachten sollten, besteht darin, dass die Anwendung Seiten mit Punkten im Namen enthalten kann. Beispielsweise `/blog/2021/01/01/hello.world` wird die `/*.*` Regel als Zielseite für eine Seite verwendet. Dies ist nicht ideal, da es sich bei der Seite nicht um ein statisches Asset handelt. Sie können dieser Regel jedoch ein Fallback-Ziel hinzufügen, um sicherzustellen, dass bei einem 404-Fehler des statischen Primitivs die Anfrage auf das Compute-Primitiv zurückfällt.

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

2. Identifizieren Sie bei der Erstellung die Dateien im `public` Ordner und geben Sie für jede Datei eine Routing-Regel aus. Dieser Ansatz ist nicht skalierbar, da die Bereitstellungsspezifikation eine Obergrenze von 25 Regeln vorschreibt.

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

3. Empfehlen Sie Ihren Framework-Benutzern, alle veränderbaren statischen Assets in einem Unterordner innerhalb des `public` Ordners zu speichern.

   Im folgenden Beispiel kann der Benutzer alle veränderlichen statischen Assets im Ordner speichern. `public/assets` Anschließend `/assets/*` kann eine Routing-Regel mit dem Pfadmuster verwendet werden, um auf alle veränderbaren statischen Assets innerhalb des `public/assets` Ordners abzuzielen.

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

4. Geben Sie einen statischen Fallback für die Catch-All-Route an. Dieser Ansatz hat Nachteile, die im nächsten Abschnitt ausführlicher beschrieben werden. Catch-All-Fallback-Routing

Catch-All-Fallback-Routing

Für SSR-FrameworksNuxt.js, bei denen beispielsweise eine Catch-All-Route für das Compute-Primitivziel angegeben ist, könnten Framework-Autoren erwägen, einen statischen Fallback für die Catch-All-Route anzugeben, um das Ordnerrouting-Problem zu lösen. `public` Diese Art von Routing-Regel verstößt jedoch gegen serverseitig gerenderte 404-Seiten. Wenn der Endbenutzer beispielsweise eine Seite besucht, die nicht existiert, rendert die Anwendung eine 404-Seite mit dem Statuscode 404. Wenn die Catch-All-Route jedoch einen statischen Fallback hat, wird die 404-Seite nicht gerendert. Stattdessen fällt die Anfrage auf das statische Primitiv zurück und endet immer noch mit einem 404-Statuscode, aber die 404-Seite wird nicht gerendert.

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

Basispfad-Routing

Von Frameworks, die die Möglichkeit bieten, den Basispfad der Anwendung zu ändern, wird erwartet, dass sie den Basispfad den statischen Assets innerhalb des `.amplify-hosting/static` Verzeichnisses voranstellen. Wenn der Basispfad beispielsweise lautet`/folder1/folder2`, dann lautet die Build-Ausgabe für ein statisches Asset namens main.css. `.amplify-hosting/static/folder1/folder2/main.css`

Das bedeutet, dass auch die Routing-Regeln aktualisiert werden müssen, um den Basispfad widerzuspiegeln. Wenn der Basispfad beispielsweise lautet`/folder1/folder2`, dann sieht die Routing-Regel für die statischen Objekte im `public` Ordner wie folgt aus.

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

In ähnlicher Weise muss auch serverseitigen Routen der Basispfad vorangestellt werden. Wenn der Basispfad beispielsweise lautet`/folder1/folder2`, dann sieht die Routing-Regel für die `/api` Route wie folgt aus.

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

Der Basispfad sollte jedoch nicht der Sammelroute vorangestellt werden. Wenn der Basispfad beispielsweise lautet`/folder1/folder2`, bleibt die Catch-All-Route wie folgt.

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

```
}
```

Beispiele für Routen in Nuxt.js

Im Folgenden finden Sie eine `deploy-manifest.json` Beispieldatei für eine Nuxt-Anwendung, die zeigt, wie Routing-Regeln angegeben werden.

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

```
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
      "runtime": "nodejs22.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

Im Folgenden finden Sie eine `deploy-manifest.json` Beispieldatei für Nuxt, die zeigt, wie Routing-Regeln einschließlich Basispfaden angegeben werden.

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

```
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
      "runtime": "nodejs22.x"
    }
  ],
  "framework": {
    "name": "nuxt",
```

```
    "version": "3.8.1"
  }
}
```

Weitere Informationen zur Verwendung des `routes` Attributs finden Sie unterVerwenden des Route-Attributs.

Bereitstellen eines Express-Servers mithilfe des Deployment-Manifests

In diesem Beispiel wird erklärt, wie ein einfacher Express-Server mithilfe der Amplify Hosting-Bereitstellungsspezifikation bereitgestellt wird. Sie können das bereitgestellte Bereitstellungsmanifest nutzen, um Routing, Rechenressourcen und andere Konfigurationen anzugeben.

Richten Sie vor der Bereitstellung auf Amplify Hosting lokal einen Express-Server ein

1. Erstellen Sie ein neues Verzeichnis für Ihr Projekt und installieren Sie Express und 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. Fügen Sie dem Stammverzeichnis Ihres Projekts eine `tsconfig.json` Datei mit dem folgenden Inhalt hinzu.

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

```
  "exclude": ["node_modules"]
}
```

3. Erstellen Sie ein Verzeichnis mit dem Namen `src` in Ihrem Projektstamm.
4. Erstellen Sie eine `index.ts` Datei im `src` Verzeichnis. Dies ist der Einstiegspunkt zu der Anwendung, die einen Express-Server startet. Der Server sollte so konfiguriert sein, dass er auf Port 3000 lauscht.

```
// 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. Fügen Sie Ihrer `package.json` Datei die folgenden Skripts hinzu.

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

6. Erstellen Sie ein Verzeichnis mit dem Namen `public` im Stammverzeichnis Ihres Projekts. Erstellen Sie dann eine Datei `hello-world.txt` mit dem folgenden Namen.

```
Hello world!
```

7. Fügen Sie Ihrem Projektstamm eine `.gitignore` Datei mit dem folgenden Inhalt hinzu.

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

Richten Sie das Amplify-Bereitstellungsmanifest ein

1. Erstellen Sie eine Datei mit dem Namen `deploy-manifest.json` im Stammverzeichnis Ihres Projekts.
2. Kopieren Sie das folgende Manifest und fügen Sie es in Ihre `deploy-manifest.json` Datei ein.

```
{
  "version": 1,
  "framework": { "name": "express", "version": "4.18.2" },
  "imageSettings": {
```

```
    "sizes": [
      100,
      200,
      1920
    ],
    "domains": [],
    "remotePatterns": [],
    "formats": [],
    "minimumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  },
  "routes": [
    {
      "path": "/_amplify/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static",
        "cacheControl": "public, max-age=2"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "runtime": "nodejs22.x",
      "entrypoint": "index.js"
    }
```

```
  ]
}
```

Das Manifest beschreibt, wie Amplify Hosting die Bereitstellung Ihrer Anwendung handhaben sollte. Die primären Einstellungen sind die folgenden.

- Version — Gibt die Version der Bereitstellungsspezifikation an, die Sie verwenden.
- Framework — Passen Sie dies an, um Ihr Express Server-Setup zu spezifizieren.
- ImageSettings — Dieser Abschnitt ist für einen Express Server optional, es sei denn, Sie kümmern sich um die Bildoptimierung.
- Routen — Diese sind entscheidend, um den Verkehr in die richtigen Bereiche Ihrer App zu leiten. Die `"kind": "Compute"` Route leitet den Verkehr an Ihre Serverlogik weiter.
- ComputeResources — Verwenden Sie diesen Abschnitt, um die Laufzeit und den Express Einstiegspunkt Ihres Servers anzugeben.

Als Nächstes richten Sie ein Post-Build-Skript ein, das die erstellten Anwendungsartefakte in das Bereitstellungspaket verschiebt. `.amplify-hosting` Die Verzeichnisstruktur entspricht der Amplify Hosting-Bereitstellungsspezifikation.

Richten Sie das Post-Build-Skript ein

1. Erstellen Sie ein Verzeichnis mit dem Namen `bin` in Ihrem Projektstamm.
2. Erstellen Sie eine Datei mit dem Namen `postbuild.sh` im `bin` Verzeichnis. Fügen Sie der Datei `postbuild.sh` die folgenden Inhalte hinzu.

```
#!/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. Fügen Sie Ihrer `package.json` Datei ein `postbuild` Skript hinzu. Die Datei sollte wie folgt aussehen.

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

4. Führen Sie den folgenden Befehl aus, um Ihre Anwendung zu erstellen.

```
npm run build
```

5. (Optional) Passen Sie Ihre Routen für Express an. Sie können die Routen in Ihrem Bereitstellungsmanifest so ändern, dass sie zu Ihrem Express-Server passen. Wenn Sie beispielsweise keine statischen Assets im `public` Verzeichnis haben, benötigen Sie möglicherweise nur die Catch-All-Route, die zu `"path": "/*"` Compute führt. Das hängt von der Einrichtung Ihres Servers ab.

Ihre endgültige Verzeichnisstruktur sollte wie folgt aussehen.

```
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
```

Stellen Sie Ihren Server bereit

1. Pushen Sie Ihren Code in Ihr Git-Repository und stellen Sie Ihre App dann auf Amplify Hosting bereit.
2. Aktualisieren Sie Ihre Build-Einstellungen `.amplify-hosting` so, dass `baseDirectory` sie auf Folgendes verweisen. Während des Builds erkennt Amplify die Manifest-Datei im `.amplify-hosting` Verzeichnis und stellt Ihren Express-Server wie konfiguriert bereit.

   ```
   version: 1
   frontend:
     phases:
       preBuild:
         commands:
           - nvm use 18
           - npm install
       build:
         commands:
           - npm run build
     artifacts:
       baseDirectory: .amplify-hosting
       files:
         - '**/*'
   ```

3. Um zu überprüfen, ob Ihre Bereitstellung erfolgreich war und Ihr Server ordnungsgemäß läuft, besuchen Sie Ihre App unter der von Amplify Hosting bereitgestellten Standard-URL.

Integration der Bildoptimierung für Framework-Autoren

Framework-Autoren können die Bildoptimierungsfunktion von Amplify mithilfe der Amplify Hosting-Bereitstellungsspezifikation integrieren. Um die Bildoptimierung zu aktivieren, muss Ihr Bereitstellungsmanifest eine Routing-Regel enthalten, die auf den Bildoptimierungsdienst abzielt. Das folgende Beispiel zeigt, wie die Routingregel konfiguriert wird.

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

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

Weitere Informationen zur Konfiguration der Einstellungen für die Bildoptimierung mithilfe der Bereitstellungsspezifikation finden Sie unterVerwenden der Amplify Hosting-Bereitstellungsspezifikation zur Konfiguration der Build-Ausgabe.

Grundlegendes zur Bildoptimierungs-API

Die Bildoptimierung kann zur Laufzeit über die Domain-URL einer Amplify-App auf dem in der Routing-Regel definierten Pfad aufgerufen werden.

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

Bei der Bildoptimierung gelten die folgenden Regeln für Bilder.

- Amplify kann die Formate GIF, APNG und SVG nicht optimieren oder in ein anderes Format konvertieren.
- SVG-Bilder werden nur bereitgestellt, wenn die `dangerouslyAllowSVG` Einstellung aktiviert ist.
- Die Breite oder Höhe von Quellbildern darf 11 MB oder 9.000 Pixel nicht überschreiten.
- Die Größenbeschränkung für ein optimiertes Bild beträgt 4 MB.
- HTTPS ist das einzige Protokoll, das für die Beschaffung von Bildern per Fernzugriff unterstützt wird URLs.

HTTP-Header

Der HTTP-Header der Accept-Anfrage wird verwendet, um die Bildformate, ausgedrückt als MIME-Typen, anzugeben, die vom Client (normalerweise ein Webbrowser) zugelassen werden. Der

Bildoptimierungsdienst versucht, das Bild in das angegebene Format zu konvertieren. Der für diesen Header angegebene Wert hat eine höhere Priorität als der Formatabfrageparameter. Ein gültiger Wert für den Accept-Header ist beispielsweise`image/png, image/webp, */*` . Die im Amplify-Bereitstellungsmanifest angegebene Formateinstellung beschränkt die Formate auf die in der Liste aufgeführten. Selbst wenn der Accept-Header nach einem bestimmten Format fragt, wird es ignoriert, wenn das Format nicht in der Zulassungsliste enthalten ist.

URI-Anfrageparameter

In der folgenden Tabelle werden die URI-Anforderungsparameter für die Bildoptimierung beschrieben.

Abfragepa rameter	Typ	Erforderlich	Description	Beispiel
URL	Zeichenfolge	Ja	Ein relativer Pfad oder eine absolute URL zum Quellbild . Für eine Remote-URL wird nur das HTTPS-Protokoll unterstützt. Der Wert muss URL-codiert sein.	`?url=https%3A%2F%2Fwww.example.com%2Fbuffalo.png`
width	Zahl	Ja	Die Breite des optimierten Bilds in Pixeln.	`?width=800`
height	Zahl	Nein	Die Höhe des optimierten Bilds in Pixeln. Wenn nicht angegeben , wird das Bild auto an die	`?height=600`

Abfragepa rameter	Typ	Erforderlich	Description	Beispiel
			Breite angepasst .	
fit	Aufzählun gswerte:`cover`,`co` `inside` `outside`	Nein	Wie das Bild an die angegebene Breite und Höhe angepasst wird.	`?width=80 0&height= 600&fit=c over`
position	Aufzählun gswerte:`center`,,`t` `right` `bottom` `left`	Nein	Eine Position, die verwendet werden soll, wenn fit `cover` oder `contain` ist.	`?fit=cont ain&posit ion=centre`
trim	Zahl	Nein	Schneidet Pixel an allen Kanten ab, die Werte enthalten, die der angegeben en Hintergru ndfarbe des Pixels oben links ähneln.	`?trim=50`

Abfragepa rameter	Typ	Erforderlich	Description	Beispiel
erweitern	Objekt	Nein	Fügt den Bildrändern Pixel hinzu, wobei die Farbe verwendet wird, die von den Pixeln am nächsten Rand abgeleitet wurde. Beim Format gibt jeder Wert die Anzahl der Pixel `{top}_{right}_{bottom}_{left}` an, die hinzugefügt werden sollen.	`?extend=10_0_5_0`
extract	Objekt	Nein	Schneidet das Bild auf das angegebene Rechteck zu, das durch oben, links, Breite und Höhe begrenzt ist. Das Format ist {left} _ {top} _ {width} _ {right}, wobei jeder Wert die Anzahl der Pixel ist, die zugeschnitten werden sollen.	`?extract=10_0_5_0`

Abfragepa rameter	Typ	Erforderlich	Description	Beispiel
Format	Zeichenfolge	Nein	Das gewünscht e Ausgabeformat für das optimiert e Bild.	`?format=w ebp`
quality	Zahl	Nein	Die Qualität des Bildes, von 1 bis 100. Wird nur bei der Konvertierung des Bildformats verwendet.	`?quality=50`
rotate	Zahl	Nein	Dreht das Bild um den angegebenen Winkel in Grad.	`?rotate=45`
Flip	Boolesch	Nein	Spiegelt das Bild vertikal (von oben nach unten) auf der X-Achse. Dies geschieht immer vor der Drehung, falls vorhanden.	`?flip`
Flop	Boolesch	Nein	Spiegelt das Bild horizontal (links-rechts) auf der Y-Achse. Dies geschieht immer vor der Drehung, falls vorhanden.	`?flop`

Abfragepa rameter	Typ	Erforderlich	Description	Beispiel
schärfen	Zahl	Nein	Durch das Schärfen wird die Definition der Kanten im Bild verbesser t. Gültige Werte liegen zwischen 0,000001 und 10.	`?sharpen=1`
median	Zahl	Nein	Wendet einen Medianfilter an. Dadurch werden Bildrauschen entfernt oder die Kanten eines Bilds geglättet.	`?sharpen=3`
verwischen	Zahl	Nein	Wendet eine Gaußsche Unschärfe mit dem angegeben en Sigma an. Gültige Werte liegen zwischen 0,3 und 1.000.	`?blur=20`

Abfragepa rameter	Typ	Erforderlich	Description	Beispiel
gamma	Zahl	Nein	Wendet eine Gammakorr ektur an, um die wahrgenom mene Helligkei t eines Bildes mit geänderte r Größe zu verbessern. Der Wert muss zwischen 1,0 und 3,0 liegen.	`?gamma=1`
negieren	Boolesch	Nein	Kehrt die Farben des Bildes um.	`?negate`
normalisieren	Boolesch	Nein	Verbessert den Bildkontr ast, indem die Luminanz so ausgedehnt wird, dass ein voller Dynamikbereich abgedeckt wird.	`?normalize`

Abfragepa rameter	Typ	Erforderlich	Description	Beispiel
threshold	Zahl	Nein	Ersetzt jedes Pixel im Bild durch ein schwarzes Pixel, wenn dessen Intensität unter dem angegebenen Schwellenwert liegt. Oder mit einem weißen Pixel, wenn es über dem Schwellenwert liegt. Gültige Werte liegen zwischen 0 und 255.	`?threshold=155`
Farbton	Zeichenfolge	Nein	Färbt das Bild mit dem bereitgestellten RGB-Wert unter Beibehaltung der Bildleuchtdichte.	`?tint=#7743CE`
Graustufen	Boolesch	Nein	Wandelt das Bild in Graustufen (Schwarzweiß) um.	`?grayscale`

Statuscodes für Antworten

In der folgenden Liste werden die Antwortstatuscodes für die Bildoptimierung beschrieben.

Erfolg — HTTP-Statuscode 200

Die Anfrage wurde erfolgreich erfüllt.

BadRequest - HTTP-Statuscode 400

- Ein Eingabeabfrageparameter wurde falsch angegeben.
- Die Remote-URL ist in der `remotePatterns` Einstellung nicht als zulässig aufgeführt.
- Die Remote-URL wird nicht in ein Bild aufgelöst.
- Die angeforderte Breite oder Höhe sind in der `sizes` Einstellung nicht als zulässig aufgeführt.
- Das angeforderte Bild ist SVG, aber die `dangerouslyAllowSvg` Einstellung ist deaktiviert.

Nicht gefunden — HTTP-Statuscode 404

Das Quellbild wurde nicht gefunden.

Inhalt zu groß — HTTP-Statuscode 413

Entweder das Quellbild oder das optimierte Bild überschreiten die maximal zulässige Größe in Byte.

Grundlegendes zum optimierten Zwischenspeichern von Bildern

Amplify Hosting speichert optimierte Bilder auf unserem CDN zwischen, sodass nachfolgende Anfragen an dasselbe Bild mit denselben Abfrageparametern vom Cache aus bedient werden. Die Gültigkeitsdauer (TTL) des Caches wird durch den Header gesteuert. `Cache-Control` In der folgenden Liste werden Ihre Optionen für die Angabe des `Cache-Control` Headers beschrieben.

- Verwenden des `Cache-Control` Schlüssels in der Routing-Regel, die auf die Bildoptimierung abzielt.
- Verwenden von benutzerdefinierten Headern, die in der Amplify-App definiert sind.
- Bei Remote-Bildern wird der vom Remote-Image zurückgegebene `Cache-Control` Header berücksichtigt.

Die in den Bildoptimierungseinstellungen `minimumCacheTTL` angegebenen Werte definieren die Untergrenze der Cache-Control max-age Direktive. Wenn beispielsweise eine Remote-Bild-URL mit a antwortet`Cache-Control s-max-age=10`, der Wert jedoch 60 `minimumCacheTTL` ist, dann wird 60 verwendet.

Verwendung von Open-Source-Adaptern für jedes SSR-Framework

Sie können jeden SSR-Framework-Build-Adapter verwenden, der für die Integration mit Amplify Hosting erstellt wurde. Jedes Framework, das einen Adapter anbietet, bestimmt, wie der Adapter konfiguriert und mit seinem Build-Prozess verbunden ist. In der Regel installieren Sie den Adapter als NPM-Entwicklungsabhängigkeit.

Nachdem Sie eine App mit einem Framework erstellt haben, erfahren Sie in der Dokumentation des Frameworks, wie Sie den Amplify Hosting-Adapter installieren und in der Konfigurationsdatei Ihrer Anwendung konfigurieren.

Erstellen Sie als Nächstes eine `amplify.yml` Datei im Stammverzeichnis Ihres Projekts. Stellen Sie in der `amplify.yml` Datei `baseDirectory` das Build-Ausgabeverzeichnis Ihrer Anwendung ein. Das Framework führt den Adapter während des Build-Prozesses aus, um die Ausgabe in das Amplify Hosting-Bereitstellungspaket umzuwandeln.

Der Name des Build-Ausgabeverzeichnisses kann beliebig sein, aber der `.amplify-hosting` Dateiname hat Bedeutung. Amplify sucht zuerst nach einem Verzeichnis, das `baseDirectory` als definiert ist. Falls es existiert, sucht Amplify dort nach der Build-Ausgabe. Wenn das Verzeichnis nicht existiert, sucht Amplify darin nach der Build-Ausgabe`.amplify-hosting`, auch wenn sie nicht vom Kunden definiert wurde.

Das Folgende ist ein Beispiel für die Build-Einstellungen für eine App. Die Einstellung `baseDirectory` gibt `.amplify-hosting` an, dass sich die Build-Ausgabe im `.amplify-hosting` Ordner befindet. Solange der Inhalt des `.amplify-hosting` Ordners der Amplify Hosting-Bereitstellungsspezifikation entspricht, wird die App erfolgreich bereitgestellt.

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

Nachdem Ihre App für die Verwendung eines Framework-Adapters konfiguriert wurde, können Sie sie auf Amplify Hosting bereitstellen. Detaillierte Anweisungen finden Sie unter Bereitstellung einer SSR-App für Amplify.

Bereitstellung einer statischen Website für Amplify aus einem Amazon S3 S3-Bucket

Sie können die Integration zwischen Amplify Hosting und Amazon S3 verwenden, um statische Website-Inhalte, auf denen gespeichert ist, S3 mit nur wenigen Klicks zu hosten. Die Bereitstellung bei Amplify Hosting bietet Ihnen die folgenden Vorteile und Funktionen.

- Automatische Bereitstellung im weltweit verfügbaren AWS Content Delivery Network (CDN), betrieben von CloudFront
- HTTPS-Unterstützung
- Verbinden Sie Ihre Website mithilfe der Amplify-Konsole ganz einfach mit einer benutzerdefinierten Domain
- Bringen Sie Ihre eigenen benutzerdefinierten SSL-Zertifikate mit
- Überwachen Sie Ihre Website mit integrierten Zugriffsprotokollen und CloudWatch Metriken
- Richten Sie einen Passwortschutz für Ihre Website ein
- Erstellen Sie eine Weiterleitung und schreiben Sie die Regeln in der Amplify-Konsole neu

Sie können den Bereitstellungsprozess von der Amplify-Konsole AWS CLI, dem oder dem AWS SDKs aus starten. Sie können Amplify nur von einem Amazon S3 S3-Allzweck-Bucket aus bereitstellen, der sich in Ihrem eigenen Konto befindet. Amplify unterstützt keinen kontoübergreifenden S3 Bucket-Zugriff.

Wenn Sie Ihre Anwendung von einem Amazon S3 S3-Allzweck-Bucket auf Amplify Hosting bereitstellen, basieren die AWS Gebühren auf dem Preismodell von Amplify. Weitere Informationen finden Sie unter AWS Amplify – Preise.

> **Important**
>
> Amplify Hosting ist nicht überall dort verfügbar, AWS-Regionen wo Amazon S3 verfügbar ist. Um eine statische Website in Amplify Hosting bereitzustellen, muss sich der Amazon-S3-Bucket für allgemeine Zwecke, der Ihre Website enthält, in einer Region befinden, in der Amplify verfügbar ist. Eine Liste der Regionen, in denen Amplify verfügbar ist, finden Sie unter Amplify-Endpunkte unter Allgemeine Amazon Web Services-Referenz.

In den folgenden Themen erfahren Sie, wie Sie eine statische Website von Amazon S3 auf Amplify Hosting bereitstellen und aktualisieren.

Themen

Bereitstellen einer statischen Website S3 über die Amplify-Konsole

Verwenden Sie die folgenden Anweisungen, um mithilfe der Amplify-Konsole eine neue statische Website aus einem Amazon S3 S3-Allzweck-Bucket bereitzustellen.

So stellen Sie mithilfe der Amplify-Konsole eine statische Website aus einem Amazon S3 S3-Allzweck-Bucket bereit

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite „Alle Apps“ die Option Neue App erstellen aus.
3. Wählen Sie auf der Seite Start building with Amplify die Option Deploy without Git aus.
4. Wählen Sie Weiter aus.
5. Gehen Sie auf der Seite Manuelles Deployment starten wie folgt vor.
    a. Geben Sie als App-Name den Namen Ihrer App ein.
    b. Geben Sie unter Branch-Name den Namen der Filiale ein, die bereitgestellt werden soll.
6. Wählen Sie als Methode Amazon S3 aus.
7. Wählen Sie für den S3Speicherort der Objekte, die Sie hosten möchten, die Option Browse aus. Wählen Sie den Amazon S3 S3-Allzweck-Bucket aus, den Sie verwenden möchten, und wählen Sie dann Präfix auswählen aus.
8. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus.

Erstellen einer Bucket-Richtlinie für die Bereitstellung einer statischen Website S3 mithilfe von AWS SDKs

Sie können das verwenden AWS SDKs , um eine statische Website von Amazon S3 auf Amplify Hosting bereitzustellen. Wenn Sie Ihre Website mithilfe eines SDK bereitstellen, müssen Sie Ihre eigene Bucket-Richtlinie erstellen, die Amplify Hosting die Erlaubnis erteilt, die Objekte in Ihrem S3 Bucket abzurufen.

Weitere Informationen zum Erstellen von Bucket-Richtlinien finden Sie unter Bucket-Richtlinien für Amazon S3 im Amazon Simple Storage Service-Benutzerhandbuch.

Die folgende Beispiel-Bucket-Richtlinie gewährt Amplify Hosting-Berechtigungen zum Auflisten von Buckets und zum Abrufen von Bucket-Objekten für die angegebene AWS-Konto Amplify-Anwendungs-ID und den angegebenen Zweig.

Zur Verwendung dieses Beispiels gehen Sie wie folgt vor:

- *amzn-s3-demo-website-bucket/prefix*Ersetzen Sie es durch den Namen des Buckets und das Präfix Ihrer Website.
- Ersetze es *111122223333* durch deine AWS-Konto ID.
- *region-id*Ersetzen Sie es durch AWS-Region das, in dem sich Ihre Amplify-Anwendung befindet, z. B. **us-east-1**
- *app_id*Ersetzen Sie es durch Ihre Amplify-Anwendungs-ID. Diese Informationen sind in der Amplify-Konsole verfügbar.
- Ersetzen Sie es *branch_name* durch Ihren Filialnamen.

> **Note**
>
> In Ihrer Bucket-Richtlinie `aws:SourceArn` muss es sich um einen URL-codierten (prozentualen) Branch-ARN ARN.

JSON

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

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

```
    ]
}
```

Aktualisierung einer statischen Website, die aus einem Bucket für Amplify bereitgestellt wurde S3

Wenn Sie eines der Objekte für eine statische Website in einem auf Amplify gehosteten S3 Allzweck-Bucket aktualisieren, müssen Sie die Anwendung erneut auf Amplify Hosting bereitstellen, damit die Änderungen wirksam werden. Amplify Hosting erkennt Änderungen am S3 Bucket nicht automatisch. Wir empfehlen Ihnen, die AWS Command Line Interface (CLI) zu verwenden, um Ihre Website zu aktualisieren.

Synchronisieren Sie Updates mit S3

Nachdem Sie Änderungen an den Projektdateien Ihrer Website vorgenommen haben, verwenden Sie den folgenden Befehl s3 sync, um die Änderungen, die Sie an Ihrem lokalen Quellverzeichnis vorgenommen haben, mit Ihrem Amazon S3 S3-Ziel-Bucket für allgemeine Zwecke zu synchronisieren. Um dieses Beispiel zu verwenden, *<source>* ersetzen Sie es durch den Namen Ihres lokalen Verzeichnisses und *<target>* durch den Namen Ihres Amazon S3 S3-Buckets.

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

Stellen Sie die Website erneut auf Amplify Hosting bereit

Verwenden Sie den folgenden Befehl amplify start-deployment, um Ihre aktualisierte Anwendung in einem Amazon S3 S3-Bucket erneut auf Amplify Hosting bereitzustellen. Um dieses Beispiel zu verwenden, *<app_id>* ersetzen Sie es durch die ID Ihrer Amplify-Anwendung, *<branch_name>* durch den Namen Ihrer Filiale sowie *s3://amzn-s3-demo-website-bucket/prefix* durch Ihren S3 Bucket und Ihr Präfix.

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

Aktualisierung einer S3 Bereitstellung zur Verwendung eines Buckets und eines Präfixes anstelle einer ZIP-Datei

Wenn Sie bereits eine bestehende statische Website aus einer ZIP-Datei in einem Amazon S3 S3-Allzweck-Bucket für Amplify Hosting bereitgestellt haben, können Sie die Anwendungsbereitstellung so aktualisieren, dass sie den Bucket-Namen und das Präfix verwendet, die die zu hostenden Objekte enthalten. Bei dieser Art der Bereitstellung müssen Sie keine separate Datei in Ihren Bucket hochladen, die den komprimierten Inhalt der Build-Ausgabe enthält.

Um eine statische Website von einer ZIP-Datei in den Bucket-Inhalt zu migrieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite Alle Apps den Namen der manuell bereitgestellten App aus, die Sie von der Verwendung einer ZIP-Datei zur direkten Verwendung der Anwendungsdateien migrieren möchten.
3. Wählen Sie auf der Übersichtsseite der Anwendung die Option Updates bereitstellen aus.
4. Wählen Sie auf der Seite Updates bereitstellen für Methode Amazon S3 aus.
5. Wählen Sie für den S3Speicherort der zu hostenden Objekte die Option Browse aus. Wählen Sie den Bucket aus, den Sie verwenden möchten, und wählen Sie dann Präfix auswählen aus.
6. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus.

Eine Anwendung auf Amplify ohne Git-Repository bereitstellen

Manuelle Bereitstellungen ermöglichen es Ihnen, Ihre Web-App mit Amplify Hosting zu veröffentlichen, ohne einen Git-Anbieter zu verbinden. Sie können einen komprimierten Ordner per Drag & Drop von Ihrem Desktop ziehen und Ihre Website in Sekundenschnelle hosten. Alternativ können Sie auf Assets in einem Amazon S3 S3-Bucket verweisen oder eine öffentliche URL zu dem Speicherort angeben, an dem Ihre Dateien gespeichert sind.

> **Note**
>
> Manuelle Bereitstellungen haben aufgrund von Einschränkungen beim Amazon S3 S3-Kopiervorgang eine maximale Größenbeschränkung für .zip-Dateien von 5 GB. Wenn eines Ihrer Build-Artefakte diese Größe überschreitet, sollten Sie erwägen, es in kleinere Archive aufzuteilen oder eine alternative Bereitstellungsmethode zu verwenden.

Für Amazon S3 können Sie auch AWS Lambda Auslöser einrichten, um Ihre Website jedes Mal zu aktualisieren, wenn neue Assets hochgeladen werden. Weitere Informationen zur Einrichtung dieses Szenarios finden Sie im Blogbeitrag Bereitstellen von auf Amazon S3, Dropbox oder Ihrem Desktop auf der AWS Amplify Konsole gespeicherten Dateien.

Amplify Hosting unterstützt keine manuellen Bereitstellungen für serverseitig gerenderte Apps (SSR). Weitere Informationen finden Sie unter Bereitstellung serverseitig gerenderter Anwendungen mit Amplify Hosting.

Manuelle Bereitstellungen per Drag-and-Drop

Um eine App manuell per Drag & Drop bereitzustellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie in der oberen rechten Ecke Neue App erstellen.
3. Wählen Sie auf der Seite Start building with Amplify die Option Deploy without Git aus. Wählen Sie anschließend Weiter.
4. Geben Sie auf der Seite Manuelle Bereitstellung starten als App-Name den Namen Ihrer App ein.

5. Geben Sie unter Branchenname einen aussagekräftigen Namen ein, z. B. **`development`** oder**`production`**.
6. Wählen Sie als Methode die Option Drag & Drop.
7. Ziehen Sie entweder einen Ordner per Drag & Drop von Ihrem Desktop in die Drop-Zone oder wählen Sie die Datei mit der Option „.zip-Ordner auswählen“ von Ihrem Computer aus. Bei der Datei, die Sie ziehen und ablegen oder auswählen, muss es sich um einen komprimierten Ordner handeln, der den Inhalt Ihrer Build-Ausgabe enthält.
8. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus.

Manuelle Amazon S3- oder URL-Bereitstellung

> **Note**
>
> Wenn Sie eine statische Website von aus bereitstellenS3, erfordert das folgende Verfahren, dass Sie einen komprimierten Ordner mit dem Inhalt Ihrer Build-Ausgabe in Ihren S3 Bucket hochladen. Wir empfehlen, dass Sie eine statische Website direkt S3 über den Bucket-Namen und das Präfix bereitstellen. Weitere Informationen zu diesem vereinfachten Verfahren finden Sie unterBereitstellung einer statischen Website für Amplify aus einem Amazon S3 S3-Bucket.

Um eine App manuell über Amazon S3 oder eine öffentliche URL bereitzustellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie in der oberen rechten Ecke Neue App erstellen.
3. Wählen Sie auf der Seite Start building with Amplify die Option Deploy without Git aus. Wählen Sie anschließend Weiter.
4. Geben Sie auf der Seite Manuelle Bereitstellung starten als App-Name den Namen Ihrer App ein.
5. Geben Sie unter Branchenname einen aussagekräftigen Namen ein, z. B. **`development`** oder**`production`**.
6. Wählen Sie als Methode entweder Amazon S3 oder Beliebige URL aus.
7. Das Verfahren zum Hochladen Ihrer Dateien hängt von der Upload-Methode ab.
   - Amazon S3

a. Wählen Sie für S3 location of objects to host„Durchsuchen S3“. Wählen Sie dann den Namen des Amazon S3 S3-Buckets aus der Liste aus. Zugriffskontrolllisten (ACLs) müssen für den ausgewählten Bucket aktiviert sein. Weitere Informationen finden Sie unter Fehlerbehebung beim Amazon S3 S3-Bucket-Zugriff für manuelle Bereitstellungen.

b. Wählen Sie den Namen der bereitzustellenden ZIP-Datei aus.

c. Wählen Sie „Präfix auswählen“.

- Beliebige URL

  - Geben Sie unter Ressourcen-URL die URL der bereitzustellenden ZIP-Datei ein.

8. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus.

> **Note**
>
> Achten Sie beim Erstellen des komprimierten Ordners darauf, dass Sie den Inhalt Ihrer Build-Ausgabe komprimieren und nicht den Ordner der obersten Ebene. Wenn Ihre Build-Ausgabe beispielsweise einen Ordner mit dem Namen „build“ oder „public“ generiert, navigieren Sie zunächst zu diesem Ordner, wählen Sie den gesamten Inhalt aus und komprimieren Sie ihn von dort aus. Wenn Sie dies nicht tun, wird die Fehlermeldung „Zugriff verweigert“ angezeigt, da das Stammverzeichnis der Site nicht ordnungsgemäß initialisiert wird.

Fehlerbehebung beim Amazon S3 S3-Bucket-Zugriff für manuelle Bereitstellungen

Wenn Sie einen Amazon S3-Bucket erstellen, verwenden Sie dessen Einstellung Amazon S3 Object Ownership, um zu steuern, ob Zugriffskontrolllisten (ACLs) für den Bucket aktiviert oder deaktiviert sind. Um eine App manuell aus einem Amazon S3 S3-Bucket auf Amplify bereitzustellen, ACLs muss sie im Bucket aktiviert sein.

Wenn Sie bei der Bereitstellung aus einem Amazon S3 S3-Bucket eine `AccessControlList` Fehlermeldung erhalten, wurde der Bucket mit ACLs deaktivierter Option erstellt und Sie müssen sie in der Amazon S3 S3-Konsole aktivieren. Anweisungen finden Sie unter Objektbesitz für einen vorhandenen Bucket festlegen im Amazon Simple Storage Service-Benutzerhandbuch.

Verwaltung der Build-Konfiguration für eine Amplify-Anwendung

Sie können die Build-Einstellungen und die Konfiguration für Ihre Amplify-Bereitstellungen anpassen. Wenn Sie eine Anwendung bereitstellen, erkennt Amplify automatisch das Frontend-Framework und die zugehörigen Build-Einstellungen. Sie können die Build-Einstellungen in der Build-Spezifikation der Anwendung (Buildspec) anpassen, um Umgebungsvariablen hinzuzufügen, Build-Befehle auszuführen und Build-Abhängigkeiten anzugeben.

Das Standard-Build-Image von Amplify enthält mehrere vorinstallierte Pakete und Abhängigkeiten. Sie können jedoch auch die Funktion für Live-Paketaktualisierungen verwenden, um entweder eine bestimmte Version anzugeben oder sicherzustellen, dass immer die neueste Version installiert ist. Wenn Sie bestimmte Abhängigkeiten haben, deren Installation während eines Builds mithilfe des Standardcontainers von Amplify lange dauert, können Sie Ihr eigenes benutzerdefiniertes Build-Image erstellen. Sie können auch die Größe der Build-Instanz anpassen, um Ihrer Anwendungsbereitstellung die benötigten CPU-, Arbeitsspeicher- und Festplattenspeicherressourcen zur Verfügung zu stellen.

Builds werden automatisch bei jedem Commit in dein Git-Repository und bei jeder neuen Bereitstellung initiiert. Du kannst die Funktion für eingehende Webhooks so einrichten, dass ein Build ohne Commit in dein Git-Repository initiiert wird.

Mit der Build-Benachrichtigungsfunktion kannst du Teammitgliedern Informationen über Erfolge und Misserfolge von Builds mitteilen.

Topics

Konfiguration der Build-Einstellungen für eine Amplify-Anwendung

Wenn Sie eine Anwendung bereitstellen, erkennt Amplify automatisch das Frontend-Framework und die zugehörigen Build-Einstellungen, indem es die `package.json` Datei der App in Ihrem Git-

Repository überprüft. Sie haben die folgenden Optionen zum Speichern der Build-Einstellungen Ihrer App:

- Speichern Sie die Build-Einstellungen in der Amplify-Konsole — Die Amplify-Konsole erkennt die Build-Einstellungen automatisch und speichert sie, sodass sie von der Amplify-Konsole abgerufen werden können. Amplify wendet diese Einstellungen auf alle Ihre Branches an, es sei denn, in Ihrem Repository ist eine `amplify.yml` Datei gespeichert.
- Speichern Sie die Build-Einstellungen in Ihrem Repository — Laden Sie die `amplify.yml` Datei herunter und fügen Sie sie dem Stammverzeichnis Ihres Repositorys hinzu.

> **Note**
>
> Die Build-Einstellungen sind nur dann im Hosting-Menü der Amplify-Konsole sichtbar, wenn eine App für die kontinuierliche Bereitstellung eingerichtet und mit einem Git-Repository verbunden ist. Anweisungen zu dieser Art der Bereitstellung finden Sie unter Erste Schritte.

Referenz zur Build-Spezifikation

Die Build-Spezifikation (buildspec) für eine Amplify-Anwendung ist eine Sammlung von YAML-Einstellungen und Build-Befehlen, die Amplify verwendet, um Ihren Build auszuführen. In der folgenden Liste werden diese Einstellungen und ihre Verwendung beschrieben.

version

Die Amplify YAML-Versionsnummer.

AppRoot

Der Pfad innerhalb des Repositorys, in dem sich diese Anwendung befindet. Wird ignoriert, sofern nicht mehrere Anwendungen definiert sind.

env

Fügen Sie diesem Abschnitt Umgebungsvariablen hinzu. Sie können Umgebungsvariablen auch mithilfe der Konsole hinzufügen.

Backend

Führen Sie Amplify CLI-Befehle aus, um im Rahmen einer kontinuierlichen Bereitstellung ein Backend bereitzustellen, Lambda-Funktionen oder GraphQL-Schemas zu aktualisieren.

Frontend

Führen Sie Frontend-Build-Befehle aus.

Test

Führen Sie Befehle während einer Testphase aus. Erfahren Sie, wie Sie Ihrer App Tests hinzufügen.

Phasen erstellen

Das Frontend, das Backend und der Test bestehen aus drei Phasen, die die Befehle darstellen, die während jeder Sequenz des Builds ausgeführt werden.

- PreBuild — Das PreBuild-Skript wird ausgeführt, bevor der eigentliche Build gestartet wird, aber nachdem Amplify Abhängigkeiten installiert hat.
- Build: Ihre Build-Befehle.
- PostBuild — Das Post-Build-Skript wird ausgeführt, nachdem der Build abgeschlossen ist und Amplify alle erforderlichen Artefakte in das Ausgabeverzeichnis kopiert hat.

Buildpath

Der Pfad, der zum Ausführen des Builds verwendet werden soll. Amplify verwendet diesen Pfad, um Ihre Build-Artefakte zu finden. Wenn Sie keinen Pfad angeben, verwendet Amplify beispielsweise den Monorepo-App-Root. `apps/app`

Artefakte>Basisverzeichnis

Das Verzeichnis, in dem Ihre Build-Artefakte existieren.

Artefakte>Dateien

Geben Sie Dateien aus Ihren Artefakten an, die Sie bereitstellen möchten. Geben Sie ein`**/*`, um alle Dateien einzubeziehen.

Cache

Gibt Abhängigkeiten zur Buildzeit an, z. B. den Ordner node_modules. Während des ersten Builds werden die hier angegebenen Pfade zwischengespeichert. Bei nachfolgenden Builds stellt Amplify den Cache auf denselben Pfaden wieder her, bevor es Ihre Befehle ausführt.

Amplify betrachtet alle bereitgestellten Cache-Pfade als relativ zu Ihrem Projektstamm. Amplify erlaubt jedoch kein Durchqueren außerhalb des Projektstamms. Wenn Sie beispielsweise einen absoluten Pfad angeben, wird der Build ohne Fehler erfolgreich ausgeführt, der Pfad wird jedoch nicht zwischengespeichert.

YAML-Syntaxreferenz für die Build-Spezifikation

Das folgende Beispiel einer Build-Spezifikation demonstriert die grundlegende YAML-Syntax.

```
version: 1
env:
  variables:
    key: value
backend:
  phases:
    preBuild:
      commands:
        - *enter command*
    build:
      commands:
        - *enter command*
    postBuild:
        commands:
        - *enter command*
frontend:
  buildpath:
  phases:
    preBuild:
      commands:
        - cd react-app
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    files:
        - location
        - location
    discard-paths: yes
    baseDirectory: location
  cache:
    paths:
        - path # A cache path relative to the project root
        - path # Traversing outside of the project root is not allowed
test:
  phases:
    preTest:
      commands:
        - *enter command*
```

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

Bearbeitung der Build-Spezifikation

Sie können die Build-Einstellungen einer Anwendung anpassen, indem Sie die Build-Spezifikation (Buildspec) in der Amplify-Konsole bearbeiten. Die Build-Einstellungen werden auf alle Branches in Ihrer App angewendet, mit Ausnahme der Branches, für die eine `amplify.yml` Datei im Git-Repository gespeichert ist.

Um die Build-Einstellungen in der Amplify-Konsole zu bearbeiten

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie die Build-Einstellungen bearbeiten möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Build-Einstellungen aus.
4. Wählen Sie auf der Seite mit den Build-Einstellungen im Abschnitt App-Build-Spezifikation die Option Bearbeiten aus.
5. Geben Sie im Fenster Build-Spezifikation bearbeiten Ihre Aktualisierungen ein.
6. Wählen Sie Speichern.

Sie können die in den folgenden Themen beschriebenen Beispiele verwenden, um Ihre Build-Einstellungen für bestimmte Szenarien zu aktualisieren.

Themen

- Festlegen branchenspezifischer Build-Einstellungen mithilfe von Skripten
- Einen Befehl einrichten, um zu einem Unterordner zu navigieren
- Bereitstellung des Backends mit dem Frontend für eine Gen-1-App

- Einstellung des Ausgabeordners
- Pakete als Teil eines Builds installieren
- Verwenden Sie eine private NPM-Registrierung
- Installieren von OS-Paketen
- Einstellung des Schlüsselwertspeichers für jeden Build
- Den Build für einen Commit überspringen
- Automatische Builds bei jedem Commit ausschalten
- Konfiguration von diff-basiertem Frontend, Build und Deployment
- Konfiguration von diff-basierten Backend-Builds für eine Gen-1-App

Festlegen branchenspezifischer Build-Einstellungen mithilfe von Skripten

Sie können Bash-Shell-Skripts verwenden, um verzweigungsspezifische Build-Einstellungen festzulegen. Das folgende Skript verwendet beispielsweise die Systemumgebungsvariable $, AWS_BRANCH um einen Befehlssatz auszuführen, wenn der Zweigname main ist, und einen anderen Befehlssatz, wenn der Zweigname dev ist.

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

Einen Befehl einrichten, um zu einem Unterordner zu navigieren

Bei Monorepos möchten Benutzer in der Lage sein, `cd` in einen Ordner zu gelangen, um den Build auszuführen. Nachdem Sie den `cd` Befehl ausgeführt haben, gilt er für alle Phasen Ihres Builds, sodass Sie den Befehl nicht in separaten Phasen wiederholen müssen.

```
version: 1
env:
  variables:
    key: value
frontend:
  phases:
    preBuild:
```

```
      commands:
        - cd react-app
        - npm ci
    build:
      commands:
        - npm run build
```

Bereitstellung des Backends mit dem Frontend für eine Gen-1-App

> **Note**
>
> Dieser Abschnitt gilt nur für Amplify Gen 1-Anwendungen. Ein Gen 1-Backend wird mit Amplify Studio und der Amplify-Befehlszeilenschnittstelle (CLI) erstellt.

Der `amplifyPush` Befehl ist ein Hilfsskript, das Ihnen bei Backend-Bereitstellungen hilft. Die Build-Einstellungen unten ermitteln automatisch die richtige Backend-Umgebung für die Bereitstellung der aktuellen Verzweigung.

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

Einstellung des Ausgabeordners

Mit den folgenden Build-Einstellungen wird der öffentliche Ordner als Ausgabeordner festgelegt.

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

Pakete als Teil eines Builds installieren

Sie können die `yarn` Befehle `npm` oder verwenden, um Pakete während des Builds zu installieren.

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

Verwenden Sie eine private NPM-Registrierung

Sie können eine private Registrierung als Umgebungsvariable hinzufügen oder Verweise darauf in Ihren Build-Einstellungen hinzufügen.

```
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
```

Installieren von OS-Paketen

Das AL2023 Image von Amplify führt Ihren Code mit einem nicht privilegierten Benutzer namens aus. `amplify` Amplify gewährt diesem Benutzer Rechte, Betriebssystembefehle mit dem `sudo` Linux-Befehl auszuführen. Wenn Sie Betriebssystempakete für fehlende Abhängigkeiten installieren möchten, können Sie Befehle wie `yum` und `rpm` with `sudo` verwenden.

Der folgende Beispiel-Build-Abschnitt demonstriert die Syntax für die Installation eines Betriebssystempakets mithilfe des `sudo` Befehls.

```
build:
```

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

Einstellung des Schlüsselwertspeichers für jeden Build

Das `envCache` stellt die Speicherung von Schlüsselwerten zur Build-Zeit bereit. Werte, die in gespeichert sind, `envCache` können nur während eines Builds geändert und beim nächsten Build wiederverwendet werden. Mithilfe von können wir Informationen über die bereitgestellte Umgebung speichern und sie dem Build-Container in aufeinanderfolgenden Builds zur Verfügung stellen. `envCache` Im Gegensatz zu Werten`envCache`, die in der gespeichert sind, werden Änderungen an Umgebungsvariablen während eines Builds nicht für future Builds beibehalten.

Beispielverwendung:

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

Den Build für einen Commit überspringen

Um einen automatischen Build für einen bestimmten Commit zu überspringen, fügen Sie den Text [skip-cd] am Ende der Commit-Nachricht ein.

Automatische Builds bei jedem Commit ausschalten

Sie können Amplify so konfigurieren, dass automatische Builds bei jedem Code-Commit deaktiviert werden. Wählen Sie zur Einrichtung App-Einstellungen, Branch-Einstellungen und suchen Sie dann den Abschnitt Branches, in dem die verbundenen Branches aufgelistet sind. Wählen Sie einen Zweig aus und klicken Sie dann auf Aktionen, auto Erstellung deaktivieren. Neue Commits für diesen Branch führen nicht mehr zu einem neuen Build.

Konfiguration von diff-basiertem Frontend, Build und Deployment

Sie können Amplify so konfigurieren, dass diff-basierte Frontend-Builds verwendet werden. Wenn diese Option aktiviert ist, versucht Amplify zu Beginn jedes Builds standardmäßig`appRoot`, einen Diff für Ihren oder den `/src/` Ordner auszuführen. Wenn Amplify keine Unterschiede feststellt, überspringt es die Schritte zum Erstellen, Testen (falls konfiguriert) und Bereitstellen des Frontends und aktualisiert Ihre gehostete App nicht.

Um das diff-basierte Frontend zu konfigurieren, zu erstellen und bereitzustellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie den Diff-basierten Frontend-Build und die Bereitstellung konfigurieren möchten.
3. Wählen Sie im Navigationsbereich Hosting, Umgebungsvariablen aus.
4. Wählen Sie im Abschnitt Umgebungsvariablen die Option Variablen verwalten aus.
5. Das Verfahren zur Konfiguration der Umgebungsvariablen hängt davon ab, ob Sie das diff-basierte Frontend-Build und Deploy aktivieren oder deaktivieren.
   - Um das diff-basierte Frontend zu aktivieren, erstellen und implementieren Sie es
     a. Geben **`AMPLIFY_DIFF_DEPLOY`**Sie im Abschnitt Variablen verwalten unter Variable den Wert ein.
     b. Geben Sie für Wert `true` ein.
   - Um das auf Diff basierende Frontend-Build und Deployment zu deaktivieren
     - Führen Sie eine der folgenden Aktionen aus:
       - Suchen **`AMPLIFY_DIFF_DEPLOY`**Sie im Abschnitt Variablen verwalten nach. Geben Sie für Wert `false` ein.
       - Entfernen Sie die `AMPLIFY_DIFF_DEPLOY` Umgebungsvariable.
6. Wählen Sie Speichern.

Optional können Sie die `AMPLIFY_DIFF_DEPLOY_ROOT` Umgebungsvariable so einstellen, dass sie den Standardpfad durch einen Pfad überschreibt, der sich auf das Stammverzeichnis Ihres Repositorys bezieht, z. B. `dist`

Konfiguration von diff-basierten Backend-Builds für eine Gen-1-App

> **Note**
>
> Dieser Abschnitt gilt nur für Amplify Gen 1-Anwendungen. Ein Gen 1-Backend wird mit Amplify Studio und der Amplify-Befehlszeilenschnittstelle (CLI) erstellt.

Sie können Amplify Hosting so konfigurieren, dass es diff-basierte Backend-Builds verwendet, indem Sie die `AMPLIFY_DIFF_BACKEND` Umgebungsvariable verwenden. Wenn Sie diff-basierte Backend-

Builds aktivieren, versucht Amplify zu Beginn jedes Builds, einen Diff für den `amplify` Ordner in Ihrem Repository auszuführen. Wenn Amplify keine Unterschiede feststellt, überspringt es den Backend-Build-Schritt und aktualisiert Ihre Backend-Ressourcen nicht. Wenn Ihr Projekt keinen `amplify` Ordner in Ihrem Repository hat, ignoriert Amplify den Wert der `AMPLIFY_DIFF_BACKEND` Umgebungsvariablen.

Wenn Sie derzeit benutzerdefinierte Befehle in den Build-Einstellungen Ihrer Backend-Phase angegeben haben, funktionieren bedingte Backend-Builds nicht. Wenn Sie möchten, dass diese benutzerdefinierten Befehle ausgeführt werden, müssen Sie sie in der Datei Ihrer App in die Frontend-Phase Ihrer Build-Einstellungen verschieben. `amplify.yml`

Um diff-basierte Backend-Builds zu konfigurieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die diff-basierte Backend-Builds konfiguriert werden sollen.
3. Wählen Sie im Navigationsbereich Hosting, Umgebungsvariablen aus.
4. Wählen Sie im Abschnitt Umgebungsvariablen die Option Variablen verwalten aus.
5. Das Verfahren zur Konfiguration der Umgebungsvariablen hängt davon ab, ob Sie diff-basierte Backend-Builds aktivieren oder deaktivieren.
   - So aktivieren Sie diff-basierte Backend-Builds
     a. Geben **`AMPLIFY_DIFF_BACKEND`**Sie im Abschnitt Variablen verwalten unter Variable den Wert ein.
     b. Geben Sie für Wert `true` ein.
   - Um Diff-basierte Backend-Builds zu deaktivieren
     - Führen Sie eine der folgenden Aktionen aus:
       - Suchen **`AMPLIFY_DIFF_BACKEND`**Sie im Abschnitt Variablen verwalten nach. Geben Sie für Wert `false` ein.
       - Entfernen Sie die `AMPLIFY_DIFF_BACKEND` Umgebungsvariable.
6. Wählen Sie Speichern.

Konfiguration der Monorepo-Build-Einstellungen

Wenn Sie mehrere Projekte oder Microservices in einem einzigen Repository speichern, wird dies als Monorepo bezeichnet. Sie können Amplify Hosting verwenden, um Anwendungen in einem

Monorepo bereitzustellen, ohne mehrere Build-Konfigurationen oder Branch-Konfigurationen zu erstellen.

Amplify unterstützt Apps in generischen Monorepos sowie Apps in Monorepos, die mit npm workspace, pnpm workspace, Yarn workspace, Nx und Turborepo erstellt wurden. Wenn Sie Ihre App bereitstellen, erkennt Amplify automatisch das von Ihnen verwendete Monorepo-Build-Tool. Amplify wendet automatisch Build-Einstellungen für Apps in einem NPM-Workspace, Yarn-Workspace oder Nx an. Turborepo- und pnpm-Apps erfordern eine zusätzliche Konfiguration. Weitere Informationen finden Sie unter Konfiguration der Turborepo- und pnpm-Monorepo-Apps.

Sie können die Build-Einstellungen für ein Monorepo in der Amplify-Konsole speichern oder die `amplify.yml` Datei herunterladen und zum Stammverzeichnis Ihres Repositorys hinzufügen. Amplify wendet die in der Konsole gespeicherten Einstellungen auf alle Ihre Branches an, es sei denn, es findet eine `amplify.yml` Datei in Ihrem Repository. Wenn eine `amplify.yml` Datei vorhanden ist, überschreiben ihre Einstellungen alle in der Amplify-Konsole gespeicherten Build-Einstellungen.

YAML-Syntaxreferenz für die Monorepo-Build-Spezifikation

Die YAML-Syntax für eine Monorepo-Build-Spezifikation unterscheidet sich von der YAML-Syntax für ein Repository, das eine einzelne Anwendung enthält. Bei einem Monorepo deklarieren Sie jedes Projekt in einer Liste von Anwendungen. Sie müssen für jede Anwendung, die Sie in Ihrer Monorepo-Build-Spezifikation deklarieren, den folgenden zusätzlichen `appRoot` Schlüssel angeben:

AppRoot

> Das Stammverzeichnis innerhalb des Repositorys, in dem die Anwendung gestartet wird. Dieser Schlüssel muss vorhanden sein und denselben Wert wie die `AMPLIFY_MONOREPO_APP_ROOT` Umgebungsvariable haben. Anweisungen zum Einstellen dieser Umgebungsvariablen finden Sie unterEinstellung der Umgebungsvariablen AMPLIFY_MONOREPO_APP_ROOT.

Das folgende Beispiel für eine Monorepo-Build-Spezifikation zeigt, wie mehrere Amplify-Anwendungen im selben Repo deklariert werden. Die beiden Apps`react-app`, und `angular-app` sind in der Liste deklariert. `applications` Der `appRoot` Schlüssel für jede App gibt an, dass sich die App im `apps` Stammordner des Repositorys befindet.

Das `buildpath` Attribut ist so eingestellt, dass `/` die App vom Monorepo-Projektstamm aus ausgeführt und erstellt wird. Das `baseDirectory` Attribut ist der relative Pfad von. `buildpath`

YAML-Syntax für die Monorepo-Build-Spezifikation

```
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*
```

Eine App, die die folgende Beispiel-Build-Spezifikation verwendet, wird unter dem Projektstamm erstellt und die Build-Artefakte befinden sich unter. `/packages/nextjs-app/.next`

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

Einstellung der Umgebungsvariablen AMPLIFY_MONOREPO_APP_ROOT

Wenn Sie eine in einem Monorepo gespeicherte App bereitstellen, muss die `AMPLIFY_MONOREPO_APP_ROOT` Umgebungsvariable der App denselben Wert haben wie der Pfad des App-Stammverzeichnisses, relativ zum Stammverzeichnis Ihres Repositorys. Zum Beispiel ein Monorepo `ExampleMonorepo` mit dem Namen eines Stammordners namens`apps`, der, `app1``app2`, enthält und die folgende Verzeichnisstruktur `app3` hat:

```
ExampleMonorepo
  apps
    app1
    app2
    app3
```

In diesem Beispiel ist der Wert der `AMPLIFY_MONOREPO_APP_ROOT` Umgebungsvariablen für`app1`. `apps/app1`

Wenn Sie eine Monorepo-App mit der Amplify-Konsole bereitstellen, legt die Konsole die `AMPLIFY_MONOREPO_APP_ROOT` Umgebungsvariable automatisch anhand des Werts fest, den Sie für den Pfad zum Stammverzeichnis der App angeben. Wenn Ihre Monorepo-App jedoch bereits in Amplify vorhanden ist oder mit Amplify bereitgestellt wird AWS CloudFormation, müssen Sie die Umgebungsvariable im Abschnitt **`AMPLIFY_MONOREPO_APP_ROOT`** Umgebungsvariablen in der Amplify-Konsole manuell festlegen.

Automatisches Einstellen der Umgebungsvariablen AMPLIFY_MONOREPO_APP_ROOT während der Bereitstellung

Die folgenden Anweisungen zeigen, wie Sie eine Monorepo-App mit der Amplify-Konsole bereitstellen. Amplify legt die `AMPLIFY_MONOREPO_APP_ROOT` Umgebungsvariable automatisch mithilfe des Stammordners der App fest, den Sie in der Konsole angeben.

Um eine Monorepo-App mit der Amplify-Konsole bereitzustellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie in der oberen rechten Ecke Neue App erstellen.
3. Wählen Sie auf der Seite „Mit Amplify erstellen“ Ihren Git-Anbieter und dann Weiter aus.
4. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:

    a. Wählen Sie den Namen Ihres Repositorys aus der Liste aus.

b. Wählen Sie den Namen des Branches, den Sie verwenden möchten.

c. Wählen Sie Meine App ist ein Monorepo

d. Geben Sie den Pfad zu Ihrer App in Ihrem Monorepo ein, zum Beispiel. **`apps/app1`**

e. Wählen Sie Weiter aus.

5. Auf der Seite mit den App-Einstellungen können Sie die Standardeinstellungen verwenden oder die Build-Einstellungen für Ihre App anpassen. Im Abschnitt Umgebungsvariablen setzt Amplify `AMPLIFY_MONOREPO_APP_ROOT` auf den Pfad, den Sie in Schritt 4d angegeben haben.
6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite „Überprüfen“ die Option Speichern und bereitstellen.

Einstellung der Umgebungsvariablen AMPLIFY_MONOREPO_APP_ROOT für eine bestehende App

Verwenden Sie die folgenden Anweisungen, um die `AMPLIFY_MONOREPO_APP_ROOT` Umgebungsvariable für eine App manuell festzulegen, die bereits auf Amplify bereitgestellt wurde oder mit CloudFormation der erstellt wurde.

Um die Umgebungsvariable AMPLIFY_MONOREPO_APP_ROOT für eine bestehende App festzulegen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie den Namen der App, für die Sie die Umgebungsvariable festlegen möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Umgebungsvariablen aus.
4. Wählen Sie auf der Seite Umgebungsvariablen die Option Variablen verwalten aus.
5. Gehen Sie im Abschnitt Variablen verwalten wie folgt vor:

   a. Wählen Sie Add new (Neuen hinzufügen) aus.

   b. Geben Sie für Variable den Schlüssel ein`AMPLIFY_MONOREPO_APP_ROOT`.

   c. Geben Sie unter Value beispielsweise den Pfad zur App ein**`apps/app1`**.

   d. Für Branch wendet Amplify standardmäßig die Umgebungsvariable auf alle Branches an.

6. Wählen Sie Speichern.

Konfiguration der Turborepo- und pnpm-Monorepo-Apps

Die Monorepo-Build-Tools Turborepo und pnpm Workspace rufen Konfigurationsinformationen aus Dateien ab. `.npmrc` Wenn Sie eine Monorepo-App bereitstellen, die mit einem dieser Tools erstellt wurde, benötigen Sie eine `.npmrc` Datei in Ihrem Projekt-Stammverzeichnis.

Stellen Sie in der `.npmrc` Datei den Linker für die Installation von Node-Paketen auf ein. `hoisted` Sie können die folgende Zeile in Ihre Datei kopieren.

```
node-linker=hoisted
```

Weitere Informationen zu *`.npmrc`* Dateien und Einstellungen finden Sie unter pnpm .npmrc in der pnpm-Dokumentation.

Pnpm ist nicht im Standard-Build-Container von Amplify enthalten. Für pnpm workspace- und Turborepo-Apps müssen Sie in der `preBuild` Phase der Build-Einstellungen Ihrer App einen Befehl zur Installation von pnpm hinzufügen.

Der folgende Beispielauszug aus einer Build-Spezifikation zeigt eine `preBuild` Phase mit einem Befehl zur Installation von pnpm.

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

Anpassen des Build-Images

Sie können ein benutzerdefiniertes Build-Image verwenden, um eine benutzerdefinierte Build-Umgebung für eine Amplify-App bereitzustellen. Wenn Sie bestimmte Abhängigkeiten haben, deren Installation während eines Builds mithilfe des Standardcontainers von Amplify lange dauert, können Sie Ihr eigenes Docker-Image erstellen und während eines Builds darauf verweisen. Bilder können auf Amazon Elastic Container Registry Public gehostet werden.

Damit ein benutzerdefiniertes Build-Image als Amplify-Build-Image funktioniert, muss es die folgenden Anforderungen erfüllen.

Anforderungen an ein benutzerdefiniertes Build-Image

1. Eine Linux-Distribution, die die GNU C-Bibliothek (Glibc) unterstützt, z. B. Amazon Linux, die für die x86-64-Architektur kompiliert wurde.
2. cURL: Wenn wir Ihr benutzerdefiniertes Image starten, laden wir unseren Build Runner in Ihrem Container herunter, weshalb cURL vorhanden sein muss. Wenn diese Abhängigkeit fehlt, schlägt der Build sofort ohne Ausgabe fehl, da unser Build-Runner keine Ausgabe erzeugen kann.
3. Git: Zum Klonen des Git-Repositorys muss Git im Image installiert sein. Wenn diese Abhängigkeit fehlt, schlägt der Schritt zum Klonen des Repositorys fehl.
4. OpenSSH: Um Ihr Repository sicher zu klonen, benötigen wir OpenSSH, um den SSH-Schlüssel während des Builds vorübergehend einzurichten. Das OpenSSH-Paket stellt die Befehle bereit, die der Build-Runner dafür benötigt.
5. Bash und The Bourne Shell: Diese beiden Hilfsprogramme werden verwendet, um Befehle während der Build-Zeit auszuführen. Wenn sie nicht installiert sind, schlagen Ihre Builds möglicherweise vor dem Start fehl.
6. Node.js+npm: Unser Build-Runner installiert Node nicht. Stattdessen ist es darauf angewiesen, dass Node und NPM im Image installiert sind. Dies ist nur für Builds erforderlich, die NPM-Pakete oder Node-spezifische Befehle erfordern. Wir empfehlen jedoch dringend, sie zu installieren, da der Amplify Build Runner diese Tools verwenden kann, um die Build-Ausführung zu verbessern, wenn sie vorhanden sind. Die Paket-Override-Funktion von Amplify verwendet NPM, um das Hugo-Extended-Paket zu installieren, wenn Sie eine Überschreibung für Hugo festlegen.

Die folgenden Pakete sind nicht erforderlich, wir empfehlen Ihnen jedoch dringend, sie zu installieren.

1. NVM (Node Version Manager): Wir empfehlen Ihnen, diesen Versionsmanager zu installieren, wenn Sie mit verschiedenen Versionen von umgehen müssenNode. Wenn Sie eine Überschreibung festlegen, werden mit der Paket-Override-Funktion von Amplify NVM die Versionen von Node.js vor jedem Build geändert.
2. Wget: Amplify kann das Wget Hilfsprogramm verwenden, um Dateien während des Build-Prozesses herunterzuladen. Wir empfehlen, dass Sie es in Ihrem benutzerdefinierten Image installieren.
3. Tar: Amplify kann das Tar Hilfsprogramm verwenden, um heruntergeladene Dateien während des Build-Prozesses zu dekomprimieren. Wir empfehlen, dass Sie es in Ihrem benutzerdefinierten Image installieren.

Konfiguration eines benutzerdefinierten Build-Images für eine App

Gehen Sie wie folgt vor, um ein benutzerdefiniertes Build-Image für eine Anwendung in der Amplify-Konsole zu konfigurieren.

So konfigurieren Sie ein in Amazon ECR gehostetes benutzerdefiniertes Build-Image

1. Informationen zum Einrichten eines öffentlichen Amazon ECR-Repositorys mit einem Docker-Image finden Sie unter Erste Schritte im Amazon ECR Public User Guide.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
3. Wählen Sie die App aus, für die Sie ein benutzerdefiniertes Build-Image konfigurieren möchten.
4. Wählen Sie im Navigationsbereich Hosting, Build-Einstellungen aus.
5. Wählen Sie auf der Seite mit den Build-Einstellungen im Abschnitt Build-Image-Einstellungen die Option Bearbeiten aus.
6. Erweitern Sie auf der Seite Build-Image-Einstellungen bearbeiten das Menü Build-Image und wählen Sie Benutzerdefiniertes Build-Image aus.
7. Geben Sie den Namen des Amazon ECR Public Repos ein, das Sie in Schritt eins erstellt haben. Hier wird Ihr Build-Image gehostet. Wenn der Name Ihres Repos beispielsweise ecr-examplerepo lautet, würden Sie Folgendes eingeben. **`public.ecr.aws/xxxxxxxx/ecr-examplerepo`**
8. Wählen Sie Speichern.

Verwenden bestimmter Paket- und Abhängigkeitsversionen im Build-Image

Live-Paket-Updates ermöglichen es Ihnen, die Versionen von Paketen und Abhängigkeiten anzugeben, die im Amplify-Standard-Build-Image verwendet werden sollen. Das Standard-Build-Image enthält mehrere vorinstallierte Pakete und Abhängigkeiten (z. B. Hugo, Amplify CLI, Yarn usw.). Bei Live-Paket-Updates können Sie die Version dieser Abhängigkeiten überschreiben und entweder eine bestimmte Version angeben oder sicherstellen, dass immer die neueste Version installiert ist.

Wenn Live-Paket-Updates aktiviert sind, aktualisiert der Build Runner vor der Ausführung Ihres Builds zunächst die angegebenen Abhängigkeiten (oder führt ein Downgrade durch). Dadurch erhöht sich die Erstellungszeit proportional zu der Zeit, die für die Aktualisierung der Abhängigkeiten benötigt wird. Der Vorteil besteht jedoch darin, dass Sie sicherstellen können, dass dieselbe Version einer Abhängigkeit zum Erstellen Ihrer App verwendet wird.

> **Warning**
>
> Wenn Sie die Version Node.js auf die neueste Version setzen, schlagen Builds fehl. Stattdessen müssen Sie eine exakte Version von Node.js angeben, z. B. `18`21.`5`, oder`v0.1.2`.

Um Live-Paket-Updates zu konfigurieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie Live-Paket-Updates konfigurieren möchten.
3. Wählen Sie im Navigationsbereich Hosting, Build-Einstellungen aus.
4. Wählen Sie auf der Seite mit den Build-Einstellungen im Abschnitt Build-Image-Einstellungen die Option Bearbeiten aus.
5. Wählen Sie auf der Seite Build-Image-Einstellungen bearbeiten in der Liste Live-Paket-Updates die Option Neu hinzufügen aus.
6. Wählen Sie unter Package die Abhängigkeit aus, die Sie überschreiben möchten.
7. Behalten Sie für Version entweder die neueste Standardversion bei oder geben Sie eine bestimmte Version der Abhängigkeit ein. Wenn Sie Latest verwenden, wird die Abhängigkeit immer auf die neueste verfügbare Version aktualisiert.
8. Wählen Sie Speichern.

Konfiguration der Build-Instanz für eine Amplify-Anwendung

Amplify Hosting bietet konfigurierbare Build-Instance-Größen, mit denen Sie die Build-Instance Ihrer Anwendung mit den CPU-, Arbeitsspeicher- und Festplattenspeicherressourcen versorgen können, die sie benötigt. Vor der Veröffentlichung dieser Funktion stellte Amplify eine Build-Instance-Konfiguration mit fester Größe von 8 GiB Speicher und 4 V CPUs bereit.

Amplify unterstützt drei Build-Instance-Typen: StandardLarge, undXLarge. Wenn Sie keinen Instanztyp angeben, verwendet Amplify die Standard Standardinstanz. Sie können den Build-Instance-Typ für eine Anwendung mithilfe der Amplify-Konsole AWS CLI, der oder der SDKs konfigurieren.

Die Kosten für jeden Build-Instance-Typ werden pro Build-Minute berechnet. Details zu den Preisen finden Sie unter AWS Amplify -Preise.

In der folgenden Tabelle werden die Rechenspezifikationen für jeden Build-Instance-Typ beschrieben:

Instanztyp erstellen	v CPUs	Arbeitsspeicher	Festplattenkapazität
Standard	4 v CPUs	8 GiB	128 GB
Large	8 V CPUs	16 GiB	128 GB
XLarge	36 v CPUs	72 GiB	256 GB

Topics

Grundlegendes zu Build-Instance-Typen

Die Einstellung für den Build-Instanztyp wird auf Anwendungsebene konfiguriert und erstreckt sich auf alle Zweige der Anwendung. Die folgenden wichtigen Details gelten für Build-Instance-Typen:

- Der Build-Instance-Typ, den Sie für eine Anwendung konfigurieren, gilt automatisch für automatisch erstellte Branches und Pull-Request-Vorschauen.
- Das Service-Kontingent für gleichzeitige Jobs gilt für alle Build-Instance-Typen in Ihrem. AWS-Konto Wenn Ihr Limit für gleichzeitige Jobs beispielsweise fünf beträgt, können Sie bis zu 5 Builds für alle Instance-Typen in Ihrem ausführen. AWS-Konto
- Die Kosten für jeden Build-Instance-Typ werden pro Build-Minute berechnet. Der Zuweisungsprozess für die Build-Instance kann zusätzlichen Zeitaufwand erfordern, bevor Ihr Build gestartet wird. Insbesondere XLarge bei größeren Instances kann es aufgrund dieser Overhead-Zeit bei Ihrem Build zu Latenz kommen, bevor der Build gestartet wird. Ihnen wird jedoch nur die tatsächliche Build-Zeit in Rechnung gestellt, nicht die Overhead-Zeit.

Sie können den Build-Instance-Typ konfigurieren, wenn Sie eine neue Anwendung erstellen, oder Sie können den Instance-Typ in einer vorhandenen Anwendung aktualisieren. Anweisungen zur Konfiguration dieser Einstellung in der Amplify-Konsole finden Sie unterKonfiguration des

Build-Instance-Typs in der Amplify-Konsole. Sie können diese Einstellung auch mit dem SDKs aktualisieren. Weitere Informationen finden Sie unter und UpdateApp APIs in der CreateAppAmplify API-Referenz.

Wenn Sie bereits Anwendungen in Ihrem Konto haben, die vor der Veröffentlichung der Funktion für den anpassbaren Build-Instanztyp erstellt wurden, verwenden diese den Standard Standard-Instanztyp. Wenn Sie den Build-Instance-Typ für eine bestehende Anwendung aktualisieren, verwenden alle Builds, die sich vor dem Update in der Warteschlange befinden oder in Bearbeitung sind, den zuvor konfigurierten Build-Instanztyp. Wenn Sie beispielsweise eine bestehende Anwendung haben, bei der der `main` Branch auf Amplify bereitgestellt ist, und Sie ihren Build-Instance-Typ von Standard auf Large aktualisieren, verwenden alle neuen Builds, die Sie von der `main` Filiale aus initiieren, den Build-Instance-Typ Large. Alle Builds, die zum Zeitpunkt der Aktualisierung des Build-Instance-Typs in Bearbeitung sind, werden jedoch weiterhin auf der Standard-Instance ausgeführt.

Konfiguration des Build-Instance-Typs in der Amplify-Konsole

Verwenden Sie das folgende Verfahren, um den Build-Instance-Typ zu konfigurieren, wenn Sie eine neue Amplify-Anwendung erstellen.

Um den Build-Instance-Typ für eine neue Anwendung zu konfigurieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite Alle Apps die Option Neue App erstellen aus.
3. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.
4. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:
    a. Wählen Sie in der Liste „Kürzlich aktualisierte Repositorys“ den Namen des Repositorys aus, mit dem Sie eine Verbindung herstellen möchten.
    b. Wählen Sie in der Branch-Liste den Namen des Repository-Branches aus, zu dem Sie eine Verbindung herstellen möchten.
    c. Wählen Sie Weiter aus.
5. Öffnen Sie auf der Seite mit den App-Einstellungen den Abschnitt Erweiterte Einstellungen.
6. Wählen Sie unter Instanztyp Build den gewünschten Instanztyp aus der Liste aus.
7. Wenn Sie eine auf der Laufzeit basierende Anwendung Node.js bereitstellen, konfigurieren Sie die Größe des Heap-Speichers so, dass ein großer Instance-Typ effektiv genutzt wird.

Sie können dies auf der Seite mit den App-Einstellungen tun, indem Sie entweder eine Umgebungsvariable festlegen oder die Build-Einstellungen aktualisieren. Weitere Informationen finden Sie unter Konfiguration des Heap-Speichers einer Anwendung für die Verwendung großer Instance-Typen.

- Legen Sie eine Umgebungsvariable fest

  a. Wählen Sie im Bereich Erweiterte Einstellungen im Abschnitt Umgebungsvariablen die Option Neu hinzufügen aus.

  b. Geben Sie als Schlüssel ein**`NODE_OPTIONS`**.

  c. Geben Sie für Wert `--max-old-space-size=`*`memory_size_in_mb`* ein. *`memory_size_in_mb`*Ersetzen Sie es durch die gewünschte Heap-Speichergröße in Megabyte.

- Aktualisieren Sie die Build-Einstellungen

  a. Wählen Sie im Abschnitt Build-Einstellungen die Option YML-Datei bearbeiten aus.

  b. Fügen Sie der `preBuild` Phase den folgenden Befehl hinzu. *`memory_size_in_mb`*Ersetzen Sie es durch die gewünschte Heap-Speichergröße in Megabyte.

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

  c. Wählen Sie Speichern.

8. Wählen Sie „Weiter“.
9. Wählen Sie auf der Seite Überprüfen die Option Speichern und bereitstellen aus.

Verwenden Sie das folgende Verfahren, um den Build-Instance-Typ für eine bestehende Amplify-Anwendung zu konfigurieren.

Um den Build-Instance-Typ für eine bestehende Anwendung zu konfigurieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie den Build-Instance-Typ konfigurieren möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Build-Einstellungen aus.
4. Wählen Sie auf der Seite mit den Build-Einstellungen im Abschnitt Erweiterte Einstellungen die Option Bearbeiten aus.

5. Wählen Sie auf der Seite Einstellungen bearbeiten für Build-Instance-Typ den gewünschten Instance-Typ aus der Liste aus.
6. Wählen Sie Speichern. Diese Änderung wird wirksam, wenn Sie die Anwendung das nächste Mal bereitstellen.
7. (Optional) Gehen Sie wie folgt vor, um die aktualisierte Anwendung sofort bereitzustellen:
   a. Wählen Sie im Navigationsbereich Overview (Übersicht) aus.
   b. Wählen Sie auf der Übersichtsseite Ihrer Anwendung den Zweig aus, den Sie erneut bereitstellen möchten.
   c. Wählen Sie auf der Seite Bereitstellung eine Bereitstellung aus, z. B. die neueste Bereitstellung. Wählen Sie dann Diese Version erneut bereitstellen aus. Eine neue Bereitstellung wird beginnen.
   d. Wenn die Bereitstellung abgeschlossen ist, zeigen die Build-Einstellungen der Anwendung, dass der Branch den aktualisierten Build-Instance-Typ verwendet.

Konfiguration des Heap-Speichers einer Anwendung für die Verwendung großer Instance-Typen

Wenn Sie speicherintensive Anwendungen erstellen, erfahren Sie in diesem Abschnitt, wie Sie Ihre Anwendung für die Verwendung großer Instance-Typen konfigurieren. Programmiersprachen und Frameworks verlassen sich häufig auf die Zuweisung von dynamischem Speicher, auch bekannt als Heap-Speicher, während der Laufzeit, um die Speicheranforderungen von Anwendungen zu verwalten. Heap-Speicher wird von der Laufzeitumgebung angefordert und vom Host-Betriebssystem zugewiesen. Standardmäßig erzwingen Laufzeitumgebungen eine maximale Heap-Größenbeschränkung, die der Anwendung zur Verfügung steht. Das bedeutet, dass für die Anwendung über die Heap-Größe hinaus kein zusätzlicher Speicher zur Verfügung steht, auch wenn das Host-Betriebssystem oder der Container über eine größere Speichermenge verfügt.

Beispielsweise erzwingt die Laufzeitumgebung JavaScript Node.js v8 eine standardmäßige Heap-Größenbeschränkung, die von mehreren Faktoren abhängt, einschließlich der Host-Speichergröße. Infolgedessen haben Large Build-Instanzen eine Standard-Heap-Größe von Node.js von 2096 MB und die XLarge Instanz hat eine Standard-Heap-Größe von 4144 MB. Standard Daher führt das Erstellen einer Anwendung mit einem Speicherbedarf von 6000 MB unter Verwendung der Standard-Heap-Größe von Node.js auf einem beliebigen Amplify-Build-Instance-Typ zu einem fehlgeschlagenen Build aufgrund eines out-of-memory Fehlers.

Um die standardmäßigen Speicherbeschränkungen für die Heap-Größe von Node.js zu umgehen, können Sie einen der folgenden Schritte ausführen:

- Setzen Sie die `NODE_OPTIONS` Umgebungsvariable in Ihrer Amplify-Anwendung auf den Wert `--max-old-space-size=`*`memory_size_in_mb`*. Geben Sie für `memory_size_in_mb` die gewünschte Heap-Speichergröße in Megabyte an.

  Detaillierte Anweisungen finden Sie unter Umgebungsvariablen setzen.

- Fügen Sie der `preBuild` Phase in der Build-Spezifikation Ihrer Amplify-Anwendung den folgenden Befehl hinzu.

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

  Sie können die Build-Spezifikation in der Amplify-Konsole oder in der `amplify.yml` Datei Ihrer Anwendung in Ihrem Projekt-Repository aktualisieren. Detaillierte Anweisungen finden Sie unter Konfiguration der Build-Einstellungen für eine Amplify-Anwendung.

  Die folgende Amplify-Build-Spezifikation legt die Heap-Speichergröße von Node.js auf 7000 MB fest, um eine React-Frontend-Anwendung zu erstellen:

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

```
      - .npm/**/*
```

Um große Instance-Typen effektiv nutzen zu können, ist es wichtig, dass eine ausreichende Heap-Speichergröße konfiguriert ist. Die Konfiguration einer kleinen Heap-Größe für eine speicherintensive Anwendung führt wahrscheinlich zu einem Build-Fehler. Die Build-Logs der Anwendung weisen möglicherweise nicht direkt auf einen out-of-memory Fehler hin, da die Laufzeit der Anwendung unerwartet abstürzen kann. Die Konfiguration einer Heap-Größe, die so groß ist wie der Host-Speicher, kann dazu führen, dass das Host-Betriebssystem andere Prozesse auslagert oder beendet, was möglicherweise Ihren Build-Prozess unterbricht. Als Referenz empfiehlt Node.js, auf einem Computer mit etwa 2000 MB Arbeitsspeicher eine maximale Heap-Größe von 1536 MB festzulegen, damit ein Teil des Speichers für andere Zwecke zur Verfügung steht.

Die optimale Heap-Größe hängt von den Anforderungen Ihrer Anwendung und der Ressourcennutzung ab. Wenn Sie auf out-of-memory Fehler stoßen, beginnen Sie mit einer moderaten Heap-Größe und erhöhen Sie sie dann schrittweise nach Bedarf. Als Richtlinie empfehlen wir, mit 6000 MB für einen Standard Instance-Typ, 12000 MB für einen Large Instance-Typ und 60000 MB für einen XLarge Instance-Typ zu beginnen.

Einen eingehenden Webhook erstellen, um einen Build zu starten

Richten Sie in der Amplify-Konsole einen eingehenden Webhook ein, um einen Build zu starten, ohne Code in Ihr Git-Repository zu übertragen. Sie können Webhooks mit Headless-CMS-Tools (wie Contentful oder GraphCMS) verwenden, um einen Build zu starten, wenn sich der Inhalt ändert, oder um tägliche Builds mit Diensten wie Zapier durchzuführen.

Um einen eingehenden Webhook zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie einen Webhook erstellen möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Build-Einstellungen aus.
4. Scrollen Sie auf der Seite mit den Build-Einstellungen nach unten zum Abschnitt Eingehende Webhooks und wählen Sie Webhook erstellen aus.
5. Gehen Sie im Dialogfeld „Webhook erstellen“ wie folgt vor:

    a. Geben Sie unter Webhook-Name einen Namen für den Webhook ein.

b. Wählen Sie für Branch to build den Branch aus, der auf eingehenden Webhook-Anfragen aufgebaut werden soll.

c. Wählen Sie Webhook erstellen.

6. Führen Sie im Abschnitt Eingehende Webhooks einen der folgenden Schritte aus:

   - Kopieren Sie die Webhook-URL und stellen Sie sie einem Headless-CMS-Tool oder einem anderen Dienst zur Verfügung, um Builds zu initiieren.
   - Führen Sie den Befehl curl in einem Terminalfenster aus, um einen neuen Build zu starten.

E-Mail-Benachrichtigungen für Builds einrichten

Sie können E-Mail-Benachrichtigungen für eine AWS Amplify App einrichten, um Stakeholder oder Teammitglieder zu benachrichtigen, wenn ein Build erfolgreich ist oder fehlschlägt. Amplify Hosting erstellt ein Amazon Simple Notification Service (SNS) -Thema in Ihrem Konto und verwendet es zur Konfiguration von E-Mail-Benachrichtigungen. Benachrichtigungen können so konfiguriert werden, dass sie für alle Zweige oder bestimmte Zweige einer Amplify-App gelten.

E-Mail-Benachrichtigungen einrichten

Verwenden Sie die folgenden Verfahren, um E-Mail-Benachrichtigungen für alle Filialen oder bestimmte Zweige einer Amplify-App einzurichten.

So richten Sie E-Mail-Benachrichtigungen für eine Amplify-App ein

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie E-Mail-Benachrichtigungen einrichten möchten.
3. Wählen Sie im Navigationsbereich Hosting, Benachrichtigungen erstellen aus. Wählen Sie auf der Seite Benachrichtigungen erstellen die Option Benachrichtigungen verwalten aus.
4. Wählen Sie auf der Seite Benachrichtigungen verwalten die Option Neue hinzufügen aus.
5. Führen Sie eine der folgenden Aktionen aus:

   - Um Benachrichtigungen für eine einzelne Filiale zu senden, geben Sie unter E-Mail die E-Mail-Adresse ein, an die Benachrichtigungen gesendet werden sollen. Wählen Sie unter Filiale den Namen der Filiale aus, für die Benachrichtigungen gesendet werden sollen.
   - Um Benachrichtigungen für alle verbundenen Filialen zu senden, geben Sie unter E-Mail die E-Mail-Adresse ein, an die Benachrichtigungen gesendet werden sollen. Wählen Sie für Filiale die Option Alle Filialen aus.

6. Wählen Sie Speichern.

Eine benutzerdefinierte Domain verbinden

Sie können eine App, die Sie mit Amplify Hosting bereitgestellt haben, mit einer benutzerdefinierten Domain verbinden. Wenn Sie Amplify verwenden, um Ihre Web-App bereitzustellen, hostet Amplify sie für Sie auf der `amplifyapp.com` Standarddomain mit einer URL wie. `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com` Wenn Sie Ihre App mit einer benutzerdefinierten Domain verbinden, sehen Benutzer, dass Ihre App auf einer benutzerdefinierten URL gehostet wird, z. `https://www.example.com`

Sie können eine benutzerdefinierte Domain über einen akkreditierten Domain-Registrar wie Amazon Route 53 oder GoDaddy erwerben. Route 53 ist der Domain Name System (DNS) -Webservice von Amazon. Weitere Informationen zur Verwendung von Route 53 finden Sie unter Was ist Amazon Route 53 Eine Liste der akkreditierten Domain-Registrare von Drittanbietern finden Sie im Verzeichnis der akkreditierten Registrare auf der ICANN-Website.

Wenn Sie Ihre benutzerdefinierte Domain einrichten, können Sie das verwaltete Standardzertifikat verwenden, das Amplify für Sie bereitstellt, oder Sie können Ihr eigenes benutzerdefiniertes Zertifikat verwenden. Sie können das für die Domain verwendete Zertifikat jederzeit ändern. Ausführliche Informationen zur Verwaltung von Zertifikaten finden Sie unterVerwenden von Zertifikaten SSL/TLS.

Bevor Sie mit der Einrichtung einer benutzerdefinierten Domäne fortfahren, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- Sie besitzen einen registrierten Domainnamen.
- Sie haben ein Zertifikat, das von ausgestellt oder importiert wurde AWS Certificate Manager.
- Sie haben Ihre App auf Amplify Hosting bereitgestellt.

  Weitere Informationen zum Ausführen dieses Schritts finden Sie unterErste Schritte mit der Bereitstellung einer App auf Amplify Hosting.
- Sie verfügen über Grundkenntnisse in Domänen und DNS-Terminologie.

  Weitere Informationen zu Domänen und DNS finden Sie unterVerstehen der DNS-Terminologie und -Konzepte.

> **Warning**
>
> Wenn DomainAssociation Sie eine Anfrage für eine Amplify-App mit einer Domain initiieren, die bereits oder zuvor mit verschiedenen Amplify-Apps in anderen AWS-Konten in derselben Region verknüpft ist oder war, wird dies als kontoübergreifende Domain-Zuordnung betrachtet. Anfragen zur kontoübergreifenden Domainzuweisung erfordern eine manuelle Überprüfung. Wenn Sie mit einer kontoübergreifenden Domain-Zuordnung fortfahren möchten, wenden Sie sich bitte an den AWS-Support, um Unterstützung zu erhalten.

Themen

Verstehen der DNS-Terminologie und -Konzepte

Wenn Sie mit den Begriffen und Konzepten im Zusammenhang mit Domain Name System (DNS) nicht vertraut sind, können Ihnen die folgenden Themen helfen, die Verfahren zum Hinzufügen benutzerdefinierter Domänen zu verstehen.

DNS-Terminologie

Im Folgenden finden Sie eine Liste von Begriffen, die im DNS gebräuchlich sind. Sie können Ihnen helfen, die Verfahren zum Hinzufügen benutzerdefinierter Domänen zu verstehen.

CNAME

Ein Canonical Record Name (CNAME) ist eine Art von DNS-Eintrag, der die Domain für eine Reihe von Webseiten maskiert und sie so aussehen lässt, als ob sie sich woanders befinden würden. Ein CNAME verweist eine Subdomain auf einen vollqualifizierten Domainnamen (FQDN). Sie können beispielsweise einen neuen CNAME-Eintrag erstellen, um die Subdomain www.example.com, wobei www die Subdomain ist, der FQDN-Domain branch-name.d1m7bkiki6tdw1.cloudfront.net zuzuordnen, die Ihrer App in der Amplify-Konsole zugewiesen wurde.

ANAME

Ein ANAME-Record ist wie ein CNAME-Record, allerdings auf Stammebene. Ein ANAME verweist das Stammverzeichnis Ihrer Domain auf einen FQDN. Dieser FQDN verweist auf eine IP-Adresse.

Namenserver

Ein Nameserver ist ein Server im Internet, der darauf spezialisiert ist, Anfragen zum Standort der verschiedenen Dienste eines Domainnamens zu bearbeiten. Wenn Sie Ihre Domain in Amazon Route 53 einrichten, ist Ihrer Domain bereits eine Liste von Nameservern zugewiesen.

NS-Eintrag

Ein NS-Eintrag verweist auf Nameserver, die Ihre Domaindetails nachschlagen.

DNS-Überprüfung

Ein Domain Name System (DNS) ist wie ein Telefonbuch, das menschenlesbare Domainnamen in computerfreundliche IP-Adressen übersetzt. Wenn Sie etwas **`https://google.com`** in einen Browser eingeben, wird im DNS-Anbieter ein Suchvorgang ausgeführt, um die IP-Adresse des Servers zu ermitteln, der die Website hostet.

DNS-Anbieter verfügen über die Datensätze der Domänen und die zugehörigen IP-Adressen. Die am häufigsten verwendeten DNS-Einträge sind CNAME-, ANAME- und NS-Einträge.

Amplify verwendet einen CNAME-Eintrag, um zu überprüfen, ob Sie Eigentümer Ihrer benutzerdefinierten Domain sind. Wenn Sie Ihre Domain bei Route 53 hosten, erfolgt die Überprüfung automatisch in Ihrem Namen. Wenn Sie Ihre Domain jedoch bei einem Drittanbieter hosten GoDaddy, müssen Sie die DNS-Einstellungen Ihrer Domain manuell aktualisieren und einen neuen CNAME-Eintrag hinzufügen, der von Amplify bereitgestellt wird.

Benutzerdefinierter Domain-Aktivierungsprozess

> **Warning**
>
> Wenn DomainAssociation Sie eine Anfrage für eine Amplify-App mit einer Domain initiieren, die bereits oder zuvor mit verschiedenen Amplify-Apps in anderen AWS-Konten in derselben Region verknüpft ist oder war, wird dies als kontoübergreifende Domain-Zuordnung betrachtet. Anfragen zur kontoübergreifenden Domainzuweisung erfordern eine manuelle Überprüfung. Wenn Sie mit einer kontoübergreifenden Domain-Zuordnung fortfahren möchten, wenden Sie sich bitte an den AWS-Support, um Unterstützung zu erhalten.

Wenn Sie Ihre Amplify-App mit einer benutzerdefinierten Domain in der Amplify-Konsole verbinden, muss Amplify mehrere Schritte ausführen, bevor Sie Ihre App mit Ihrer benutzerdefinierten Domain anzeigen können. In der folgenden Liste werden die einzelnen Schritte bei der Einrichtung und Aktivierung der Domain beschrieben.

SSL/TLS-Erstellung

Wenn Sie ein verwaltetes Zertifikat verwenden, AWS Amplify stellt es ein SSL/TLS-Zertifikat für die Einrichtung einer sicheren benutzerdefinierten Domain aus.

SSL/TLS-Konfiguration und -Überprüfung

Vor der Ausstellung eines verwalteten Zertifikats überprüft Amplify, ob Sie der Eigentümer der Domain sind. Für Domains, die von Amazon Route 53 verwaltet werden, aktualisiert Amplify automatisch den DNS-Bestätigungseintrag. Für Domains, die außerhalb von Route 53 verwaltet werden, müssen Sie den in der Amplify-Konsole bereitgestellten DNS-Bestätigungseintrag manuell zu Ihrer Domain bei einem DNS-Drittanbieter hinzufügen.

Wenn Sie ein benutzerdefiniertes Zertifikat verwenden, sind Sie für die Überprüfung des Domainbesitzes verantwortlich.

Aktivierung der Domain

Die Domain wurde erfolgreich verifiziert. Für Domains, die außerhalb von Route 53 verwaltet werden, müssen Sie die in der Amplify-Konsole bereitgestellten CNAME-Einträge manuell zu Ihrer Domain bei einem DNS-Drittanbieter hinzufügen.

Verwenden von Zertifikaten SSL/TLS

Ein SSL/TLS Zertifikat ist ein digitales Dokument, das es Webbrowsern ermöglicht, mithilfe des sicheren SSL/TLS Protokolls verschlüsselte Netzwerkverbindungen zu Websites zu identifizieren und herzustellen. Wenn Sie Ihre benutzerdefinierte Domain einrichten, können Sie das verwaltete Standardzertifikat verwenden, das Amplify für Sie bereitstellt, oder Sie können Ihr eigenes benutzerdefiniertes Zertifikat verwenden.

Mit einem verwalteten Zertifikat stellt Amplify ein SSL/TLS Zertifikat für alle mit Ihrer App verbundenen Domains aus, sodass der gesamte Datenverkehr über HTTPS/2 gesichert ist. Das von AWS Certificate Manager (ACM) generierte Standardzertifikat ist 13 Monate gültig und verlängert sich automatisch, solange Ihre App bei Amplify gehostet wird.

> **Warning**
>
> Amplify kann das Zertifikat nicht verlängern, wenn der CNAME-Bestätigungseintrag in den DNS-Einstellungen Ihres Domain-Anbieters geändert oder gelöscht wurde. Sie müssen die Domain in der Amplify-Konsole löschen und erneut hinzufügen.

Um ein benutzerdefiniertes Zertifikat verwenden zu können, müssen Sie zunächst ein Zertifikat von einer Drittanbieter-Zertifizierungsstelle Ihrer Wahl erwerben. Amplify Hosting unterstützt zwei Arten von Zertifikaten: RSA (Rivest-Shamir-Adleman) und ECDSA (Elliptic Curve Digital Signature Algorithm). Jeder Zertifikatstyp muss den folgenden Anforderungen entsprechen.

RSA-Zertifikate

- Amplify Hosting unterstützt 1024-Bit-, 2048-Bit-, 3072-Bit- und 4096-Bit-RSA-Schlüssel.
- AWS Certificate Manager (ACM) stellt RSA-Zertifikate mit bis zu 2048-Bit-Schlüsseln aus.
- Um ein 3072-Bit- oder 4096-Bit-RSA-Zertifikat zu verwenden, beziehen Sie das Zertifikat extern und importieren Sie es in ACM. Es wird dann für die Verwendung mit Amplify Hosting verfügbar sein.

ECDSA-Zertifikate

- Amplify Hosting unterstützt 256-Bit-Schlüssel.
- Verwenden Sie die elliptische Prime256v1-Kurve, um ein ECDSA-Zertifikat für Amplify Hosting zu erhalten.

Nachdem Sie ein Zertifikat erhalten haben, importieren Sie es in. AWS Certificate Manager ACM ist ein Dienst, mit dem Sie auf einfache Weise öffentliche und private SSL/TLS Zertifikate für die Verwendung mit und Ihre internen verbundenen Ressourcen bereitstellen, verwalten AWS-Services und bereitstellen können. Stellen Sie sicher, dass Sie das Zertifikat in der Region USA Ost (Nord-Virginia) (us-east-1) anfordern oder importieren.

Stellen Sie sicher, dass Ihr benutzerdefiniertes Zertifikat alle Subdomains abdeckt, die Sie hinzufügen möchten. Sie können einen Platzhalter am Anfang Ihres Domainnamens verwenden, um mehrere Subdomains abzudecken. Wenn Ihre Domain beispielsweise ist`example.com`, können Sie die Wildcard-Domain einbeziehen. `*.example.com` Dies gilt für Subdomains wie `product.example.com` und. `api.example.com`

Sobald Ihr benutzerdefiniertes Zertifikat in ACM verfügbar ist, können Sie es bei der Einrichtung der Domain auswählen. Anweisungen zum Importieren von Zertifikaten in AWS Certificate Manager finden Sie unter Zertifikate importieren AWS Certificate Manager in im AWS Certificate Manager Benutzerhandbuch.

Wenn Sie Ihr benutzerdefiniertes Zertifikat in ACM erneuern oder erneut importieren, aktualisiert Amplify die mit Ihrer benutzerdefinierten Domain verknüpften Zertifikatsdaten. Bei importierten Zertifikaten verwaltet ACM die Verlängerungen nicht automatisch. Sie sind dafür verantwortlich, Ihre benutzerdefinierten Zertifikate zu erneuern und erneut zu importieren.

Sie können das für eine Domain verwendete Zertifikat jederzeit ändern. Sie können beispielsweise vom verwalteten Standardzertifikat zu einem benutzerdefinierten Zertifikat oder von einem benutzerdefinierten Zertifikat zu einem verwalteten Zertifikat wechseln. Darüber hinaus können Sie das verwendete benutzerdefinierte Zertifikat in ein anderes benutzerdefiniertes Zertifikat ändern. Anweisungen zum Aktualisieren von Zertifikaten finden Sie unter Aktualisieren des SSL/TLS Zertifikats für eine Domain.

Hinzufügen einer von Amazon Route 53 verwalteten benutzerdefinierten Domain

Amazon Route 53 ist ein hochverfügbarer und skalierbarer DNS-Service. Weitere Informationen finden Sie unter Amazon Route 53 im Amazon Route 53 Developer Guide. Wenn Sie bereits eine Route 53-Domain haben, verwenden Sie die folgenden Anweisungen, um Ihre benutzerdefinierte Domain mit Ihrer Amplify-App zu verbinden.

Um eine benutzerdefinierte Domain hinzuzufügen, die von Route 53 verwaltet wird

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie Ihre App aus, die Sie mit einer benutzerdefinierten Domain verbinden möchten.
3. Wählen Sie im Navigationsbereich Hosting, Benutzerdefinierte Domains aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domains die Option Domain hinzufügen aus.
5. Geben Sie den Namen Ihrer Root-Domain ein. Wenn der Name Ihrer Domain beispielsweise lautet https://example.com, geben Sie ein**`example.com`**.

   Wenn Sie mit der Eingabe beginnen, werden alle Stammdomänen, die Sie bereits in Route 53 verwalten, in der Liste angezeigt. Sie können die Domain, die Sie verwenden möchten, aus der Liste auswählen. Wenn Sie die Domain noch nicht besitzen und sie verfügbar ist, können Sie die Domain in Amazon Route 53 erwerben.
6. Nachdem Sie Ihren Domainnamen eingegeben haben, wählen Sie Domain konfigurieren.
7. Standardmäßig erstellt Amplify automatisch zwei Subdomaineinträge für Ihre Domain. Wenn Ihr Domainname beispielsweise example.com lautet, werden Ihnen die Subdomains https://www.example.comangezeigt, https://example.comwobei eine Weiterleitung von der Root-Domain zur WWW-Subdomain eingerichtet ist.

   (Optional) Sie können die Standardkonfiguration ändern, wenn Sie nur Subdomains hinzufügen möchten. Um die Standardkonfiguration zu ändern, wählen Sie im Navigationsbereich Umschreibungen und Weiterleitungen aus und konfigurieren Sie dann Ihre Domain.
8. Wählen Sie das zu SSL/TLS verwendende Zertifikat aus. Sie können entweder das verwaltete Standardzertifikat verwenden, das Amplify für Sie bereitstellt, oder ein benutzerdefiniertes Drittanbieterzertifikat, in das Sie importiert AWS Certificate Manager haben.
   - Verwenden Sie das von Amplify verwaltete Standardzertifikat.
     - Wählen Sie Amplify Managed Certificate.
   - Verwenden Sie ein benutzerdefiniertes Zertifikat eines Drittanbieters.
     a. Wählen Sie Benutzerdefiniertes SSL-Zertifikat.
     b. Wählen Sie das zu verwendende Zertifikat aus der Liste aus.
9. Wählen Sie Domain hinzufügen.

> **Note**
>
> Es kann bis zu 24 Stunden dauern, bis der DNS sich verbreitet und das Zertifikat ausgestellt hat. Hilfe zur Behebung von aufgetretenen Fehlern finden Sie unter. Problembehandlung bei benutzerdefinierten Domains

Hinzufügen einer benutzerdefinierten Domain, die von einem DNS-Drittanbieter verwaltet wird

Wenn Sie Amazon Route 53 nicht zur Verwaltung Ihrer Domain verwenden, können Sie Ihrer mit Amplify bereitgestellten App eine benutzerdefinierte Domain hinzufügen, die von einem DNS-Drittanbieter verwaltet wird.

Wenn Sie dies verwenden GoDaddy, finden Sie spezifische Anweisungen the section called “Aktualisierung von DNS-Einträgen für eine Domain, die verwaltet wird von GoDaddy” für diesen Anbieter.

So fügen Sie eine benutzerdefinierte Domain hinzu, die von einem DNS-Drittanbieter verwaltet wird

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie Ihre App aus, zu der Sie eine benutzerdefinierte Domain hinzufügen möchten.
3. Wählen Sie im Navigationsbereich Hosting, Benutzerdefinierte Domains aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domains die Option Domain hinzufügen aus.
5. Geben Sie den Namen Ihrer Root-Domain ein. Wenn der Name Ihrer Domain beispielsweise lautet https://example.com, geben Sie ein**`example.com`**.
6. Amplify erkennt, dass Sie keine Route 53-Domain verwenden, und bietet Ihnen die Möglichkeit, eine gehostete Zone in Route 53 zu erstellen.
   - Um eine gehostete Zone in Route 53 zu erstellen
     a. Wählen Sie Hosting-Zone auf Route 53 erstellen aus.
     b. Wählen Sie Domain konfigurieren.
     c. Die Nameserver der gehosteten Zone werden in der Konsole angezeigt. Rufen Sie die Website Ihres DNS-Anbieters auf und fügen Sie die Nameserver zu Ihren DNS-Einstellungen hinzu.

d. Wählen Sie Ich habe die oben genannten Nameserver zu meiner Domainregistrierung hinzugefügt.

e. Fahren Sie mit Schritt sieben fort.

- Um mit der manuellen Konfiguration fortzufahren

  a. Wählen Sie Manuelle Konfiguration

  b. Wählen Sie Domain konfigurieren.

  c. Fahren Sie mit Schritt 7 fort.

7. Standardmäßig erstellt Amplify automatisch zwei Subdomaineinträge für Ihre Domain. Wenn Ihr Domainname beispielsweise example.com lautet, werden Ihnen die Subdomains https://www.example.comangezeigt, https://example.comwobei eine Weiterleitung von der Root-Domain zur WWW-Subdomain eingerichtet ist.

   (Optional) Sie können die Standardkonfiguration ändern, wenn Sie nur Subdomains hinzufügen möchten. Um die Standardkonfiguration zu ändern, wählen Sie im Navigationsbereich Umschreibungen und Weiterleitungen aus und konfigurieren Sie Ihre Domain.

8. Wählen Sie das zu SSL/TLS verwendende Zertifikat aus. Sie können entweder das verwaltete Standardzertifikat verwenden, das Amplify für Sie bereitstellt, oder ein benutzerdefiniertes Drittanbieterzertifikat, in das Sie importiert AWS Certificate Manager haben.

   - Verwenden Sie das von Amplify verwaltete Standardzertifikat.

     - Wählen Sie Amplify Managed Certificate.

   - Verwenden Sie ein benutzerdefiniertes Zertifikat eines Drittanbieters.

     a. Wählen Sie Benutzerdefiniertes SSL-Zertifikat.

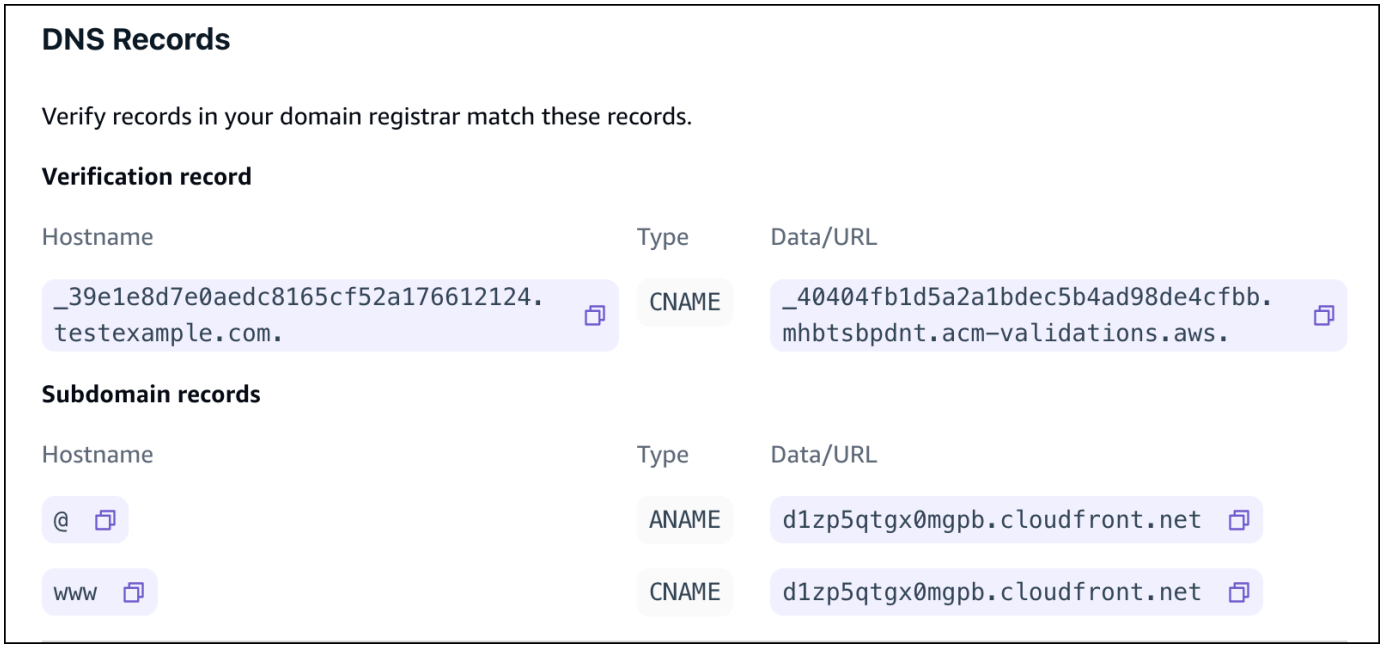
     b. Wählen Sie das zu verwendende Zertifikat aus der Liste aus.

9. Wählen Sie Domain hinzufügen.

10. Wenn Sie in Schritt 6 die Option Gehostete Zone auf Route 53 erstellen ausgewählt haben, fahren Sie mit Schritt 15 fort.

    Wenn Sie sich für Manuelle Konfiguration entschieden haben, müssen Sie in Schritt 6 Ihre DNS-Einträge bei Ihrem Drittanbieter für Domains aktualisieren.

    Wählen Sie im Menü Aktionen die Option DNS-Einträge anzeigen aus. Der folgende Screenshot zeigt die DNS-Einträge, die in der Konsole angezeigt werden.



11. Führen Sie eine der folgenden Aktionen aus:

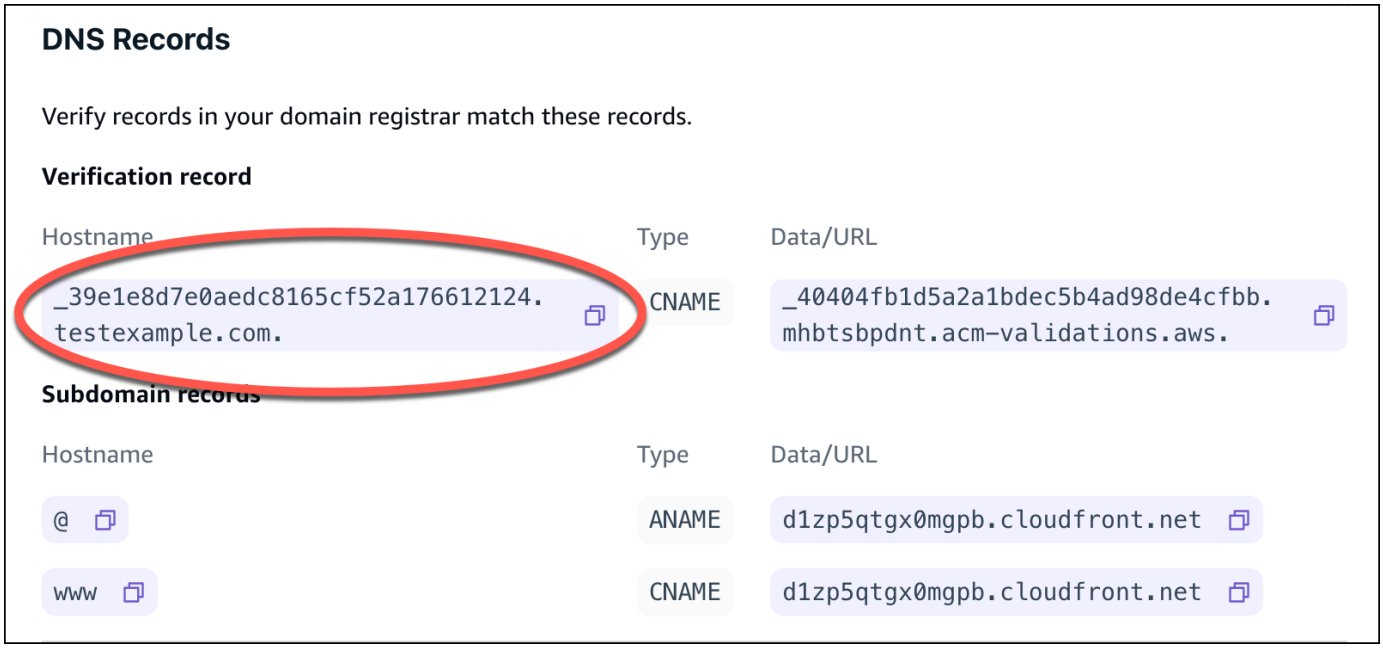
    - Wenn Sie verwenden GoDaddy, gehen Sie zuAktualisierung von DNS-Einträgen für eine Domain, die verwaltet wird von GoDaddy.
    - Wenn Sie einen anderen DNS-Drittanbieter verwenden, fahren Sie mit dem nächsten Schritt in diesem Verfahren fort.

12. Gehen Sie zur Website Ihres DNS-Anbieters, melden Sie sich bei Ihrem Konto an und suchen Sie nach den DNS-Verwaltungseinstellungen für Ihre Domain. Sie werden zwei CNAME-Einträge konfigurieren.

13. Konfigurieren Sie den ersten CNAME-Eintrag so, dass er Ihre Subdomain auf den AWS Validierungsserver verweist.

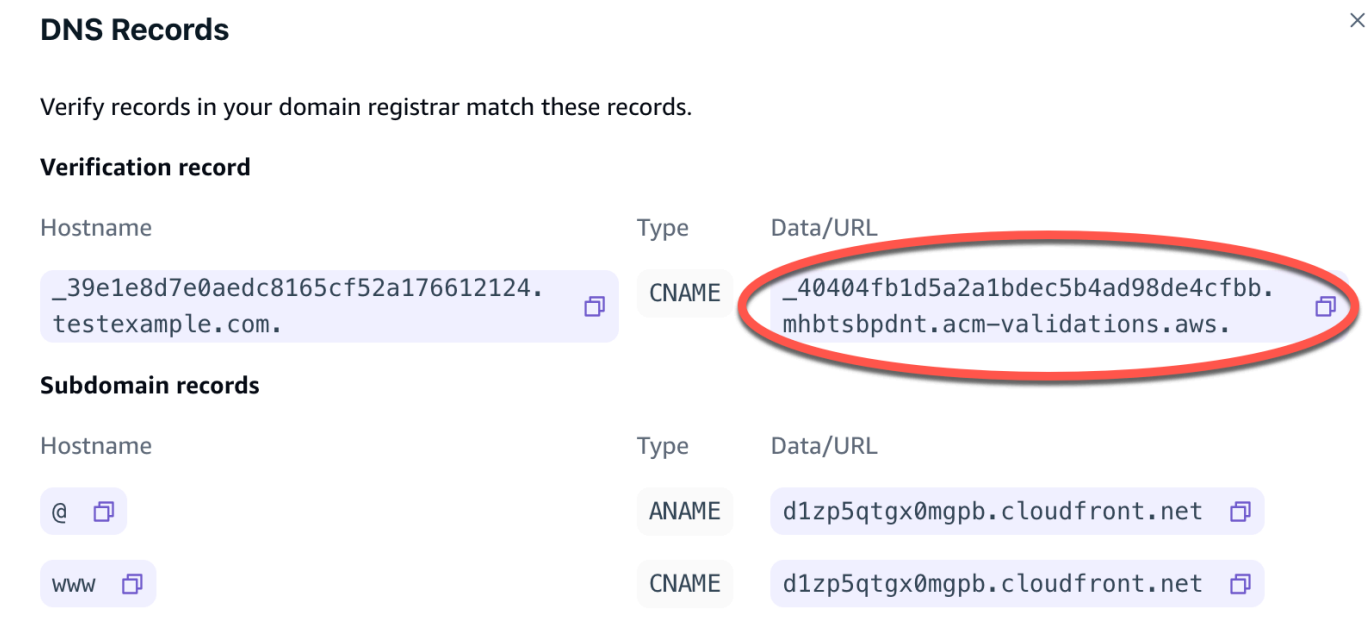
    Wenn die Amplify-Konsole einen DNS-Eintrag zur Überprüfung der Inhaberschaft Ihrer Subdomain anzeigt, z. B. _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, geben Sie nur für den CNAME-Record den Subdomainnamen ein. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

    Der folgende Screenshot zeigt den Speicherort des zu verwendenden Bestätigungsdatensatzes.



Wenn die Amplify-Konsole einen ACM-Validierungsservereintrag wie _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws anzeigt, geben Sie als CNAME-Datensatzwert ein. **`_cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`**

Der folgende Screenshot zeigt den Speicherort des zu verwendenden ACM-Bestätigungsdatensatzes.



Amplify verwendet diese Informationen, um die Inhaberschaft Ihrer Domain zu überprüfen und ein SSL/TLS Zertifikat für Ihre Domain zu generieren. Sobald Amplify den Besitz Ihrer Domain bestätigt hat, wird der gesamte Traffic über HTTPS/2 bedient.

> **Note**
>
> Das von AWS Certificate Manager (ACM) generierte Standard-Amplify-Zertifikat ist 13 Monate gültig und verlängert sich automatisch, solange Ihre App bei Amplify gehostet wird. Amplify kann das Zertifikat nicht erneuern, wenn der CNAME-Bestätigungsdatensatz geändert oder gelöscht wurde. Sie müssen die Domain in der Amplify-Konsole löschen und erneut hinzufügen.

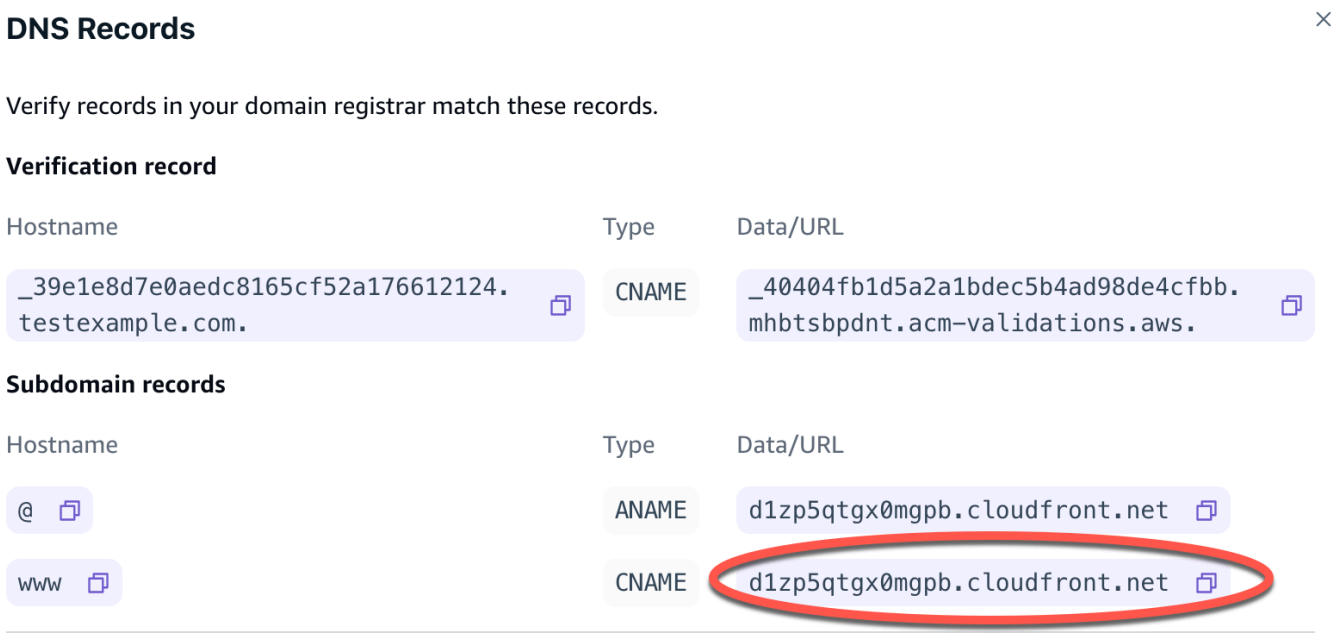
> **Important**
>
> Es ist wichtig, dass Sie diesen Schritt kurz nach dem Hinzufügen Ihrer benutzerdefinierten Domain in der Amplify-Konsole ausführen. Das AWS Certificate Manager (ACM) versucht sofort, die Inhaberschaft zu überprüfen. Mit der Zeit werden die Kontrollen seltener. Wenn Sie Ihre CNAME-Einträge einige Stunden nach der Erstellung Ihrer App hinzufügen oder aktualisieren, kann dies dazu führen, dass Ihre App im Status „Ausstehende Überprüfung“ hängen bleibt.

14. Konfigurieren Sie einen zweiten CNAME-Eintrag, um Ihre Subdomains auf die Amplify-Domain zu verweisen. Wenn Ihre Subdomain beispielsweise www.example.com ist, geben Sie www als Subdomainnamen ein.

    Wenn die Amplify-Konsole die Domain für Ihre App als d111111abcdef8.cloudfront.net anzeigt, geben Sie für die Amplify-Domain ein. **`d111111abcdef8.cloudfront.net`**

    Wenn Sie Produktions-Traffic haben, empfehlen wir Ihnen, diesen CNAME-Eintrag zu aktualisieren, nachdem Ihr Domainstatus in der Amplify-Konsole als VERFÜGBAR angezeigt wird.

    Der folgende Screenshot zeigt den Speicherort des zu verwendenden Domainnamen-Eintrags.



15. Konfigurieren Sie den ANAME/ALIAS Datensatz so, dass er auf die Root-Domain Ihrer App verweist (zum Beispiel https://example.com). Ein ANAME-Eintrag verweist das Stammverzeichnis Ihrer Domain auf einen Hostnamen. Wenn Sie Produktionsdatenverkehr haben, empfehlen wir Ihnen, Ihren ANAME-Eintrag zu aktualisieren, nachdem Ihr Domainstatus in der Konsole als VERFÜGBAR angezeigt wird. Für DNS-Anbieter, die keinen ANAME/ALIAS Support anbieten, empfehlen wir dringend, Ihr DNS auf Route 53 zu migrieren. Weitere Informationen finden Sie unter Amazon Route 53 als Ihren DNS-Service konfigurieren.

> **Note**
>
> Die Überprüfung des Domainbesitzes und der DNS-Weitergabe für Domains von Drittanbietern kann bis zu 48 Stunden dauern. Hilfe zur Behebung von aufgetretenen Fehlern finden Sie unter Problembehandlung bei benutzerdefinierten Domains.

Aktualisierung von DNS-Einträgen für eine Domain, die verwaltet wird von GoDaddy

Wenn GoDaddy es Ihr DNS-Anbieter ist, verwenden Sie die folgenden Anweisungen, um Ihre DNS-Einträge in der GoDaddy Benutzeroberfläche zu aktualisieren, um die Verbindung Ihrer Amplify-App mit Ihrer GoDaddy Domain abzuschließen.

Um eine benutzerdefinierte Domain hinzuzufügen, die verwaltet wird von GoDaddy

1. Bevor Sie Ihre DNS-Einträge mit aktualisieren können GoDaddy, führen Sie die Schritte eins bis neun des Verfahrens durchthe section called “Hinzufügen einer benutzerdefinierten Domain, die von einem DNS-Drittanbieter verwaltet wird”.
2. Loggen Sie sich in Ihr GoDaddy Konto ein.
3. Suchen Sie in Ihrer Domainliste nach der Domain, die Sie hinzufügen möchten, und wählen Sie Manage DNS aus.
4. GoDaddy Zeigt auf der DNS-Seite im Bereich DNS-Einträge eine Liste der Einträge für Ihre Domain an. Sie müssen zwei neue CNAME-Einträge hinzufügen.
5. Erstellen Sie den ersten CNAME-Eintrag, um Ihre Subdomains auf die Amplify-Domain zu verweisen.

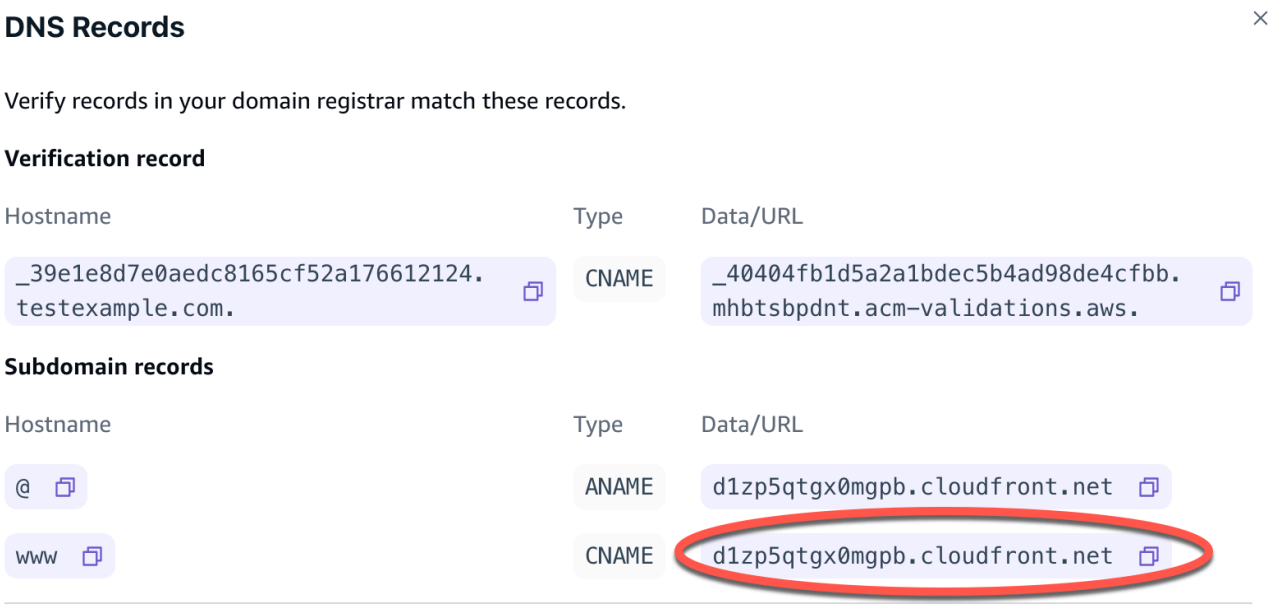
   a. Wählen Sie im Abschnitt DNS-Einträge die Option Neuen Eintrag hinzufügen aus.

   b. Wählen Sie als Typ die Option CNAME aus.

   c. Geben Sie als Name nur die Subdomain ein. Wenn Ihre Subdomain beispielsweise www.example.com ist, geben Sie www als Namen ein.

   d. Sehen Sie sich für Value Ihre DNS-Einträge in der Amplify-Konsole an und geben Sie dann den Wert ein. Wenn die Amplify-Konsole die Domain für Ihre App als d111111abcdef8.cloudfront.net anzeigt, geben Sie Value ein.
   `d111111abcdef8.cloudfront.net`

   Der folgende Screenshot zeigt den Speicherort des zu verwendenden Domainnamen-Datensatzes.



e. Wählen Sie Speichern.

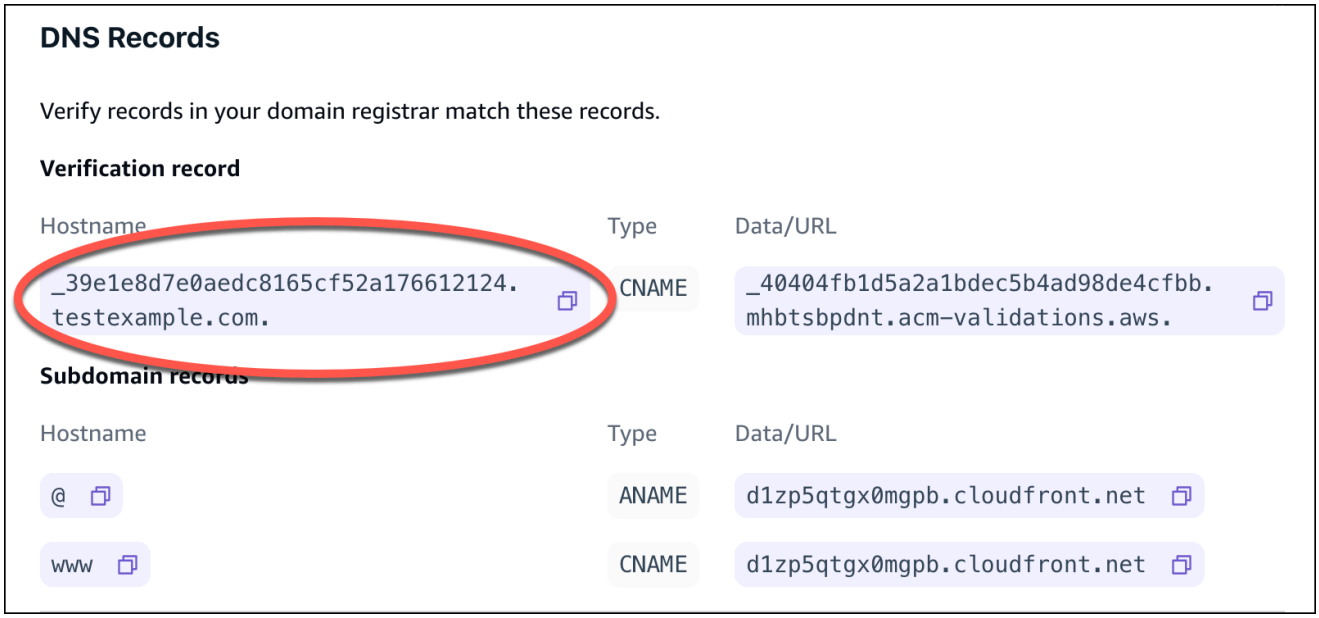
6. Erstellen Sie den zweiten CNAME-Eintrag, der auf den AWS Certificate Manager (ACM) - Validierungsserver verweist. Ein einzelnes validiertes ACM generiert ein SSL/TLS Zertifikat für Ihre Domain.

   a. Wählen Sie als Typ die Option CNAME aus.

   b. Geben Sie als Name die Subdomain ein.

      Wenn der DNS-Eintrag in der Amplify-Konsole zur Überprüfung der Inhaberschaft Ihrer Subdomain beispielsweise _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com lautet, geben Sie nur Name ein. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

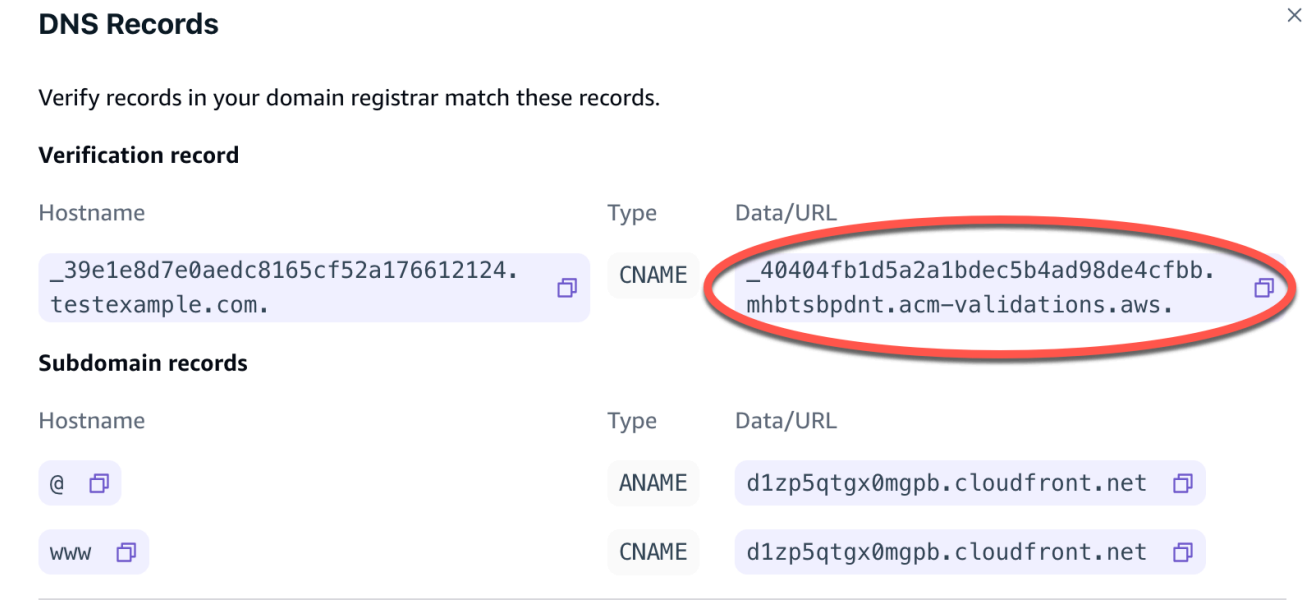
      Der folgende Screenshot zeigt den Speicherort des zu verwendenden Bestätigungsdatensatzes.



c. Geben Sie unter Value das ACM-Validierungszertifikat ein.

Wenn der Validierungsserver beispielsweise _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws ist, geben Sie _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws für Value ein.

Der folgende Screenshot zeigt den Speicherort des zu verwendenden ACM-Bestätigungsdatensatzes.



d. Wählen Sie Speichern.

> **Note**
>
> Das von AWS Certificate Manager (ACM) generierte Standard-Amplify-Zertifikat ist 13 Monate gültig und verlängert sich automatisch, solange Ihre App bei Amplify gehostet wird. Amplify kann das Zertifikat nicht erneuern, wenn der CNAME-Bestätigungsdatensatz geändert oder gelöscht wurde. Sie müssen die Domain in der Amplify-Konsole löschen und erneut hinzufügen.

7. Dieser Schritt ist für Subdomains nicht erforderlich. GoDaddy unterstützt keinen ANAME/ALIAS records. For DNS providers that do not have ANAME/ALIAS Support. Wir empfehlen dringend, Ihr DNS zu Amazon Route 53 zu migrieren. Weitere Informationen finden Sie unter Amazon Route 53 als Ihren DNS-Service konfigurieren.

   Wenn Sie Ihren Anbieter behalten GoDaddy und die Root-Domain aktualisieren möchten, fügen Sie Forwarding hinzu und richten Sie eine Domain-Forward ein:

   a. Suchen Sie auf der DNS-Seite nach dem Menü oben auf der Seite und wählen Sie Weiterleitung aus.
   b. Wählen Sie im Bereich Domain die Option Weiterleitung hinzufügen aus.
   c. Wählen Sie http://und geben Sie dann als Ziel-URL den Namen Ihrer Subdomain ein, zu der Sie weiterleiten möchten (z. B. www.example.com).
   d. Wählen Sie als Weiterleitungstyp die Option Temporär (302) aus.
   e. Wählen Sie „Speichern“.

Aktualisierung des SSL/TLS Zertifikats für eine Domain

Sie können das SSL/TLS Zertifikat, das für eine Domain verwendet wird, jederzeit ändern. Sie können beispielsweise von der Verwendung eines verwalteten Zertifikats zur Verwendung eines benutzerdefinierten Zertifikats wechseln. Dies ist hilfreich, wenn Sie das Zertifikat und dessen Ablaufbenachrichtigungen verwalten möchten. Sie können auch das benutzerdefinierte Zertifikat ändern, das für die Domain verwendet wird. Wenn Sie Änderungen am SSL-Zertifikat vornehmen, führt dies zu keinen Ausfallzeiten für Ihre aktive Domain. Weitere Informationen zu Zertifikaten finden Sie unter Verwenden von SSL/TLS-Zertifikaten.

Gehen Sie wie folgt vor, um den Zertifikatstyp oder das benutzerdefinierte Zertifikat zu aktualisieren, das für eine Domain verwendet wird.

Um das Zertifikat einer Domain zu aktualisieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie Ihre App aus, die Sie aktualisieren möchten.
3. Wählen Sie im Navigationsbereich Hosting, Benutzerdefinierte Domains aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domänen die Option Domänenkonfiguration aus.
5. Suchen Sie auf der Detailseite für Ihre Domain den Abschnitt Benutzerdefiniertes SSL-Zertifikat. Das Verfahren zur Aktualisierung Ihres Zertifikats hängt von der Art der Änderung ab, die Sie vornehmen möchten.
   - Um von einem benutzerdefinierten Zertifikat zum standardmäßigen verwalteten Amplify-Zertifikat zu wechseln
     - Wählen Sie Amplify Managed Certificate.
   - Um von einem verwalteten Zertifikat zu einem benutzerdefinierten Zertifikat zu wechseln
     a. Wählen Sie Benutzerdefiniertes SSL-Zertifikat.
     b. Wählen Sie das zu verwendende Zertifikat aus der Liste aus.
   - Um ein benutzerdefiniertes Zertifikat in ein anderes benutzerdefiniertes Zertifikat zu ändern
     - Wählen Sie für Benutzerdefiniertes SSL-Zertifikat das neue zu verwendende Zertifikat aus der Liste aus.
6. Wählen Sie Speichern. Die Statusdetails für die Domain geben an, dass Amplify den SSL-Erstellungsprozess für ein verwaltetes Zertifikat oder den Konfigurationsprozess für ein benutzerdefiniertes Zertifikat eingeleitet hat.

Verwaltung von Subdomains

Eine Subdomain ist der Teil Ihrer URL, der vor Ihrem Domainnamen steht. Zum Beispiel ist www die Subdomain von www.amazon.com und aws ist die Subdomain von aws.amazon.com. Wenn Sie bereits eine Produktionswebsite haben, möchten Sie möglicherweise nur eine Subdomain verbinden. Subdomains können auch mehrstufig sein, zum Beispiel hat beta.alpha.example.com die mehrstufige Subdomain beta.alpha.

Nur um eine Subdomain hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie Ihre App aus, zu der Sie eine Subdomain hinzufügen möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Benutzerdefinierte Domains aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domains die Option Domain hinzufügen aus.
5. Geben Sie den Namen Ihrer Root-Domain ein und wählen Sie dann Domain konfigurieren. Wenn der Name Ihrer Domain beispielsweise lautet https://example.com, geben Sie example.com ein.
6. Wählen Sie Exclude root und ändern Sie den Namen der Subdomain. Wenn die Domain beispielsweise example.com lautet, können Sie sie so ändern, dass nur die Subdomain Alpha hinzugefügt wird.
7. Wählen Sie Domain hinzufügen.

Um eine mehrstufige Subdomain hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie Ihre App aus, zu der Sie eine mehrstufige Subdomain hinzufügen möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Benutzerdefinierte Domains aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domains die Option Domain hinzufügen aus.
5. Geben Sie den Namen einer Domain mit einer Subdomain ein, wählen Sie Exclude root aus und ändern Sie die Subdomain, um eine neue Ebene hinzuzufügen.

   Wenn Sie beispielsweise eine Domain namens alpha.example.com haben und eine mehrstufige Subdomain beta.alpha.example.com erstellen möchten, würden Sie beta als Subdomänenwert eingeben.
6. Wählen Sie Domain hinzufügen.

Um eine Subdomain hinzuzufügen oder zu bearbeiten

Nachdem Sie einer App eine benutzerdefinierte Domain hinzugefügt haben, können Sie eine bestehende Subdomain bearbeiten oder eine neue Subdomain hinzufügen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie Ihre App aus, für die Sie Subdomains verwalten möchten.

3. Wählen Sie im Navigationsbereich Hosting und dann Benutzerdefinierte Domains aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domänen die Option Domänenkonfiguration aus.
5. Im Abschnitt Subdomains können Sie Ihre vorhandenen Subdomains nach Bedarf bearbeiten.
6. (Optional) Um eine neue Subdomain hinzuzufügen, wählen Sie Neu hinzufügen.
7. Wählen Sie Speichern.

Platzhalter-Subdomänen einrichten

Amplify Hosting unterstützt jetzt Wildcard-Subdomains. Eine Wildcard-Subdomain ist eine Sammel-Subdomain, mit der Sie bestehende und nicht existierende Subdomains auf einen bestimmten Zweig Ihrer Anwendung verweisen können. Wenn Sie einen Platzhalter verwenden, um alle Subdomains in einer App einem bestimmten Zweig zuzuordnen, können Sie den Benutzern Ihrer App in jeder Subdomain denselben Inhalt bereitstellen und müssen nicht jede Subdomain einzeln konfigurieren.

Um eine Wildcard-Subdomain zu erstellen, geben Sie ein Sternchen (*) als Subdomänennamen an. Wenn Sie beispielsweise die Platzhalter-Subdomain `*.example.com` für einen bestimmten Zweig Ihrer App angeben, wird jede URL, die mit example.com endet, an den Zweig weitergeleitet. In diesem Fall werden Anfragen für `dev.example.com` und an die `prod.example.com` Subdomain weitergeleitet. `*.example.com`

Beachten Sie, dass Amplify Platzhalter-Subdomains nur für eine benutzerdefinierte Domain unterstützt. Sie können diese Funktion nicht mit der Standarddomain verwenden. `amplifyapp.com`

Die folgenden Anforderungen gelten für Wildcard-Subdomänen:

- Der Subdomainname darf nur mit einem Sternchen (*) angegeben werden.
- Sie können keinen Platzhalter verwenden, um einen Teil eines Subdomain-Namens wie folgt zu ersetzen: *domain.example.com.
- Sie können eine Subdomain nicht in der Mitte eines Domainnamens ersetzen, wie hier: subdomain.*.example.com.
- Standardmäßig decken alle von Amplify bereitgestellten Zertifikate alle Subdomains für eine benutzerdefinierte Domain ab.

Um eine Wildcard-Subdomain hinzuzufügen oder zu löschen

Nachdem Sie einer App eine benutzerdefinierte Domain hinzugefügt haben, können Sie eine Wildcard-Subdomain für einen App-Branch hinzufügen.

1. Melden Sie sich bei der Amplify Hosting-Konsole an AWS-Managementkonsole und öffnen Sie sie.
2. Wählen Sie Ihre App aus, für die Sie Wildcard-Subdomains verwalten möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Benutzerdefinierte Domains aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domänen die Option Domänenkonfiguration aus.
5. Im Abschnitt Subdomains können Sie Wildcard-Subdomains hinzufügen oder löschen.
   - Um eine neue Wildcard-Subdomain hinzuzufügen
     a. Wählen Sie Add new (Neuen hinzufügen) aus.
     b. Geben Sie für die Subdomain eine ein. *****
     c. Wählen Sie für Ihren App-Branch einen Zweignamen aus der Liste aus.
     d. Wählen Sie Speichern.
   - Um eine Wildcard-Subdomain zu löschen
     a. Wählen Sie neben dem Namen der Subdomain die Option Entfernen aus. Der Datenverkehr zu einer Subdomain, die nicht explizit konfiguriert ist, wird gestoppt, und Amplify Hosting gibt für diese Anfragen einen 404-Statuscode zurück.
     b. Wählen Sie Speichern.

Automatische Subdomains für eine benutzerdefinierte Amazon Route 53-Domain einrichten

Nachdem eine App mit einer benutzerdefinierten Domain in Route 53 verbunden wurde, können Sie mit Amplify automatisch Subdomains für neu verbundene Filialen erstellen. Wenn Sie beispielsweise Ihren Dev-Branch verbinden, kann Amplify automatisch dev.exampledomain.com erstellen. Wenn Sie einen Branch löschen, werden alle zugehörigen Subdomains automatisch gelöscht.

Um die automatische Erstellung von Subdomains für neu verbundene Filialen einzurichten

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.

2. Wählen Sie eine App aus, die mit einer benutzerdefinierten Domain verbunden ist, die in Route 53 verwaltet wird.
3. Wählen Sie im Navigationsbereich Hosting und dann Benutzerdefinierte Domänen aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Domänen die Option Domänenkonfiguration aus.
5. Aktivieren Sie im Abschnitt Automatische Erstellung von Subdomains die Funktion.

> **Note**
>
> Diese Funktion ist nur für Root-Domains verfügbar, z. B. exampledomain.com. Die Amplify-Konsole zeigt dieses Kontrollkästchen nicht an, wenn Ihre Domain bereits eine Subdomain ist, z. B. dev.exampledomain.com.

Webvorschauen mit Subdomains

Nachdem Sie die automatische Erstellung von Subdomains anhand der obigen Anweisungen aktiviert haben, können Sie auf die Pull-Request-Webvorschauen Ihrer App auch mit automatisch erstellten Subdomains zugreifen. Wenn ein Pull-Request geschlossen wird, werden der zugehörige Branch und die Subdomain automatisch gelöscht. Weitere Informationen zum Einrichten von Webvorschauen für Pull Requests finden Sie unter. Webvorschauen für Pull-Requests

Problembehandlung bei benutzerdefinierten Domains

Wenn Sie beim Hinzufügen einer benutzerdefinierten Domain zu einer App in der AWS Amplify Konsole auf Probleme stoßen, finden Sie weitere Informationen Fehlerbehebung bei benutzerdefinierten Domains im Kapitel Amplify-Fehlerbehebung. Wenn Sie dort keine Lösung für Ihr Problem finden, wenden Support Sie sich an. Weitere Informationen finden Sie unter Erstellen eines Support-Falls im Benutzerhandbuch von AWS Support .

Firewall-Unterstützung für von Amplify gehostete Websites

Die Firewall-Unterstützung für von Amplify gehostete Websites ermöglicht es Ihnen, Ihre Webanwendungen durch eine direkte Integration mit AWS WAF zu schützen. AWS WAF ermöglicht es Ihnen, eine Reihe von Regeln zu konfigurieren, die als Web Access Control List (Web ACL) bezeichnet werden und Webanfragen auf der Grundlage von anpassbaren Websicherheitsregeln und -bedingungen, die Sie definieren, zulassen, blockieren oder überwachen (zählen). Wenn Sie Ihre Amplify-App mit integrieren AWS WAF, erhalten Sie mehr Kontrolle und Einblick in den von Ihrer App akzeptierten HTTP-Verkehr. Weitere Informationen dazu finden Sie AWS WAF unter How AWS WAF Works im AWS WAF Developer Guide.

Firewall-Unterstützung ist in allen Bereichen verfügbar, AWS-Regionen in denen Amplify Hosting tätig ist. Diese Integration fällt unter eine AWS WAF globale Ressource, ähnlich CloudFront wie. Web ACLs kann an mehrere Amplify Hosting-Apps angehängt werden, sie müssen sich jedoch in derselben Region befinden.

Sie können AWS WAF es verwenden, um Ihre Amplify-App vor gängigen Web-Exploits wie SQL-Injection und Cross-Site-Scripting zu schützen. Diese könnten die Verfügbarkeit und Leistung Ihrer App beeinträchtigen, die Sicherheit gefährden oder übermäßig viele Ressourcen verbrauchen. Sie können beispielsweise Regeln erstellen, um Anfragen aus bestimmten IP-Adressbereichen, Anfragen von CIDR-Blöcken, Anfragen, die aus einem bestimmten Land oder einer bestimmten Region stammen, oder Anfragen, die unerwarteten SQL-Code oder Skripting enthalten, zuzulassen oder zu blockieren.

Sie können auch Regeln erstellen, die mit einer bestimmten Zeichenfolge oder einem regulären Ausdrucksmuster in HTTP-Headern, einer Methode, einer Abfragezeichenfolge, einer URI und dem Anforderungstext (nur für die ersten 8 KB) übereinstimmen. Darüber hinaus können Sie Regeln erstellen, um Ereignisse von bestimmten Benutzeragenten, Bots und Content Scrapern zu blockieren. Beispielsweise können Sie mit durchsatzbasierten Regeln die Anzahl der zulässigen Webanforderungen angeben, die von jeder Client-IP in einem sich anschließenden, fortlaufend aktualisierten 5-Minuten-Zeitraum zugelassen werden.

Weitere Informationen zu den unterstützten Regeltypen und zusätzlichen AWS WAF Funktionen finden Sie im AWS WAF Entwicklerhandbuch und in der AWS WAF API-Referenz.

> **Important**
>
> Sicherheit ist eine gemeinsame Verantwortung zwischen Ihnen AWS und Ihnen. AWS WAF ist nicht die Lösung für alle Probleme mit der Internetsicherheit, und Sie müssen sie so konfigurieren, dass sie Ihre Sicherheits- und Compliance-Ziele erfüllt. Weitere Informationen zur Anwendung des Modells der gemeinsamen Verantwortung bei der Nutzung AWS WAF finden Sie unter Sicherheit bei der Nutzung des AWS WAF Dienstes.

Themen

Aktivierung AWS WAF für eine Amplify-Anwendung in der AWS-Managementkonsole

Sie können den AWS WAF Schutz für eine Amplify-App entweder in der Amplify-Konsole oder in der Konsole aktivieren. AWS WAF

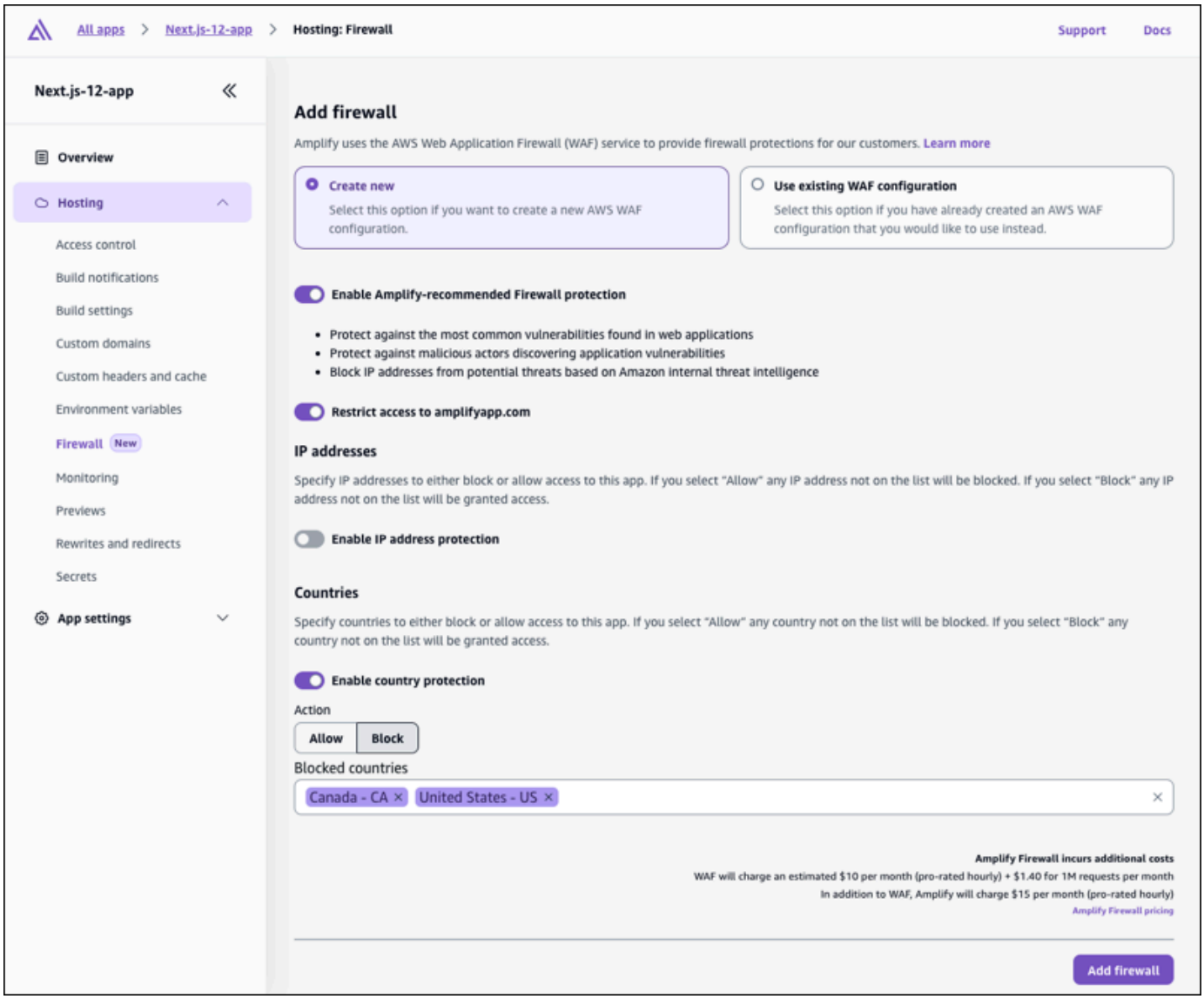
- Amplify-Konsole — Sie können die Firewall-Funktionen für eine bestehende Amplify-App aktivieren, indem Sie Ihrer App in der Amplify-Konsole eine AWS WAF Web-ACL zuordnen. Verwenden Sie den Ein-Klick-Schutz, um eine Web-ACL mit vorkonfigurierten Regeln zu erstellen, die wir für die meisten Apps als bewährte Methode betrachten. Sie haben die Möglichkeit, den Zugriff nach IP-Adresse und Land anzupassen. Die Anweisungen in diesem Abschnitt beschreiben die Einrichtung von Schutzmaßnahmen mit einem Klick.
- AWS WAF Konsole — Verwenden Sie eine vorkonfigurierte Web-ACL, die Sie in der AWS WAF Konsole oder mithilfe der erstellen. AWS WAF APIs Sie müssen eine Website erstellen ACLs , die Sie mit einer Amplify-App in der Region Global (CloudFront) verknüpfen möchten. In Ihrem gibt es ACLs möglicherweise bereits regionale Websites AWS-Konto, die jedoch nicht mit Amplify kompatibel sind. Anleitungen für die ersten Schritte finden Sie unter Einrichtung AWS WAF und Komponenten im AWS WAF Entwicklerhandbuch.

Gehen Sie wie folgt vor, AWS WAF um eine vorhandene App in der Amplify-Konsole zu aktivieren.

AWS WAF Für eine bestehende Amplify-App aktivieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite Alle Apps den Namen der bereitgestellten App aus, für die die Firewall-Funktion aktiviert werden soll.
3. Wählen Sie im Navigationsbereich Hosting und dann Firewall aus.

   Der folgende Screenshot zeigt, wie Sie in der Amplify-Konsole zur Seite „Firewall hinzufügen“ navigieren.



4. Auf der Seite „Firewall hinzufügen“ hängen Ihre Aktionen davon ab, ob Sie eine neue AWS WAF Konfiguration erstellen oder eine bestehende verwenden möchten.

   - Erstellen Sie eine neue AWS WAF Konfiguration.

     a. Wählen Sie Neu erstellen.

     b. Aktivieren Sie optional eine der folgenden Konfigurationen:

        i. Aktivieren Sie die Option „Von Amplify empfohlenen Firewall-Schutz aktivieren“.

        ii. Aktiviere die Option Zugriff auf amplifyapp.com einschränken, um den Zugriff auf deine App in der Standard-Amplify-Domain zu verhindern.

        iii. Aktivieren Sie für IP-Adressen die Option IP-Adressschutz aktivieren.

           A. Wählen Sie unter Aktion die Option Zulassen aus, wenn Sie die IP-Adressen angeben möchten, die Zugriff haben und alle anderen gesperrt werden sollen. Wählen Sie Blockieren, wenn Sie die IP-Adressen angeben möchten, die gesperrt werden und auf die alle anderen zugreifen können.

           B. Wählen Sie als IP-Version entweder IPV4oder aus IPV6.

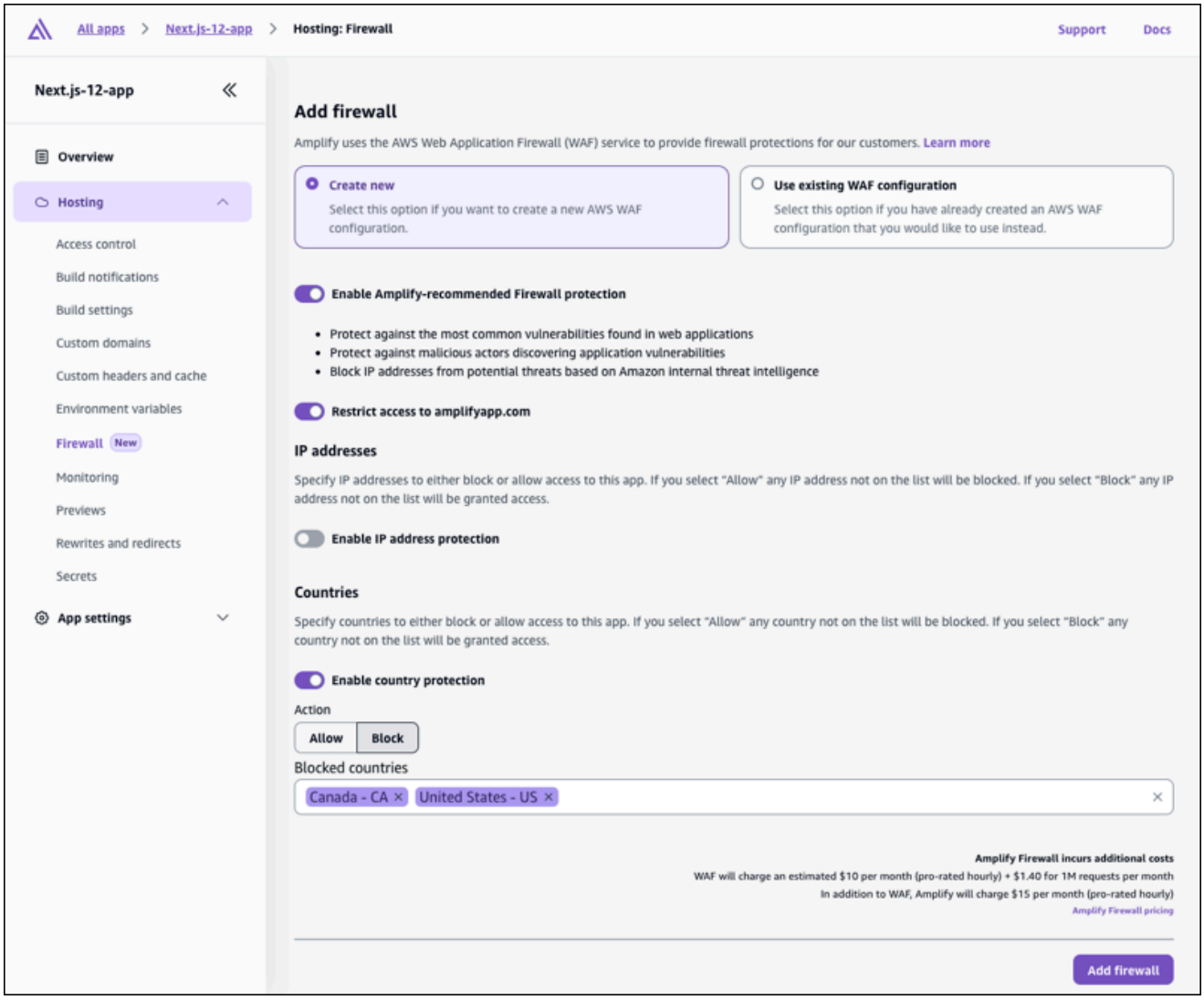
           C. Geben Sie im Textfeld IP-Adressen entweder Ihre erlaubten oder gesperrten IP-Adressen, eine pro Zeile, im CIDR-Format ein.

        iv. Aktivieren Sie für Länder die Option Länderschutz aktivieren.

           A. Wählen Sie unter Aktion die Option Zulassen aus, wenn Sie angeben möchten, welche Länder Zugriff haben und alle anderen Länder gesperrt werden sollen. Wählen Sie Blockieren, wenn Sie angeben möchten, welche Länder gesperrt werden und alle anderen Länder Zugriff haben sollen.

           B. Wählen Sie unter Länder entweder Ihre erlaubten oder Ihre gesperrten Länder aus der Liste aus.

   Der folgende Screenshot zeigt, wie Sie eine neue AWS WAF Konfiguration für eine App aktivieren.



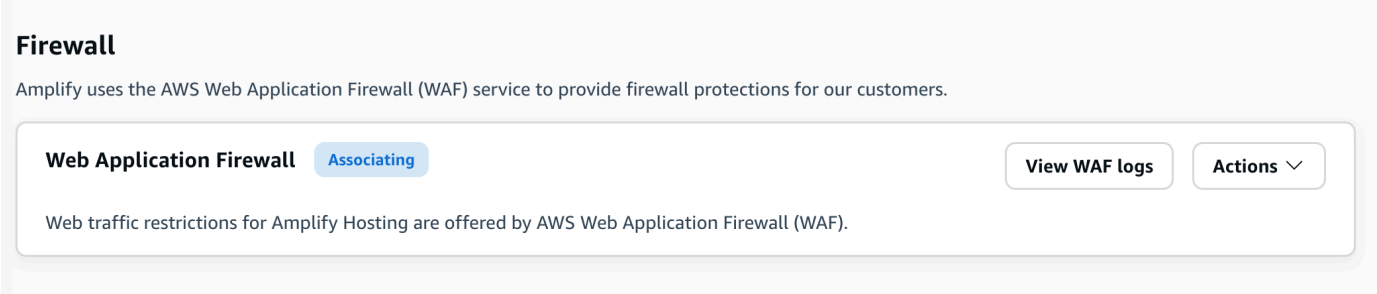
- Verwenden Sie eine vorhandene AWS WAF Konfiguration.

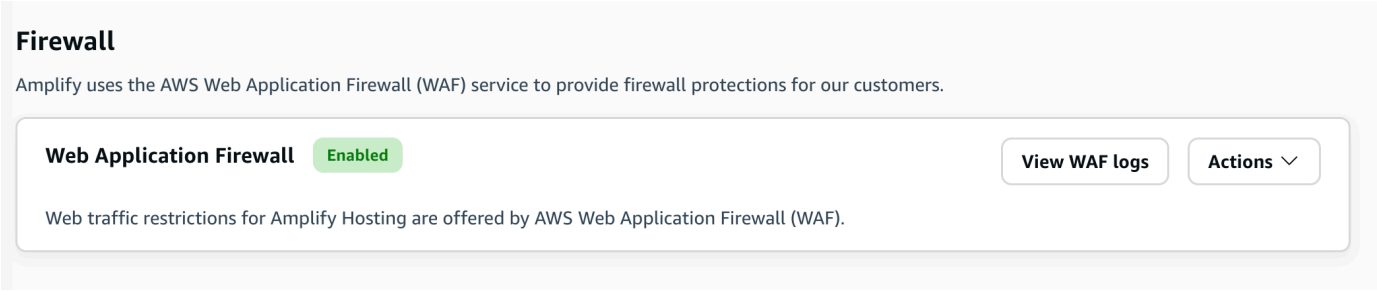
  a. Wählen Sie Bestehende AWS WAF Konfiguration verwenden aus.

  b. Wählen Sie eine gespeicherte Konfiguration aus der Liste der Web-Ins ACLs AWS WAF in Ihrem aus AWS-Konto. Die Web-ACL, die Sie mit Ihrer Amplify-App verknüpfen, muss in der Region Global (CloudFront) erstellt werden. In Ihrem gibt es ACLs möglicherweise bereits regionale Websites AWS-Konto, die jedoch nicht mit Amplify kompatibel sind.

5. Wählen Sie Firewall hinzufügen.
6. Auf der Firewall-Seite wird der Status Zuordnen angezeigt, was darauf hinweist, dass die AWS WAF Einstellungen weitergegeben werden. Wenn der Vorgang abgeschlossen ist, ändert sich der Status in Aktiviert.

Die folgenden Screenshots zeigen den Firewall-Fortschrittsstatus in der Amplify-Konsole und geben an, wann die AWS WAF Konfiguration Zuordnen und Aktiviert ist.





Trennen Sie eine Web-ACL von einer Amplify-Anwendung

Sie können keine Web-ACL löschen, die mit einer Amplify-App verknüpft ist. Sie müssen zuerst die Web-ACL von der App in der Amplify-Konsole trennen. Dann können Sie es in der AWS WAF Konsole löschen.

So trennen Sie eine Web-ACL von einer Amplify-App

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite Alle Apps den Namen der App aus, zu der Sie eine Web-ACL trennen möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Firewall aus.
4. Wählen Sie auf der Firewall-Seite Aktionen und dann Firewall trennen aus.
5. Geben Sie im Bestätigungsfenster den Text ein **`disassociate`** und wählen Sie dann Firewall trennen aus.
6. Auf der Firewall-Seite wird der Status Disassociated angezeigt, was darauf hinweist, dass die AWS WAF Einstellungen weitergegeben werden.

Wenn der Vorgang abgeschlossen ist, können Sie die Web-ACL in der Konsole löschen. AWS WAF

Aktivierung AWS WAF für eine Amplify-Anwendung mit dem AWS CDK

Sie können das verwenden AWS Cloud Development Kit (AWS CDK) , um es AWS WAF für eine Amplify-Anwendung zu aktivieren. Weitere Informationen zur Verwendung des CDK finden Sie unter Was ist das CDK? im AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch.

Das folgende TypeScript Codebeispiel zeigt, wie Sie eine AWS CDK App mit zwei CDK-Stacks erstellen: einen für Amplify und einen für. AWS WAF Beachten Sie, dass der AWS WAF Stack in der Region USA Ost (Nord-Virginia) (us-east-1) bereitgestellt werden muss. Der Amplify-Anwendungsstapel kann in einer anderen Region bereitgestellt werden. Sie müssen die Web-ACL, die Sie der Amplify-App zuordnen möchten, in der Region Global (CloudFront) erstellen. In Ihrem gibt es ACLs möglicherweise bereits regionale Websites AWS-Konto, die jedoch nicht mit Amplify kompatibel sind.

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

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

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

export class WAFStack extends cdk.Stack {
```

```
  constructor(scope: Construct, id: string, props: WafStackProps) {
    super(scope, id, props);
    const webAcl = new wafv2.CfnWebACL(this, "WebACL", {
      defaultAction: { allow: {} },
      scope: "CLOUDFRONT",
      rules: [
        // Add your own rules here.
      ],
      visibilityConfig: {
        cloudWatchMetricsEnabled: true,
        metricName: "my-metric-name",
        sampledRequestsEnabled: true,
      },
    });

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

const app = new cdk.App();

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

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

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

So lässt sich Amplify integrieren mit AWS WAF

Die folgende Liste enthält spezifische Informationen zur Integration der Firewall-Unterstützung AWS WAF und zu den Einschränkungen, die bei der Erstellung von Websites ACLs und deren Verknüpfung mit Amplify-Apps zu berücksichtigen sind.

- Sie können sie AWS WAF für jede Art von Amplify-App aktivieren. Dazu gehören alle unterstützten Frameworks, serverseitig gerenderte (SSR) Apps und vollständig statische Websites. AWS WAF wird für Amplify Gen 1- und Gen 2-Apps unterstützt.
- Sie müssen eine Website erstellen ACLs , die Sie mit einer Amplify-App in der Region Global (CloudFront) verknüpfen möchten. In Ihrem gibt es ACLs möglicherweise bereits regionale Websites AWS-Konto, die jedoch nicht mit Amplify kompatibel sind.
- Die Web-ACL und die Amplify-App müssen in derselben AWS-Konto Datei erstellt werden. Sie können AWS Firewall Manager es verwenden, um AWS WAF Regeln übergreifend zu replizieren AWS-Konten, um es einfacher zu machen, Organisationsregeln zentralisiert und über mehrere verteilt zu halten. AWS-Konten Weitere Informationen finden Sie unter AWS Firewall Manager im AWS WAF -Entwicklerhandbuch.
- Sie können dieselbe Web-ACL für mehrere Amplify-Apps in derselben AWS-Konto verwenden. Alle Apps müssen sich in derselben Region befinden.
- Wenn Sie einer Amplify-App eine Web-ACL zuordnen, wird die Web-ACL standardmäßig an jeden Zweig in der App angehängt. Wenn Sie neue Zweige erstellen, erhalten diese die Web-ACL.
- Wenn Sie einer Amplify-App eine Web-ACL zuordnen, wird sie automatisch mit allen Domains der App verknüpft. Sie können jedoch Regeln konfigurieren, die für einen einzelnen Domainnamen gelten, mithilfe von Host-Header-Abgleichsregeln.
- Sie können keine Web-ACL löschen, die mit einer Amplify-App verknüpft ist. Bevor Sie eine Web-ACL in der AWS WAF Konsole löschen, müssen Sie sie von der App trennen.

Amplify Sie die Web-ACL-Ressourcenrichtlinie

Damit Amplify auf Ihre Web-ACL zugreifen kann, wird der Web-ACL während der Zuordnung eine Ressourcenrichtlinie beigefügt. Amplify erstellt diese Ressourcenrichtlinie automatisch, Sie können sie jedoch mithilfe der AWS WAFV2 GetPermissionPolicyAPI anzeigen. Die folgenden IAM-Berechtigungen sind erforderlich, um einer Amplify-App eine Web-ACL zuzuordnen.

- verstärken: ACL AssociateWeb
- wafv2: ACL AssociateWeb
- WAFv2: PutPermissionPolicy
- WAF 2: GetPermissionPolicy

Firewall-Preise für Amplify-Anwendungen

Die Kosten für die Implementierung AWS WAF in einer Amplify-Anwendung werden auf der Grundlage der folgenden zwei Komponenten berechnet:

- AWS WAF Nutzung — Ihre AWS WAF Nutzung wird Ihnen gemäß dem AWS WAF Preismodell in Rechnung gestellt. AWS WAF Die Gebühren basieren auf den von Ihnen erstellten Web-Zugriffskontrolllisten (Web ACLs), der Anzahl der Regeln, die Sie pro Web-ACL hinzufügen, und der Anzahl der Webanfragen, die Sie erhalten. Details zu den Preisen finden Sie unter AWS WAF - Preise.
- Kosten für Amplify Amplify-Hosting-Integration — Es fallen Gebühren in Höhe von 15,00 USD pro Monat und App an, wenn Sie eine Web-ACL an eine Amplify-Anwendung anhängen. Dies wird anteilig stündlich berechnet.

Bereitstellungen mit Funktionsverzweigungen und Team-Workflows

Amplify Hosting wurde für die Zusammenarbeit mit Feature Branch und GitFlow Workflows entwickelt. Amplify verwendet Git-Branches, um jedes Mal, wenn Sie einen neuen Branch in Ihrem Repository verbinden, ein neues Deployment zu erstellen. Nachdem Sie Ihren ersten Branch verbunden haben, erstellen Sie weitere Feature-Branches.

Um einer App einen Zweig hinzuzufügen

1. Wählen Sie die App aus, zu der Sie einen Zweig hinzufügen möchten.
2. Wählen Sie App-Einstellungen und dann Filialeinstellungen.
3. Wählen Sie auf der Seite mit den Filialeinstellungen die Option Filiale hinzufügen aus.
4. Wählen Sie einen Branch aus Ihrem Repository aus.
5. Wählen Sie Zweig hinzufügen.
6. Stellen Sie Ihre App erneut bereit.

Nachdem Sie einen Branch hinzugefügt haben, sind für Ihre App zwei Bereitstellungen in den Amplify-Standarddomänen verfügbar, z. B. https://main.appid.amplifyapp.comund. https://dev.appid.amplifyapp.com Dies kann davon abweichen team-to-team, aber in der Regel verfolgt der Hauptzweig den Veröffentlichungscode und ist Ihr Produktionszweig. Die develop-Verzweigung wird als Integrationsverzweigung zum Testen von neuen Funktionen verwendet. Auf diese Weise können Betatester unveröffentlichte Funktionen in der Bereitstellung in der Entwicklungsabteilung testen, ohne dass sich dies auf die Endanwender in der Hauptniederlassung auswirkt.

Themen

Team-Workflows mit Full-Stack-Apps von Amplify Gen 2

AWS Amplify Gen 2 bietet ein TypeScript basiertes Entwicklererlebnis, bei dem der Code an erster Stelle steht, um Backends zu definieren. Weitere Informationen zu Fullstack-Workflows mit Amplify Gen 2-Anwendungen finden Sie unter Fullstack-Workflows in den Amplify-Dokumenten.

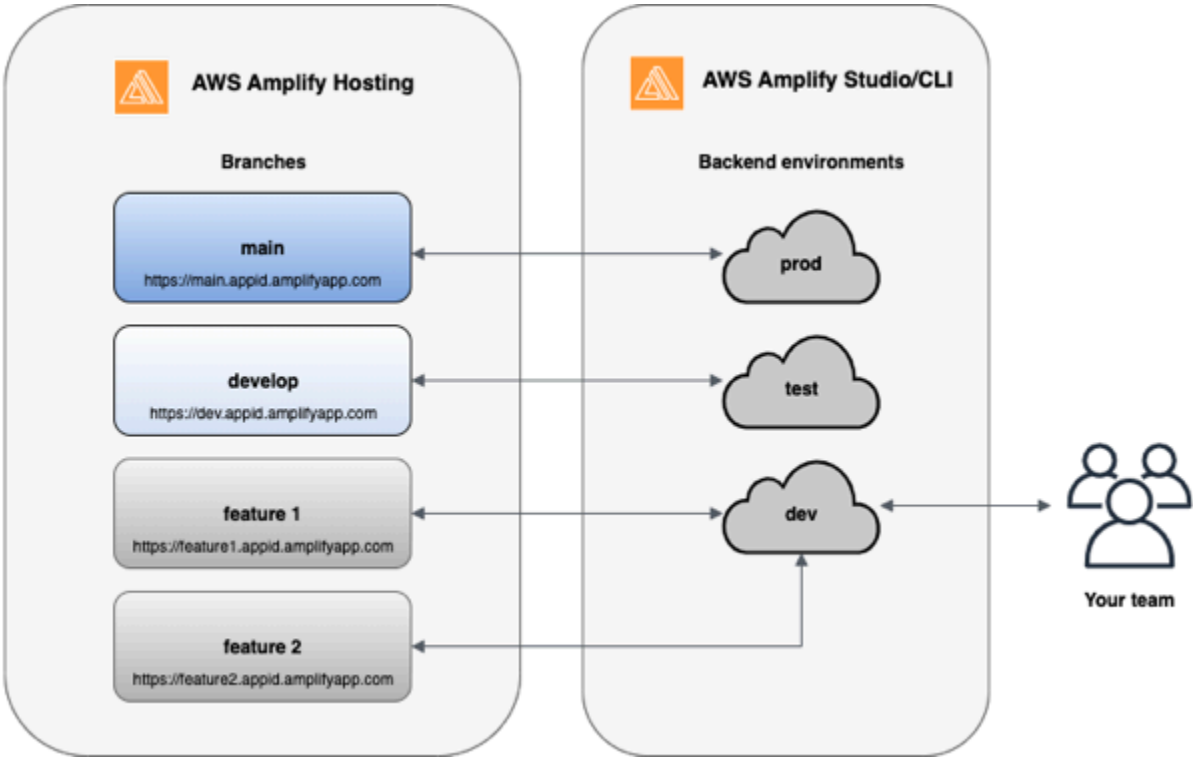
Team-Workflows mit Fullstack Amplify Gen 1-Apps

Eine Feature-Branch-Bereitstellung besteht aus einem Frontend und einer optionalen Backend-Umgebung. Das Frontend wird erstellt und in einem globalen Content Delivery Network (CDN) bereitgestellt, während das Backend von Amplify Studio oder der Amplify CLI bereitgestellt wird. AWS Informationen zur Einrichtung dieses Bereitstellungsszenarios finden Sie unter. Ein Backend für eine Anwendung erstellen

Amplify Hosting stellt kontinuierlich Backend-Ressourcen wie GraphQL APIs - und Lambda-Funktionen mit Ihren Feature Branch-Bereitstellungen bereit. Sie können die folgenden Verzweigungsmodelle verwenden, um Ihr Backend und Frontend mit Amplify Hosting bereitzustellen.

Workflow für Funktionsverzweigungen

- Erstellen Sie mit Amplify Studio oder der Amplify CLI Backend-Umgebungen für Produktion, Test und Entwicklung.
- Ordnen Sie das Prod-Backend dem Hauptzweig zu.
- Ordnen Sie das Test-Backend dem Entwicklungszweig zu.
- Teammitglieder können die Dev-Backend-Umgebung zum Testen einzelner Feature-Branches verwenden.



1. Installieren Sie die Amplify-CLI, um ein neues Amplify-Projekt zu starten.

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

2. Initialisieren Sie eine prod-Backend-Umgebung für Ihr Projekt. Wenn Sie kein Projekt haben, erstellen Sie eines mit Bootstrap-Tools wie create-react-app oder 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. Fügen Sie test- und dev-Backend-Umgebungen hinzu.

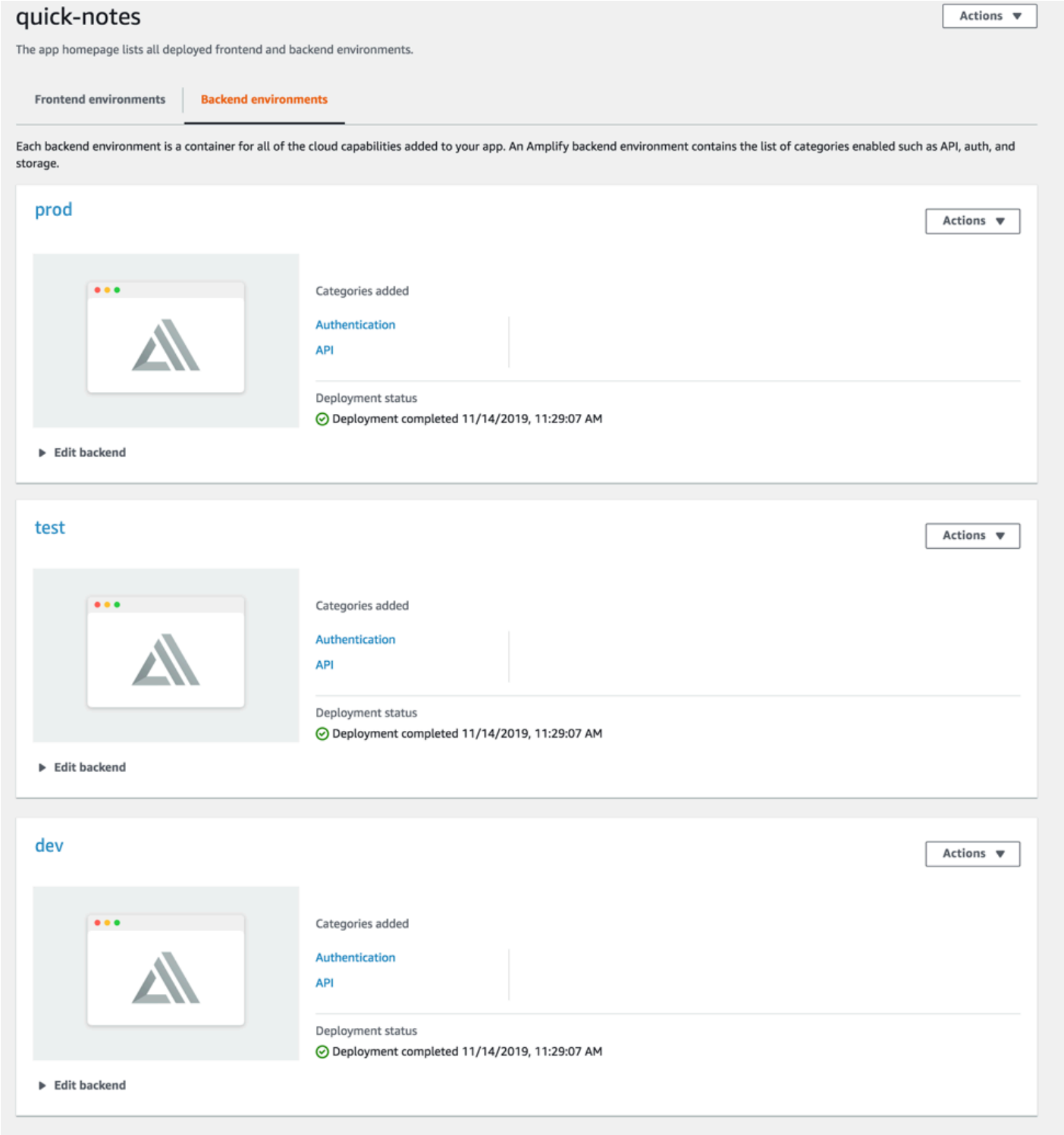
```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: test
...
amplify push
```

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

4. Push Code in ein Git-Repository deiner Wahl (in diesem Beispiel gehen wir davon aus, dass du auf Main gepusht hast).

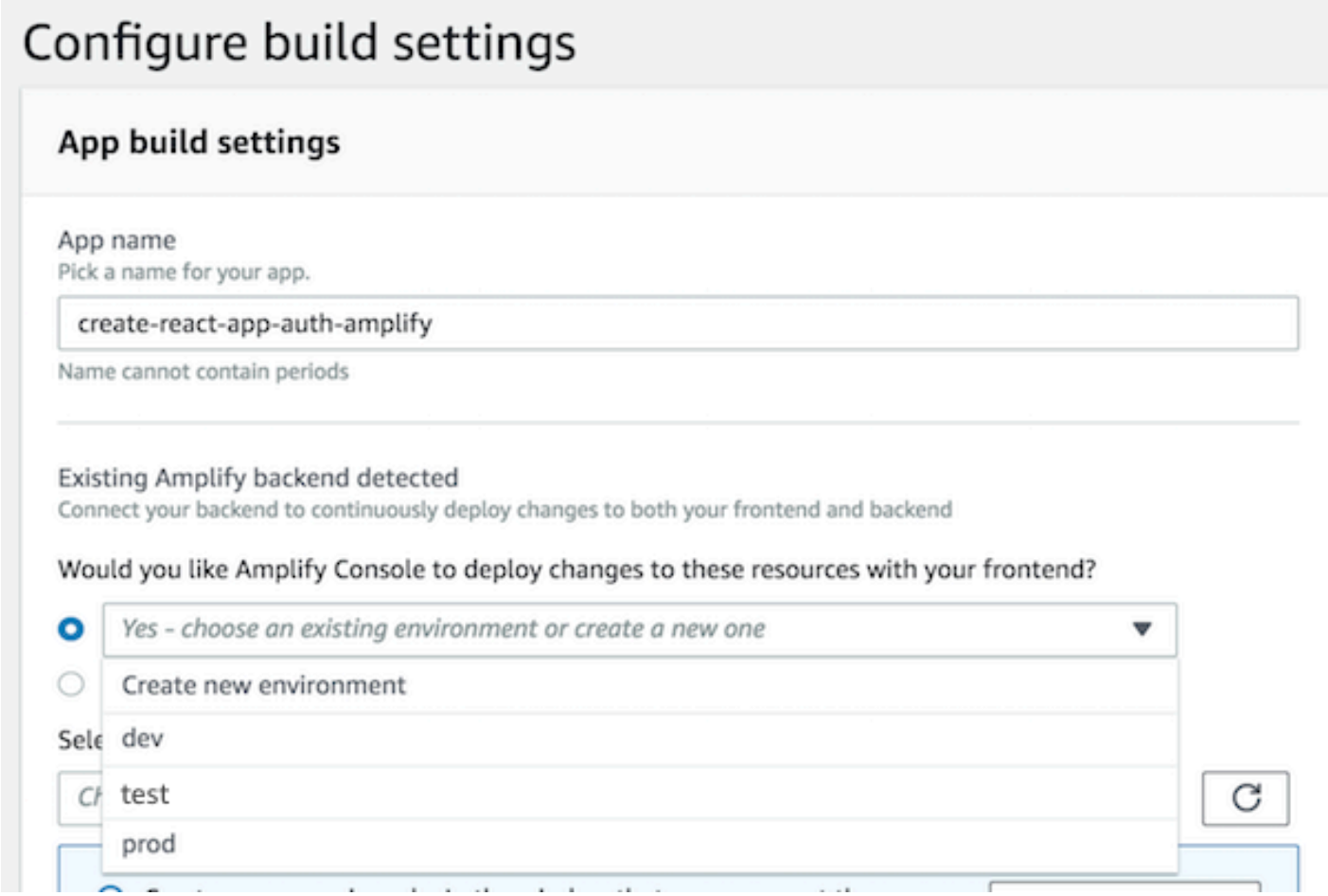
```
git commit -am 'Added dev, test, and prod environments'
git push origin main
```

5. Besuchen Sie Amplify in der AWS-Managementkonsole , um Ihre aktuelle Backend-Umgebung zu sehen. Navigieren Sie vom Breadcrumb aus eine Ebene nach oben, um eine Liste aller Backend-Umgebungen anzuzeigen, die auf der Registerkarte Backend-Umgebungen erstellt wurden.

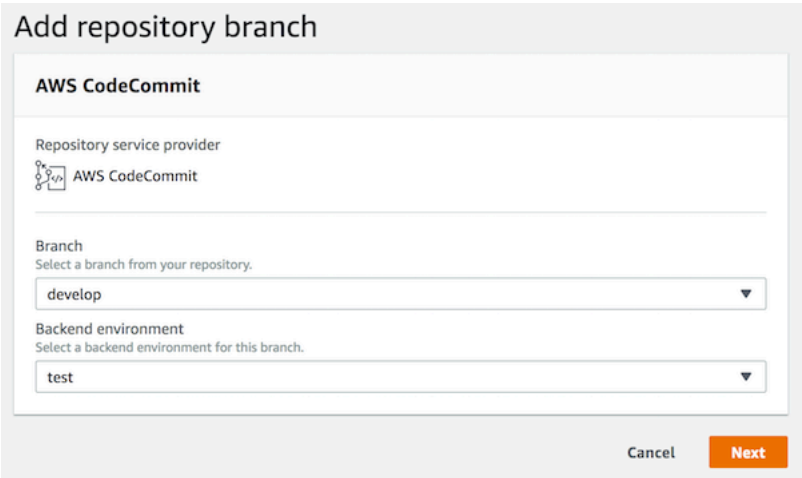


6. Wechseln Sie zur Registerkarte Frontend-Umgebungen und verbinden Sie Ihren Repository-Anbieter und Ihren Hauptzweig.
7. Wählen Sie auf der Seite mit den Build-Einstellungen eine bestehende Backend-Umgebung aus, um eine kontinuierliche Bereitstellung mit dem Hauptzweig einzurichten. Wählen Sie prod

aus der Liste aus und weisen Sie Amplify die Servicerolle zu. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus. Nach Abschluss des Builds erhalten Sie eine Bereitstellung in der Hauptabteilung, die unter verfügbar ist. https://main.appid.amplifyapp.com



8. Connect den Entwicklungszweig in Amplify (gehen Sie davon aus, dass Develop und Main Branch zu diesem Zeitpunkt identisch sind). Wählen Sie die test-Backend-Umgebung aus.



9. Amplify ist jetzt eingerichtet. Sie können jetzt neue Funktionen in einer Funktionsverzweigung verwenden. Fügen Sie Backend-Funktionen mithilfe der dev-Backend-Umgebung aus Ihrer lokalen Workstation hinzu.

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

10. Wenn die Arbeit mit der Funktion abgeschlossen ist, führen Sie einen Commit für Ihren Code durch und erstellen Sie eine Pull-Anforderung zur internen Überprüfung.

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

11. Um eine Vorschau der Änderungen anzuzeigen, rufen Sie die Amplify-Konsole auf und verbinden Sie Ihren Feature-Branch. Hinweis: Wenn Sie das auf Ihrem System AWS CLI installiert haben (nicht die Amplify-CLI), können Sie eine Filiale direkt von Ihrem Terminal aus verbinden. Um nach Ihrer appid zu suchen, gehen Sie zu App settings > General > AppARN (App-Einstellungen > Allgemein > AppARN: arn:aws:amplify:<region>:<region>:apps/<appid>

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

12. Auf Ihre Funktion können Sie unter zugreifen https://newinternet.appid.amplifyapp.com, um sie mit Ihren Teamkollegen zu teilen. Wenn alles korrekt ist, führen Sie die Pull-Anforderung mit der develop-Verzweigung zusammen.

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

13. Damit wird ein Build gestartet, der sowohl das Backend als auch das Frontend in Amplify mit einer Filialbereitstellung unter aktualisiert. https://dev.appid.amplifyapp.com Sie können diesen Link für interne Beteiligte freigeben, damit sie die neue Funktion überprüfen können.

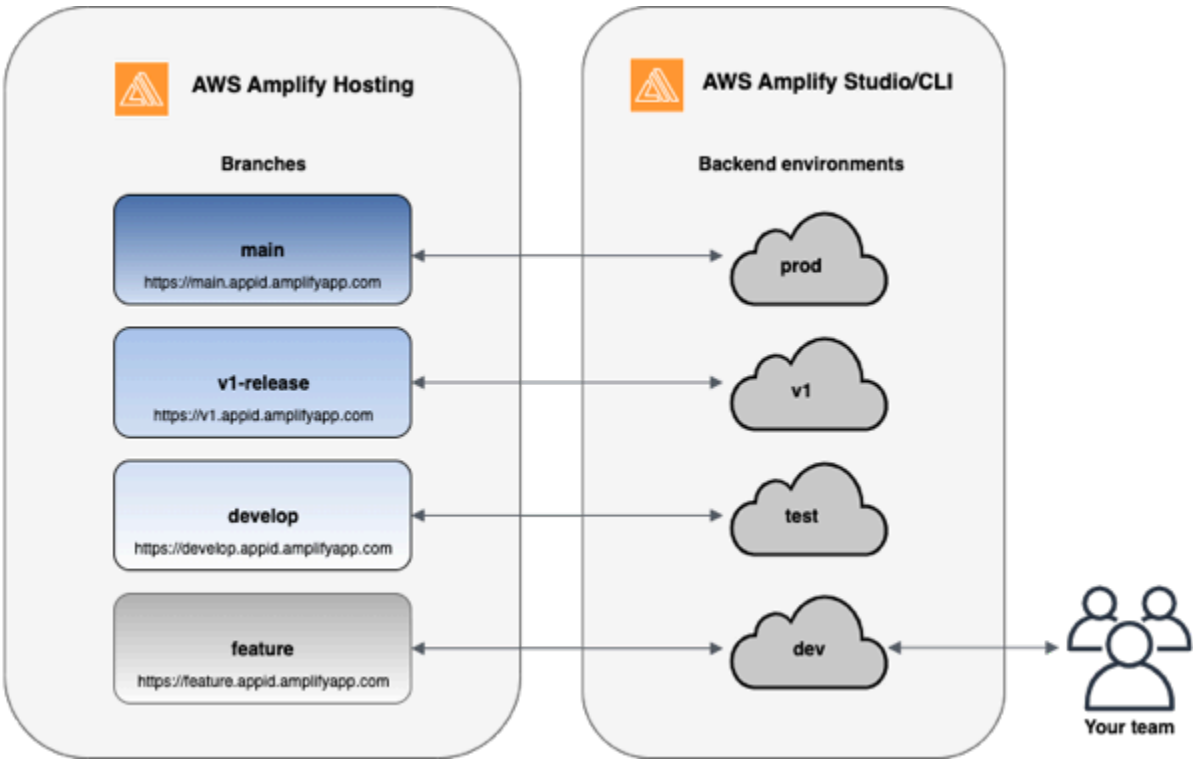
14. Löschen Sie Ihren Feature-Branch aus Git, Amplify und entfernen Sie die Backend-Umgebung aus der Cloud (Sie können jederzeit eine neue einrichten, indem Sie „amplify env checkout prod“ ausführen und „amplify env add“ ausführen).

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

GitFlow Arbeitsablauf

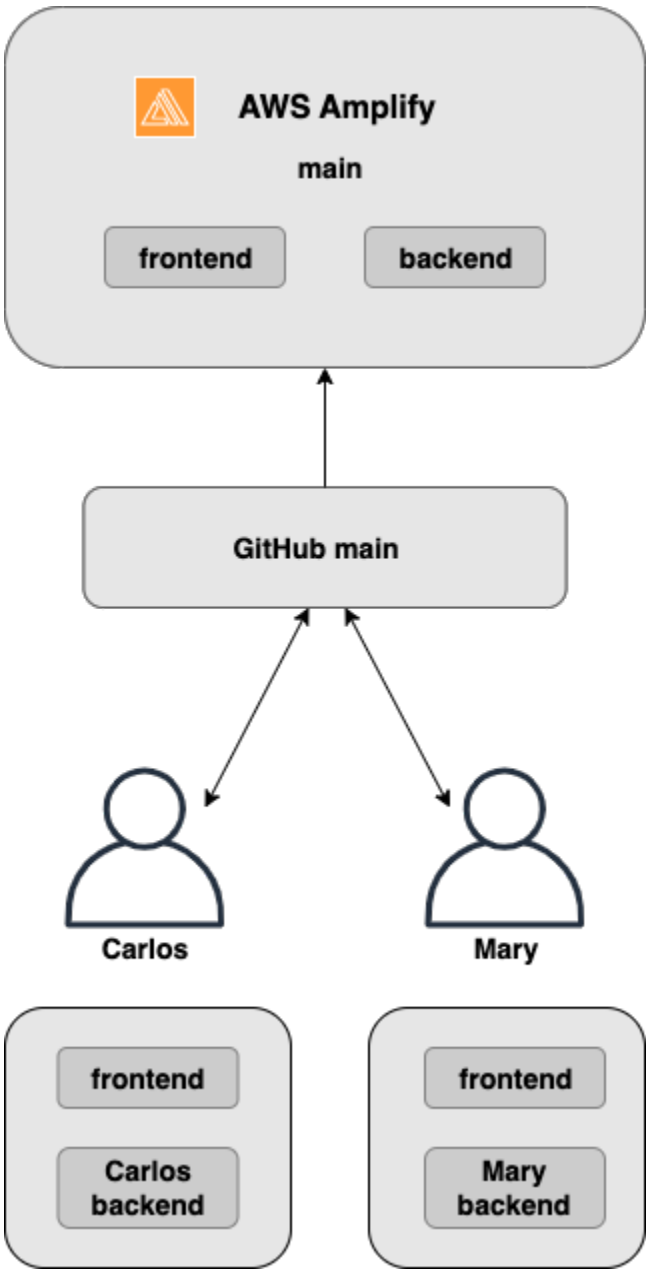
GitFlow verwendet zwei Zweige, um den Verlauf des Projekts aufzuzeichnen. Der Hauptzweig verfolgt nur den Release-Code, und der Entwicklungszweig wird als Integrationszweig für neue Funktionen verwendet. GitFlow vereinfacht die parallel Entwicklung, indem neue Entwicklungen von abgeschlossenen Arbeiten isoliert werden. Neue Entwicklungen (z. B. Funktionen und normale Fehlerbehebungen) erfolgen in feature-Verzweigungen. Wenn der Entwickler den Code zur Veröffentlichung freigeben möchte, wird die feature-Verzweigung wieder mit der develop-Verzweigung zur Integration zusammengeführt. Die einzigen Commits für den Hauptzweig sind Zusammenführungen von Release-Branches und Hotfix-Branches (um Fehler im Notfall zu beheben).

Das folgende Diagramm zeigt ein empfohlenes Setup mit. GitFlow Führen Sie einfach die im obigen Abschnitt „Workflow für Funktionsverzweigungen“ beschriebenen Schritte aus.



Sandbox pro Entwickler

- Jeder Entwickler in einem Team erstellt eine Sandbox-Umgebung in der Cloud, die vom lokalen Computer unabhängig ist. Auf diese Weise können Entwickler isoliert voneinander arbeiten, ohne die Änderungen anderer Teammitglieder zu überschreiben.
- Jeder Zweig in Amplify hat sein eigenes Backend. Dadurch wird sichergestellt, dass Amplify das Git-Repository als zentrale Informationsquelle für die Implementierung von Änderungen verwendet, anstatt sich darauf zu verlassen, dass Entwickler im Team ihr Backend oder Frontend manuell von ihren lokalen Computern aus in die Produktion bringen.



1. Installieren Sie die Amplify-CLI, um ein neues Amplify-Projekt zu starten.

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

2. Initialisieren Sie eine Mary-Backend-Umgebung für Ihr Projekt. Wenn Sie kein Projekt haben, erstellen Sie eines mit Bootstrap-Tools wie create-react-app oder 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. Push Code in ein Git-Repository deiner Wahl (in diesem Beispiel gehen wir davon aus, dass du auf Main gepusht hast).

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

4. Connect dein Repo > Main mit Amplify.
5. Die Amplify-Konsole erkennt Backend-Umgebungen, die von der Amplify-CLI erstellt wurden. Wählen Sie in der Dropdownliste die Option Neue Umgebung erstellen aus und weisen Sie Amplify die Servicerolle zu. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus. Nach Abschluss des Builds erhalten Sie eine Bereitstellung in der Hauptzweige https://main.appid.amplifyapp.commit einer neuen Backend-Umgebung, die mit der Filiale verknüpft ist.
6. Connect den Entwicklungszweig in Amplify (gehen Sie davon aus, dass Develop und Main Branch zu diesem Zeitpunkt identisch sind) und wählen Sie Create

Musterbasierte Feature-Branch-Bereitstellungen

Mit musterbasierten Branch-Bereitstellungen können Sie automatisch Branches, die einem bestimmten Muster entsprechen, für Amplify bereitstellen. Produktteams, die Feature Branch oder GitFlow Workflows für ihre Releases verwenden, können jetzt Muster definieren, **`release**`** um beispielsweise Git-Branches, die mit „release“ beginnen, automatisch auf einer gemeinsam nutzbaren URL bereitzustellen.

1. Wähle „App-Einstellungen“ und dann „Branch-Einstellungen“.
2. Wählen Sie auf der Seite mit den Filialeinstellungen die Option Bearbeiten aus.
3. Wählen Sie Automatische Branch-Erkennung, um Zweige, die einem Mustersatz entsprechen, automatisch mit Amplify zu verbinden.
4. Geben Sie im Feld Automatische Branch-Erkennung — Muster die Muster für die automatische Bereitstellung von Zweigen ein.
   - **`*`**— Stellt alle Branches in Ihrem Repository bereit.
   - **`release*`**— Stellt alle Branches bereit, die mit dem Wort „release“ beginnen.
   - **`release*/`**— Stellt alle Branches bereit, die einem 'release /'-Muster entsprechen.

- Geben Sie mehrere Muster in einer durch Kommas getrennten Liste an. Beispiel, **`release*, feature*`**.

5. Richten Sie den automatischen Kennwortschutz für alle Zweige ein, die automatisch erstellt werden, indem Sie die Zugriffskontrolle für automatische Branches auswählen.
6. Für Gen-1-Anwendungen, die mit einem Amplify-Backend erstellt wurden, können Sie wählen, ob Sie für jede verbundene Filiale eine neue Umgebung erstellen oder alle Zweige auf ein vorhandenes Backend verweisen möchten.
7. Wählen Sie Speichern.

Musterbasierte Feature-Branch-Bereitstellungen für eine App, die mit einer benutzerdefinierten Domain verbunden ist

Sie können musterbasierte Feature-Branch-Bereitstellungen für eine App verwenden, die mit einer benutzerdefinierten Amazon Route 53-Domain verbunden ist.

- Anweisungen zur Einrichtung musterbasierter Feature-Branch-Bereitstellungen finden Sie unter Automatische Subdomains für eine benutzerdefinierte Amazon Route 53-Domain einrichten
- Anweisungen zum Verbinden einer Amplify-App mit einer benutzerdefinierten Domain, die in Route 53 verwaltet wird, finden Sie unter Hinzufügen einer von Amazon Route 53 verwalteten benutzerdefinierten Domain
- Weitere Informationen zur Verwendung von Route 53 finden Sie unter Was ist Amazon Route 53

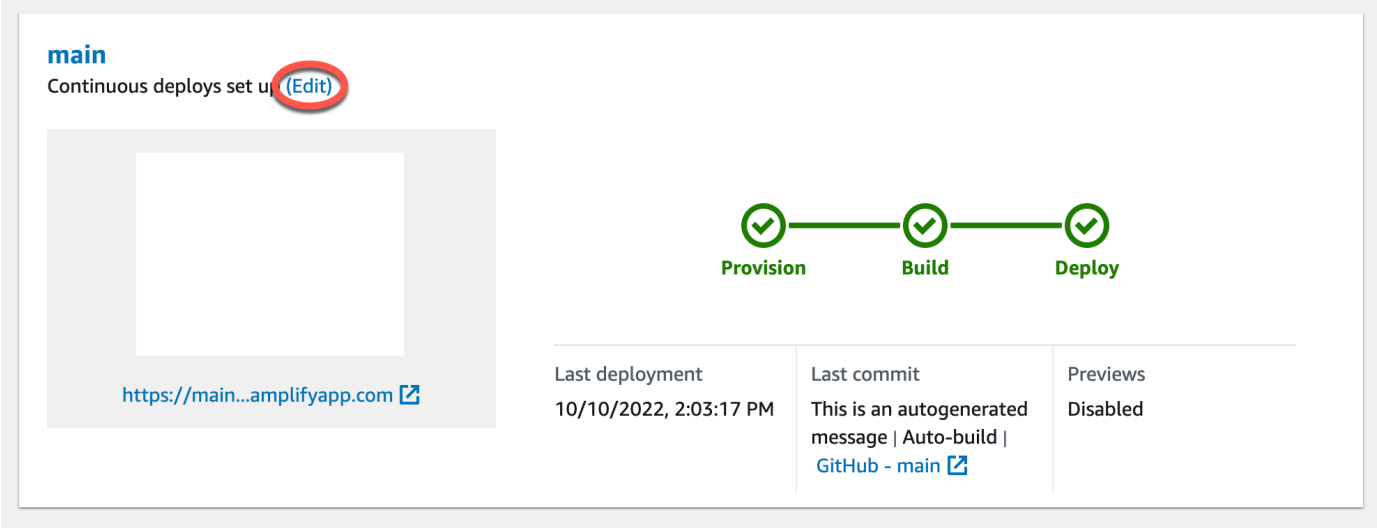
Automatische Generierung der Amplify-Konfiguration während der Erstellung (nur Gen 1-Apps)

> **Note**
>
> Die Informationen in diesem Abschnitt gelten nur für Apps der 1. Generation. Wenn Sie automatisch Infrastruktur- und Anwendungscodeänderungen aus Feature-Branches für eine Gen-2-App bereitstellen möchten, finden Sie in den Amplify-Dokumenten weitere Informationen unter Fullstack-Branch-Bereitstellungen.

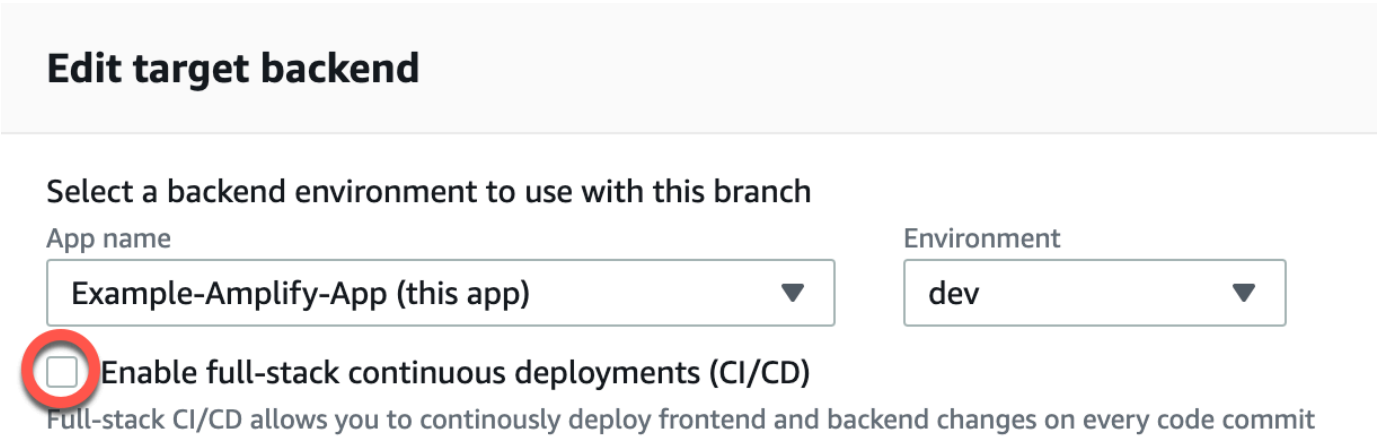
Amplify unterstützt die automatische Build-Time-Generierung der `aws-exports.js` Amplify-Konfigurationsdatei für Gen-1-Apps. Indem Sie Full-Stack-CI/CD-Bereitstellungen deaktivieren, ermöglichen Sie Ihrer App, die `aws-exports.js` Datei automatisch zu generieren und sicherzustellen, dass zur Build-Zeit keine Aktualisierungen an Ihrem Backend vorgenommen werden.

Zur automatischen Generierung während der Erstellung **`aws-exports.js`**

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, die Sie bearbeiten möchten.
3. Wählen Sie den Tab Hosting-Umgebungen.
4. Suchen Sie den Zweig, den Sie bearbeiten möchten, und wählen Sie Bearbeiten.



5. Deaktivieren Sie auf der Seite Ziel-Backend bearbeiten die Option Enable Full-Stack Continuous Deployments (CI/CD), um Full-Stack-CI/CD für dieses Backend zu deaktivieren.



6. Wählen Sie eine bestehende Servicerolle aus, um Amplify die erforderlichen Berechtigungen zu erteilen, um Änderungen an Ihrem App-Backend vorzunehmen. Wenn Sie eine Servicerolle erstellen müssen, wählen Sie Neue Rolle erstellen. Weitere Informationen zum Erstellen einer Servicerolle finden Sie unter Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen.
7. Wählen Sie Speichern. Amplify wendet diese Änderungen an, wenn Sie die App das nächste Mal erstellen.

Bedingte Backend-Builds (nur Apps der Generation 1)

> **Note**
>
> Die Informationen in diesem Abschnitt gelten nur für Apps der Generation 1. Amplify Gen 2 bietet ein TypeScript basiertes Entwicklererlebnis, bei dem der Code an erster Stelle steht. Daher ist diese Funktion für Backends der zweiten Generation nicht erforderlich.

Amplify unterstützt bedingte Backend-Builds für alle Branches in einer Gen-1-App. Um bedingte Backend-Builds zu konfigurieren, setzen Sie die `AMPLIFY_DIFF_BACKEND` Umgebungsvariable auf. `true` Durch die Aktivierung bedingter Backend-Builds können Builds beschleunigt werden, bei denen Änderungen nur am Frontend vorgenommen werden.

Wenn Sie diff-basierte Backend-Builds aktivieren, versucht Amplify zu Beginn jedes Builds, einen Diff für den `amplify` Ordner in Ihrem Repository auszuführen. Wenn Amplify keine Unterschiede feststellt, überspringt es den Backend-Build-Schritt und aktualisiert Ihre Backend-Ressourcen nicht. Wenn Ihr Projekt keinen `amplify` Ordner in Ihrem Repository hat, ignoriert Amplify den Wert der `AMPLIFY_DIFF_BACKEND` Umgebungsvariablen. Anweisungen zum Einstellen der `AMPLIFY_DIFF_BACKEND` Umgebungsvariablen finden Sie unter. Konfiguration von diff-basierten Backend-Builds für eine Gen-1-App

Wenn Sie derzeit benutzerdefinierte Befehle in den Build-Einstellungen Ihrer Backend-Phase angegeben haben, funktionieren bedingte Backend-Builds nicht. Wenn Sie möchten, dass diese benutzerdefinierten Befehle ausgeführt werden, müssen Sie sie in der Datei Ihrer App in die Frontend-Phase Ihrer Build-Einstellungen verschieben. `amplify.yml` Weitere Informationen zum Aktualisieren der `amplify.yml` Datei finden Sie unterReferenz zur Build-Spezifikation.

Verwenden Sie Amplify-Backends für mehrere Apps (nur Gen-1-Apps)

> **Note**
>
> Die Informationen in diesem Abschnitt gelten nur für Apps der 1. Generation. Wenn Sie Backend-Ressourcen für eine Gen 2-App gemeinsam nutzen möchten, finden Sie in den Amplify-Dokumenten weitere Informationen unter Ressourcen branchenübergreifend teilen.

Mit Amplify können Sie bestehende Backend-Umgebungen für all Ihre Gen 1-Apps in einer bestimmten Region wiederverwenden. Sie können dies tun, wenn Sie eine neue App erstellen, einen neuen Zweig mit einer vorhandenen App verbinden oder ein vorhandenes Frontend so aktualisieren, dass es auf eine andere Backend-Umgebung verweist.

Backends wiederverwenden, wenn Sie eine neue App erstellen

Um ein Backend wiederzuverwenden, wenn Sie eine neue Amplify-App erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Gehen Sie wie folgt vor, um ein neues Backend für dieses Beispiel zu erstellen:
   a. Wählen Sie im Navigationsbereich Alle Apps aus.
   b. Wählen Sie Neue App, App erstellen aus.
   c. Geben Sie einen Namen für Ihre App ein, z. **`Example-Amplify-App`** B.
   d. Wählen Sie Bereitstellung bestätigen aus.
3. Um ein Frontend mit Ihrem neuen Backend zu verbinden, wählen Sie den Tab Hosting-Umgebungen.
4. Wähle deinen Git-Anbieter und dann Connect branch.
5. Wähle auf der Seite Repository-Zweig hinzufügen für Kürzlich aktualisierte Repositorys deinen Repository-Namen aus. Wählen Sie für Branch den Branch aus Ihrem Repository aus, zu dem Sie eine Verbindung herstellen möchten.
6. Gehen Sie auf der Seite Build-Einstellungen wie folgt vor:

a. Wählen Sie unter App-Name die App aus, die zum Hinzufügen einer Backend-Umgebung verwendet werden soll. Sie können die aktuelle App oder eine andere App in der aktuellen Region auswählen.

b. Wählen Sie unter Umgebung den Namen der Backend-Umgebung aus, die Sie hinzufügen möchten. Sie können eine bestehende Umgebung verwenden oder eine neue erstellen.

c. Standardmäßig CI/CD ist Full-Stack ausgeschaltet. Wenn Sie Full-Stack-CI/CD ausschalten, wird die App nur im Pull-Only-Modus ausgeführt. Zur Build-Zeit generiert Amplify automatisch nur die `aws-exports.js` Datei, ohne Ihre Backend-Umgebung zu ändern.

d. Wählen Sie eine bestehende Servicerolle aus, um Amplify die erforderlichen Berechtigungen zu erteilen, um Änderungen an Ihrem App-Backend vorzunehmen. Wenn Sie eine Servicerolle erstellen müssen, wählen Sie Neue Rolle erstellen. Weitere Informationen zum Erstellen einer Servicerolle finden Sie unter Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen.

e. Wählen Sie Weiter aus.

7. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus.

Verwenden Sie Backends wieder, wenn Sie eine Filiale mit einer vorhandenen App verbinden

Um ein Backend wiederzuverwenden, wenn Sie eine Filiale mit einer vorhandenen Amplify-App verbinden

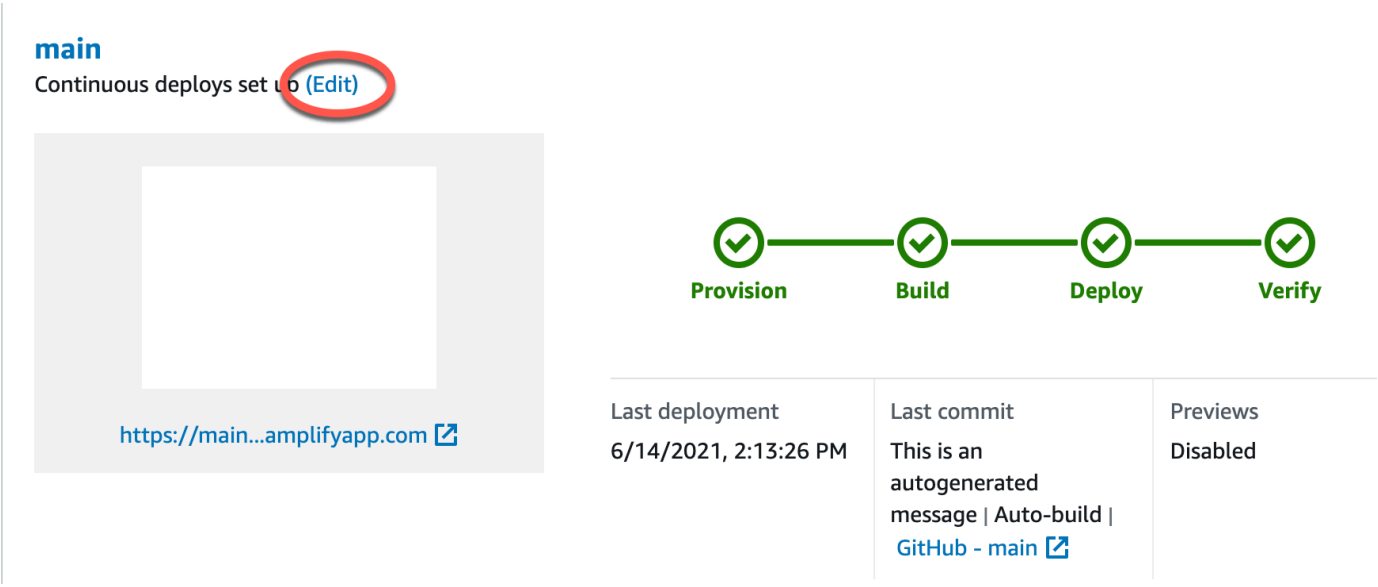
1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, mit der Sie eine neue Filiale verbinden möchten.
3. Wählen Sie im Navigationsbereich App-Einstellungen, Allgemein aus.
4. Wählen Sie im Abschnitt Filialen die Option Filiale Connect aus.
5. Wählen Sie auf der Seite Repository-Zweig hinzufügen für Branch den Branch aus Ihrem Repository aus, zu dem Sie eine Verbindung herstellen möchten.
6. Wählen Sie als App-Name die App aus, die zum Hinzufügen einer Backend-Umgebung verwendet werden soll. Sie können die aktuelle App oder eine andere App in der aktuellen Region auswählen.
7. Wählen Sie unter Umgebung den Namen der Backend-Umgebung aus, die Sie hinzufügen möchten. Sie können eine bestehende Umgebung verwenden oder eine neue erstellen.

8. Wenn Sie eine Servicerolle einrichten müssen, um Amplify die erforderlichen Berechtigungen zu erteilen, um Änderungen an Ihrem App-Backend vorzunehmen, werden Sie von der Konsole aufgefordert, diese Aufgabe auszuführen. Weitere Informationen zum Erstellen einer Servicerolle finden Sie unter Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen.
9. Standardmäßig ist Full-Stack ausgeschaltet CI/CD . Wenn Sie Full-Stack deaktivieren CI/CD , wird die App nur im Pull-Only-Modus ausgeführt. Zur Build-Zeit generiert Amplify automatisch nur die `aws-exports.js` Datei, ohne Ihre Backend-Umgebung zu ändern.
10. Wählen Sie Weiter aus.
11. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus.

Bearbeiten Sie ein vorhandenes Frontend so, dass es auf ein anderes Backend verweist

Um eine Frontend-Amplify-App so zu bearbeiten, dass sie auf ein anderes Backend verweist

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie das Backend bearbeiten möchten.
3. Wählen Sie den Tab Hosting-Umgebungen.
4. Suchen Sie den Zweig, den Sie bearbeiten möchten, und wählen Sie Bearbeiten.



5. Wählen Sie auf der Seite Wählen Sie eine Backend-Umgebung aus, die mit diesem Zweig verwendet werden soll, als App-Name die Frontend-App aus, für die Sie die Backend-Umgebung

bearbeiten möchten. Sie können die aktuelle App oder eine andere App in der aktuellen Region auswählen.

6. Wählen Sie unter Backend-Umgebung den Namen der Backend-Umgebung aus, die hinzugefügt werden soll.
7. Standardmäßig ist Full-Stack CI/CD aktiviert. Deaktivieren Sie diese Option, um Full-Stack CI/CD für dieses Backend zu deaktivieren. Wenn Sie Full-Stack deaktivieren, wird CI/CD die App nur im Pull-Only-Modus ausgeführt. Zur Build-Zeit generiert Amplify automatisch nur die `aws-exports.js` Datei, ohne die Backend-Umgebung zu ändern.
8. Wählen Sie Speichern. Amplify wendet diese Änderungen an, wenn Sie die App das nächste Mal erstellen.

Ein Backend für eine Anwendung erstellen

Mit können AWS Amplify Sie eine Fullstack-Anwendung mit Daten-, Authentifizierungs-, Speicher- und Frontend-Hosting erstellen, für die Sie bereitgestellt werden. AWS

AWS Amplify Gen 2 bietet ein TypeScript basiertes Entwicklererlebnis, bei dem der Code an erster Stelle steht, um Backends zu definieren. Um zu erfahren, wie Sie Amplify Gen 2 verwenden, um ein Backend zu erstellen und mit Ihrer App zu verbinden, siehe Backend erstellen und verbinden in den Amplify-Dokumenten.

Wenn Sie nach der Dokumentation zum Erstellen eines Backends für eine Gen 1-App mithilfe der CLI und Amplify Studio suchen, finden Sie in den Gen 1 Amplify-Dokumenten unter Build & Connect Backend.

Themen

Erstellen Sie ein Backend für eine Gen 2-App

Ein Tutorial, das Sie durch die Schritte zur Erstellung einer Amplify Gen 2-Fullstack-Anwendung mit einem TypeScript basierten Backend führt, finden Sie unter Erste Schritte in den Amplify-Dokumenten.

Erstellen Sie ein Backend für eine Gen 1-App

In diesem Tutorial richten Sie einen CI/CD Fullstack-Workflow mit Amplify ein. Sie werden eine Frontend-App für Amplify Hosting bereitstellen. Anschließend erstellen Sie mit Amplify Studio ein Backend. Schließlich verbinden Sie das Cloud-Backend mit der Frontend-App.

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Voraussetzungen erfüllen.

Melden Sie sich an für eine AWS-Konto

Wenn Sie noch kein AWS Kunde sind, müssen Sie einen erstellen, AWS-Konto indem Sie den Online-Anweisungen folgen. Durch die Registrierung können Sie auf Amplify und andere AWS Dienste zugreifen, die Sie mit Ihrer Anwendung verwenden können.

Erstellen Sie ein Git-Repository

Amplify unterstützt GitHub, Bitbucket GitLab, und. AWS CodeCommit Schieben Sie Ihre Anwendung in Ihr Git-Repository.

Installieren Sie die Amplify Command Line Interface (CLI)

Anweisungen finden Sie unter Installieren der Amplify CLI in der Amplify Framework-Dokumentation.

Schritt 1: Stellen Sie ein Frontend bereit

Wenn Sie eine bestehende Frontend-App in einem Git-Repository haben, die Sie für dieses Beispiel verwenden möchten, können Sie mit den Anweisungen zur Bereitstellung einer Frontend-App fortfahren.

Wenn Sie eine neue Frontend-App für dieses Beispiel erstellen müssen, können Sie den Anweisungen Create React App in der Dokumentation Create React App folgen.

Um eine Frontend-App bereitzustellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite Alle Apps die Option Neue App und dann Web-App hosten in der oberen rechten Ecke aus.
3. Wähle deinen GitHub, Bitbucket- oder AWS CodeCommit Repository-Anbieter aus und wähle dann Weiter. GitLab
4. Amplify autorisiert den Zugriff auf Ihr Git-Repository. Für GitHub Repositorien verwendet Amplify jetzt die GitHub Apps-Funktion, um den Amplify-Zugriff zu autorisieren.

   Weitere Informationen zur Installation und Autorisierung der App finden Sie unter. GitHub Amplify-Zugriff auf Repositorys einrichten GitHub
5. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:

a. Wählen Sie in der Liste „Kürzlich aktualisierte Repositorys“ den Namen des Repositorys aus, mit dem Sie eine Verbindung herstellen möchten.

b. Wählen Sie in der Branch-Liste den Namen des Repository-Branches aus, zu dem Sie eine Verbindung herstellen möchten.

c. Wählen Sie Weiter aus.

6. Wählen Sie auf der Seite Build-Einstellungen konfigurieren die Option Weiter aus.
7. Wählen Sie auf der Seite „Überprüfen“ die Option Speichern und bereitstellen aus. Wenn die Bereitstellung abgeschlossen ist, können Sie Ihre App in der `amplifyapp.com` Standarddomäne anzeigen.

> **Note**
>
> Um die Sicherheit Ihrer Amplify-Anwendungen zu erhöhen, ist die Domain amplifyapp.com in der Public Suffix List (PSL) registriert. Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, falls Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Amplify-Anwendungen einrichten müssen. Diese Vorgehensweise trägt dazu bei, Ihre Domain vor CSRF-Versuchen (Cross-Site Request Forgery Attempts, Anforderungsfälschung zwischen Websites) zu schützen. Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

Schritt 2: Erstellen Sie ein Backend

Nachdem Sie eine Frontend-App für Amplify Hosting bereitgestellt haben, können Sie ein Backend erstellen. Verwenden Sie die folgenden Anweisungen, um ein Backend mit einer einfachen Datenbank und einem GraphQL-API-Endpunkt zu erstellen.

Um ein Backend zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite Alle Apps die App aus, die Sie in Schritt 1 erstellt haben.
3. Wählen Sie auf der Startseite der App den Tab Backend-Umgebungen und dann Erste Schritte aus. Dadurch wird der Einrichtungsprozess für eine Standard-Staging-Umgebung eingeleitet.
4. Wählen Sie nach Abschluss der Einrichtung Launch Studio, um auf die Staging-Backend-Umgebung in Amplify Studio zuzugreifen.

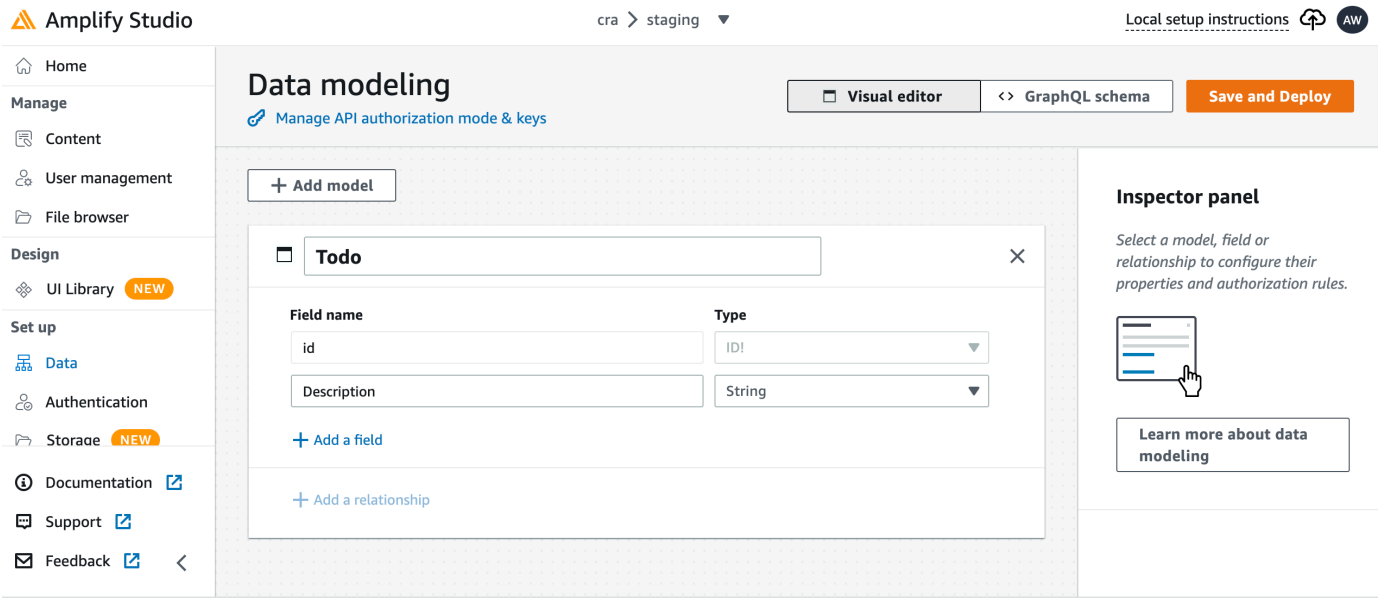
Amplify Studio ist eine visuelle Oberfläche, mit der Sie Ihr Backend erstellen und verwalten und die Entwicklung Ihrer Frontend-Benutzeroberfläche beschleunigen können. Weitere Informationen zu Amplify Studio finden Sie in der Amplify Studio-Dokumentation.

Verwenden Sie die folgenden Anweisungen, um eine einfache Datenbank mit der Visual Backend Builder-Oberfläche von Amplify Studio zu erstellen.

Erstellen Sie ein Datenmodell

1. Wählen Sie auf der Startseite der Staging-Umgebung Ihrer App die Option Datenmodell erstellen aus. Dadurch wird der Datenmodell-Designer geöffnet.
2. Wählen Sie auf der Seite Datenmodellierung die Option Modell hinzufügen aus.
3. Geben Sie für den Titel ein**`Todo`**.
4. Wählen Sie Feld hinzufügen aus.
5. Geben Sie als Feldname ein**`Description`**.

   Der folgende Screenshot ist ein Beispiel dafür, wie Ihr Datenmodell im Designer aussehen wird.



6. Wählen Sie Speichern und Bereitstellen.
7. Kehren Sie zur Amplify Hosting-Konsole zurück und die Bereitstellung der Staging-Umgebung wird fortgesetzt.

Während der Bereitstellung erstellt Amplify Studio alle erforderlichen AWS Ressourcen im Backend, einschließlich einer AWS AppSync GraphQL-API für den Datenzugriff und einer Amazon DynamoDB-

Tabelle zum Hosten der Todo-Elemente. Amplify verwendet CloudFormation , um Ihr Backend bereitzustellen, sodass Sie Ihre Backend-Definition als speichern können. infrastructure-as-code

Schritt 3: Backend mit Frontend Connect

Nachdem Sie nun ein Frontend bereitgestellt und ein Cloud-Backend erstellt haben, das ein Datenmodell enthält, müssen Sie sie verbinden. Verwenden Sie die folgenden Anweisungen, um Ihre Backend-Definition mit der Amplify CLI in Ihr lokales App-Projekt zu übertragen.

Um ein Cloud-Backend mit einem lokalen Frontend zu verbinden

1. Öffnen Sie ein Terminalfenster und navigieren Sie zum Stammverzeichnis Ihres lokalen Projekts.
2. Führen Sie den folgenden Befehl im Terminalfenster aus und ersetzen Sie den roten Text durch die eindeutige App-ID und den Namen der Backend-Umgebung für Ihr Projekt.

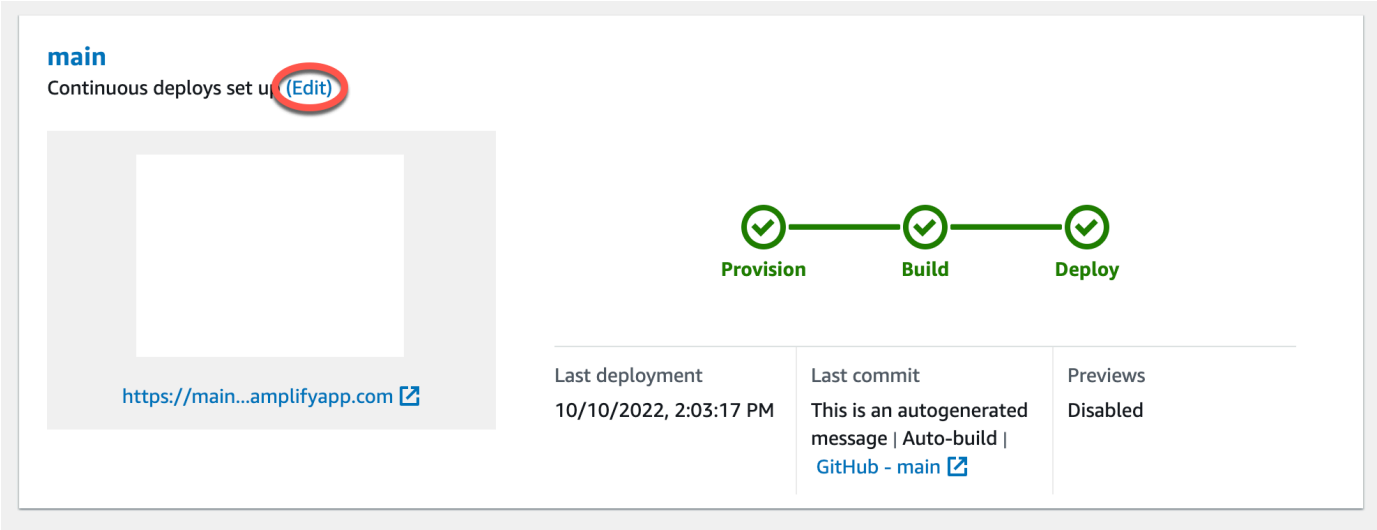
   ```
   amplify pull --appId abcd1234 --envName staging
   ```

3. Folgen Sie den Anweisungen im Terminalfenster, um die Einrichtung des Projekts abzuschließen.

Jetzt können Sie den Build-Prozess so konfigurieren, dass das Backend zum Workflow für die kontinuierliche Bereitstellung hinzugefügt wird. Verwenden Sie die folgenden Anweisungen, um einen Frontend-Zweig mit einem Backend in der Amplify Hosting-Konsole zu verbinden.

Um einen Frontend-App-Branch und ein Cloud-Backend zu verbinden

1. Wählen Sie auf der Startseite der App den Tab Hosting-Umgebungen aus.
2. Suchen Sie den Hauptzweig und wählen Sie Bearbeiten.



3. Wählen Sie im Fenster Ziel-Backend bearbeiten unter Umgebung den Namen des Backends aus, zu dem eine Verbindung hergestellt werden soll. Wählen Sie in diesem Beispiel das Staging-Backend aus, das Sie in Schritt 2 erstellt haben.

   Standardmäßig ist Full-Stack CI/CD aktiviert. Deaktivieren Sie diese Option, um Full-Stack CI/CD für dieses Backend zu deaktivieren. Wenn Sie Full-Stack deaktivieren, wird CI/CD die App nur im Pull-Only-Modus ausgeführt. Zur Build-Zeit generiert Amplify automatisch nur die `aws-exports.js` Datei, ohne Ihre Backend-Umgebung zu ändern.

4. Als Nächstes müssen Sie eine Servicerolle einrichten, um Amplify die erforderlichen Berechtigungen zu erteilen, um Änderungen an Ihrem App-Backend vorzunehmen. Sie können entweder eine bestehende Servicerolle verwenden oder eine neue erstellen. Detaillierte Anweisungen finden Sie unter Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen.

5. Nachdem Sie eine Servicerolle hinzugefügt haben, kehren Sie zum Fenster Ziel-Backend bearbeiten zurück und wählen Sie Speichern.

6. Um die Verbindung des Staging-Backends mit dem Hauptzweig der Frontend-App abzuschließen, führen Sie einen neuen Build Ihres Projekts durch.

   Führen Sie eine der folgenden Aktionen aus:

   - Pushen Sie aus Ihrem Git-Repository Code, um einen Build in der Amplify-Konsole zu initiieren.
   - Navigieren Sie in der Amplify-Konsole zur Seite mit den Build-Details der App und wählen Sie Diese Version erneut bereitstellen.

Nächste Schritte

Richten Sie Feature-Branch-Bereitstellungen ein

Folgen Sie unserem empfohlenen Arbeitsablauf, um Feature-Branch-Bereitstellungen mit mehreren Backend-Umgebungen einzurichten.

Erstellen Sie eine Frontend-Benutzeroberfläche in Amplify Studio

Verwenden Sie Studio, um Ihre Frontend-Benutzeroberfläche mit einer Reihe von ready-to-use UI-Komponenten zu erstellen, und verbinden Sie sie dann mit Ihrem App-Backend. Weitere Informationen und Tutorials finden Sie im Benutzerhandbuch für Amplify Studio in der Amplify Framework-Dokumentation.

Erweiterte Bereitstellungsfunktionen

In diesem Kapitel werden erweiterte Bereitstellungsfunktionen behandelt, die Ihren Amplify Hosting-Workflow verbessern. Diese Funktionen bieten zusätzliche Kontrollen und Funktionen, mit denen Teams Bereitstellungen effektiver verwalten, die Codequalität sicherstellen und die Sicherheit während des gesamten Entwicklungszyklus gewährleisten können.

Erfahren Sie, wie Sie Ihre Feature-Branches mit einer Passwortauthentifizierung schützen können, um den Zugriff auf unveröffentlichte Funktionen einzuschränken. Aktivieren Sie Webvorschauen für Pull-Requests, um Änderungen in einer eindeutigen Vorschau zu überprüfen, URLs bevor Sie Code mit Produktionszweigen zusammenführen. Richten Sie end-to-end Tests mit dem Cypress-Framework ein, um Regressionen abzufangen, bevor Sie den Code in die Produktion bringen. Die Button-Funktion Deploy to Amplify ist zwar nicht mehr verfügbar, aber Sie können mit Amplify Hosting weiterhin problemlos Anwendungen direkt aus Ihrem Repository bereitstellen.

Topics

Beschränken des Zugriffs auf die Filialen einer Amplify-App

Wenn Sie an unveröffentlichten Funktionen arbeiten, können Sie Funktionszweige mit einem Passwort schützen, um den Zugriff auf bestimmte Benutzer zu beschränken. Wenn die Zugriffskontrolle für einen Branch eingerichtet ist, werden Benutzer aufgefordert, einen Benutzernamen und ein Passwort einzugeben, wenn sie versuchen, auf die URL für den Branch zuzugreifen.

Sie können ein Passwort festlegen, das für eine einzelne Filiale oder global für alle verbundenen Filialen gilt. Wenn die Zugriffskontrolle sowohl auf Filialebene als auch auf globaler Ebene aktiviert ist, hat das Passwort auf Filialebene Vorrang vor einem Passwort auf globaler (Anwendungs-) Ebene.

Amplify drosselt fehlgeschlagene Anfragen, die versuchen, auf kennwortgeschützte Ressourcen zuzugreifen. Dieses Verhalten schützt Anwendungen vor Wörterbuchangriffen oder anderen Versuchen, Daten hinter Zugriffskontrollen zu lesen.

Gehen Sie wie folgt vor, um ein Passwort festzulegen, um den Zugriff auf die Branches einer Amplify-App einzuschränken.

Um Passwörter für Feature-Branches festzulegen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie Feature-Branch-Passwörter einrichten möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Access Control aus.
4. Wählen Sie im Abschnitt Einstellungen für die Zugriffskontrolle die Option Zugriff verwalten aus.
5. Führen Sie auf der Seite „Zugriffskontrolle verwalten“ einen der folgenden Schritte aus.
   - Um einen Benutzernamen und ein Passwort festzulegen, die für alle verbundenen Filialen gelten
     - Aktivieren Sie die Option Zugriff für alle Filialen verwalten. Wenn Sie beispielsweise Main -, Dev - und Feature-Branches miteinander verbunden haben, können Sie für alle Branches denselben Benutzernamen und dasselbe Passwort verwenden.
   - Um einen Benutzernamen und ein Passwort festzulegen, die für einen einzelnen Branch gelten
     a. Deaktivieren Sie die Option Zugriff für alle Filialen verwalten.
     b. Suchen Sie die Filiale, die Sie verwalten möchten. Wählen Sie unter Zugriffseinstellungen die Option Eingeschränktes Passwort erforderlich aus.
     c. Geben Sie unter Nutzername einen Nutzernamen ein.
     d. Geben Sie in das Feld Passwort ein Passwort ein.
   - Wählen Sie Speichern.
6. Wenn Sie die Zugriffskontrolle für eine serverseitig gerenderte (SSR) App verwalten, stellen Sie die App erneut bereit, indem Sie einen neuen Build aus Ihrem Git-Repository ausführen. Dieser Schritt ist erforderlich, damit Amplify Ihre Einstellungen für die Zugriffskontrolle anwenden kann.

Webvorschauen für Pull-Requests

Webvorschauen bieten Teams für Entwicklung und Qualitätssicherung (QA) die Möglichkeit, eine Vorschau der Änderungen anhand von Pull-Requests (PRs) anzuzeigen, bevor der Code in einen Produktions- oder Integrationszweig übertragen wird. Mithilfe von Pull-Requests können Sie andere über Änderungen informieren, die Sie an einen Branch in einem Repository übertragen haben.

Nachdem ein Pull Request geöffnet wurde, kannst du die potenziellen Änderungen mit deinen Mitarbeitern besprechen und überprüfen und Folge-Commits hinzufügen, bevor deine Änderungen mit dem Basis-Branch zusammengeführt werden.

Bei einer Webvorschau wird jede Pull-Anfrage, die an dein Repository gestellt wird, an eine eindeutige Vorschau-URL gesendet, die sich völlig von der URL unterscheidet, die deine Hauptseite verwendet. Für Apps mit Backend-Umgebungen, die über die Amplify CLI oder Amplify Studio bereitgestellt werden, erstellt jede Pull-Anfrage (nur private Git-Repositorys) ein temporäres Backend, das gelöscht wird, wenn der PR geschlossen wird.

Wenn die Webvorschau für Ihre App aktiviert ist, wird jeder PR auf das Amplify-Kontingent von 50 Filialen pro App angerechnet. Um zu vermeiden, dass dieses Kontingent überschritten wird, stellen Sie sicher, dass Sie Ihre schließen. PRs Weitere Informationen zu Kontingenten finden Sie unter Amplify Kontingente für Hosting-Dienste.

> **Note**
>
> Derzeit ist die `AWS_PULL_REQUEST_ID` Umgebungsvariable nicht verfügbar, wenn Sie sie AWS CodeCommit als Repository-Anbieter verwenden.

Sicherheit in der Webvorschau

Aus Sicherheitsgründen können Sie die Webvorschau für alle Apps mit privaten Repositorys aktivieren, aber nicht für alle Apps mit öffentlichen Repositorys. Wenn Ihr Git-Repository öffentlich ist, können Sie Vorschauen nur für Apps einrichten, für die keine IAM-Servicerolle erforderlich ist. Beispielsweise benötigen Apps mit Backends und Apps, die auf der `WEB_COMPUTE` Hosting-Plattform bereitgestellt werden, eine IAM-Servicerolle. Daher können Sie für diese Arten von Apps keine Webvorschauen aktivieren, wenn ihr Repository öffentlich ist. Amplify erzwingt diese Einschränkung, um zu verhindern, dass Dritte beliebigen Code einreichen, der mit den IAM-Rollenberechtigungen Ihrer App ausgeführt werden würde.

Wenn Webvorschauen für eine Anwendung in einem öffentlichen Repository mit einer SSR-Compute-Rolle aktiviert sind, müssen Sie sorgfältig verwalten, welche Branches auf die Rolle zugreifen können. Wir empfehlen, keine Rolle auf App-Ebene zu verwenden. Stattdessen sollten Sie eine Rechenrolle auf Filialebene anhängen. Auf diese Weise können Sie nur den Branchen Berechtigungen gewähren, die Zugriff auf bestimmte Ressourcen benötigen. Weitere Informationen finden Sie unter Hinzufügen einer SSR-Compute-Rolle, um den Zugriff auf Ressourcen zu ermöglichen AWS.

Aktivieren Sie Webvorschauen für Pull-Requests

Für Apps, die in einem GitHub Repo gespeichert sind, verwenden Webvorschauen die Amplify GitHub App für den Repo-Zugriff. Wenn Sie Webvorschauen in einer vorhandenen Amplify-App aktivieren, die Sie zuvor von einem GitHub Repo aus bereitgestellt haben, das OAuth für den Zugriff verwendet wurde, müssen Sie die App zuerst migrieren, um die Amplify-App zu verwenden. GitHub Anweisungen zur Migration finden Sie unter. Migration einer vorhandenen OAuth App zur GitHub Amplify App

So aktivieren Sie Webvorschauen für Pull-Requests

1. Wählen Sie Hosting und dann Vorschauen aus.

   > **Note**
   >
   > Vorschauen sind im Menü mit den App-Einstellungen nur sichtbar, wenn eine App für die kontinuierliche Bereitstellung eingerichtet und mit einem Git-Repository verbunden ist. Anweisungen zu dieser Art der Bereitstellung findest du unter Erste Schritte mit vorhandenem Code.

2. Gehen Sie nur für GitHub Repositorys wie folgt vor, um die Amplify GitHub App in Ihrem Konto zu installieren und zu autorisieren:

   a. Wählen Sie im Fenster GitHub App installieren, um Vorschauen zu aktivieren die Option App installieren. GitHub

   b. Wählen Sie das GitHub Konto aus, in dem Sie die Amplify GitHub App konfigurieren möchten.

   c. Auf GitHub.com wird eine Seite geöffnet, auf der Sie die Repository-Berechtigungen für Ihr Konto konfigurieren können.

   d. Führen Sie eine der folgenden Aktionen aus:

      - Um die Installation auf alle Repositorys anzuwenden, wählen Sie Alle Repositorys.
      - Um die Installation auf die von Ihnen ausgewählten Repositorys zu beschränken, wählen Sie Nur ausgewählte Repositorys. Stellen Sie sicher, dass das Repo für die App, für die Sie Webvorschauen aktivieren, in die von Ihnen ausgewählten Repositorys aufgenommen wird.

   e. Wählen Sie Speichern aus.

3. Nachdem Sie die Vorschauen für Ihr Repo aktiviert haben, kehren Sie zur Amplify-Konsole zurück, um Vorschauen für bestimmte Branches zu aktivieren. Wählen Sie auf der Vorschauseite einen Zweig aus der Liste aus und klicken Sie auf Einstellungen bearbeiten.
4. Aktiviere auf der Seite „Vorschaueinstellungen verwalten“ die Option „Pull-Request-Vorschauen“. Wählen Sie dann Confirm (Bestätigen) aus.
5. Führen Sie für Fullstack-Anwendungen einen der folgenden Schritte aus:
   - Wählen Sie „Neue Backend-Umgebung für jeden Pull-Request erstellen“. Mit dieser Option können Sie Änderungen testen, ohne die Produktion zu beeinträchtigen.
   - Wählen Sie Alle Pull-Requests für diesen Branch auf eine bestehende Umgebung verweisen aus.
6. Wählen Sie Bestätigen aus.

Wenn Sie das nächste Mal eine Pull-Anfrage für die Filiale einreichen, erstellt Amplify Ihre PR und stellt sie auf einer Vorschau-URL bereit. Nachdem die Pull-Anfrage geschlossen wurde, wird die Vorschau-URL gelöscht und jede temporäre Backend-Umgebung, die mit der Pull-Anfrage verknüpft ist, wird gelöscht. Nur für GitHub Repositorys kannst du direkt über die Pull-Anfrage in deinem Konto auf eine Vorschau deiner GitHub URL zugreifen.

Zugriff auf die Webvorschau mit Subdomains

Webvorschauen für Pull-Requests sind über Subdomains für eine Amplify-App zugänglich, die mit einer von Amazon Route 53 verwalteten benutzerdefinierten Domain verbunden ist. Wenn der Pull-Request geschlossen wird, werden die mit dem Pull-Request verknüpften Branches und Subdomains automatisch gelöscht. Dies ist das Standardverhalten für Webvorschauen, nachdem Sie musterbasierte Feature-Branch-Bereitstellungen für Ihre App eingerichtet haben. Anweisungen zur Einrichtung automatischer Subdomains finden Sie unter. Automatische Subdomains für eine benutzerdefinierte Amazon Route 53-Domain einrichten

Einrichtung von end-to-end Cypress-Tests für Ihre Amplify-Anwendung

Sie können end-to-end (E2E) -Tests in der Testphase Ihrer Amplify-App ausführen, um Regressionen catch, bevor Sie den Code in die Produktion bringen. Die Testphase kann in der Build-Spezifikation YAML konfiguriert werden. Derzeit können Sie während eines Builds nur das Cypress-Testframework ausführen.

Cypress ist ein JavaScript basiertes Test-Framework, mit dem Sie E2E-Tests in einem Browser ausführen können. Ein Tutorial, das zeigt, wie Sie E2E-Tests einrichten, finden Sie im Blogbeitrag Ausführen von end-to-end Cypress-Tests für Ihre CI/CD Fullstack-Bereitstellung mit Amplify.

Hinzufügen von Cypress-Tests zu einer vorhandenen Amplify-Anwendung

Sie können Cypress-Tests zu einer vorhandenen App hinzufügen, indem Sie die Build-Einstellungen der App in der Amplify-Konsole aktualisieren. Die Build-Spezifikation YAML enthält eine Sammlung von Build-Befehlen und zugehörigen Einstellungen, die Amplify verwendet, um Ihren Build auszuführen. Verwenden Sie den `test` Schritt, um alle Testbefehle zur Build-Zeit auszuführen. Für E2E-Tests bietet Amplify Hosting eine tiefere Integration mit Cypress, mit der Sie einen UI-Bericht für Ihre Tests erstellen können.

In der folgenden Liste werden die Testeinstellungen und deren Verwendung beschrieben.

Vor dem Test

Installieren Sie die Abhängigkeiten, die für die Ausführung von Cypress-Tests erforderlich sind. Amplify Hosting verwendet Mochawesome, um einen Bericht zu erstellen, in dem Sie Ihre Testergebnisse einsehen können, und warten Sie, bis der Localhost-Server während des Builds eingerichtet ist.

Test

Führen Sie Cypress-Befehle aus, um Tests mit Mochawesome durchzuführen.

Nach dem Test

Der Mochawesome-Bericht wird aus der JSON-Ausgabe generiert. Beachten Sie, dass Sie, wenn Sie Yarn verwenden, diesen Befehl im unbeaufsichtigten Modus ausführen müssen, um den Mochawesome-Bericht zu generieren. Für Yarn können Sie den folgenden Befehl verwenden.

```
yarn run --silent mochawesome-merge cypress/report/mochawesome-report/
mochawesome*.json > cypress/report/mochawesome.json
```

Artefakte > Basisverzeichnis

Das Verzeichnis, von dem aus Tests ausgeführt werden.

Artefakte> configFilePath

Die generierten Testberichtsdaten.

Artefakte>Dateien

Die generierten Artefakte (Screenshots und Videos) stehen zum Herunterladen zur Verfügung.

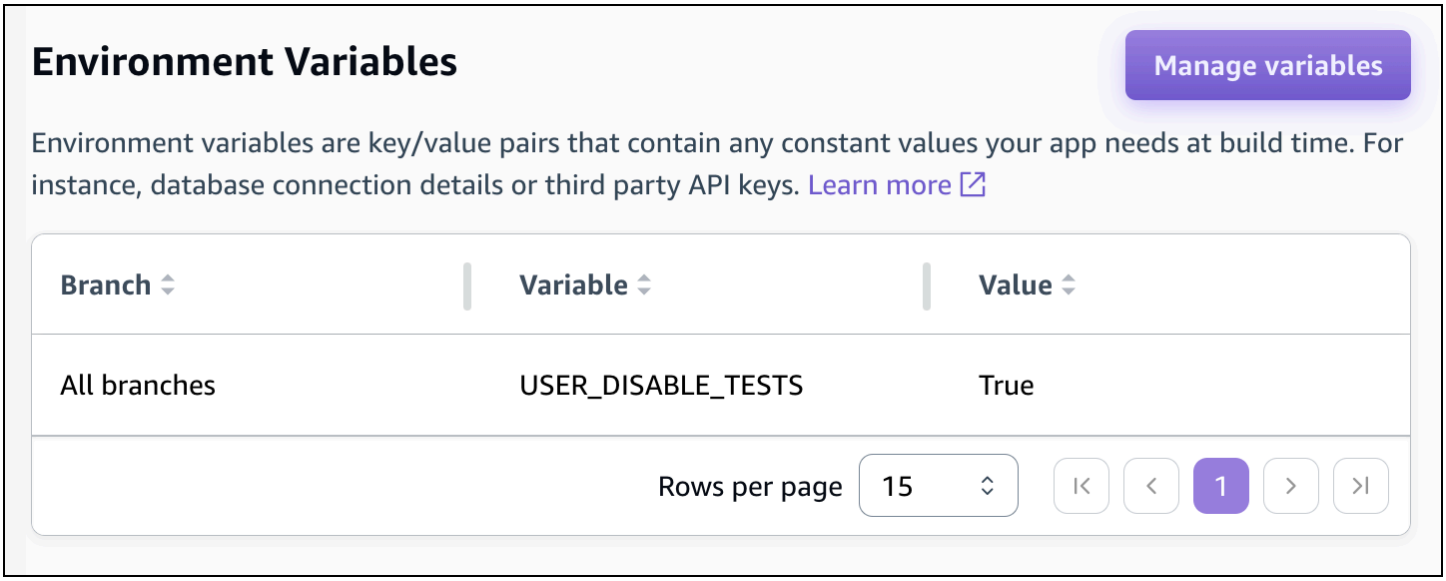
Der folgende Beispielauszug aus einer `amplify.yml` Build-Spezifikationsdatei zeigt, wie Sie Ihrer App Cypress-Tests hinzufügen.

```
test:
  phases:
    preTest:
      commands:
        - npm ci
        - npm install -g pm2
        - npm install -g wait-on
        - npm install mocha mochawesome mochawesome-merge mochawesome-report-generator
        - pm2 start npm -- start
        - wait-on http://localhost:3000
    test:
      commands:
        - 'npx cypress run --reporter mochawesome --reporter-options
 "reportDir=cypress/report/mochawesome-
report,overwrite=false,html=false,json=true,timestamp=mmddyyyy_HHMMss"'
    postTest:
      commands:
        - npx mochawesome-merge cypress/report/mochawesome-report/mochawesome*.json >
 cypress/report/mochawesome.json
        - pm2 kill
  artifacts:
    baseDirectory: cypress
    configFilePath: '**/mochawesome.json'
    files:
      - '**/*.png'
      - '**/*.mp4'
```

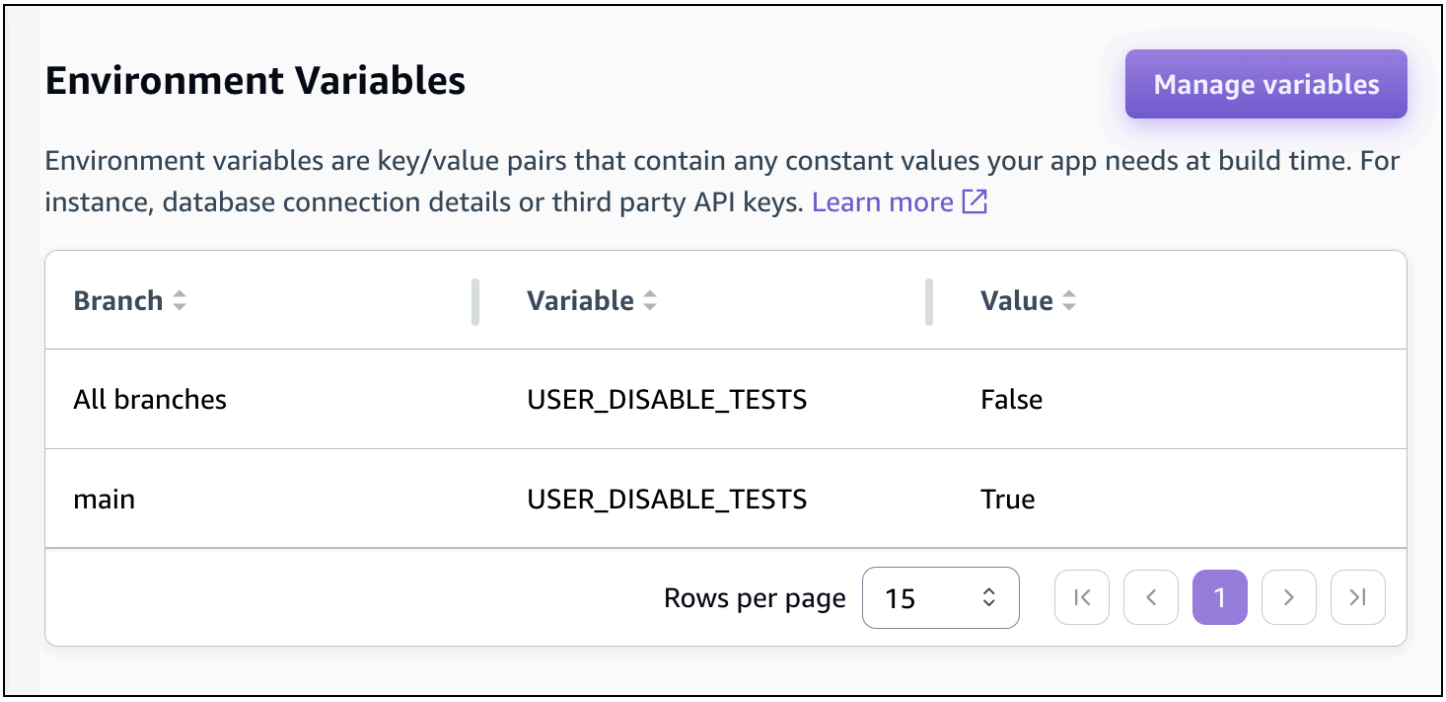
Tests für eine Amplify-Anwendung oder einen Amplify-Zweig ausschalten

Nachdem die Testkonfiguration zu Ihren `amplify.yml` Build-Einstellungen hinzugefügt wurde, wird der `test` Schritt für jeden Build in jedem Branch ausgeführt. Wenn Sie die Ausführung von Tests global deaktivieren oder nur Tests für bestimmte Zweige ausführen möchten, können Sie die USER_DISABLE_TESTS Umgebungsvariable verwenden, ohne Ihre Build-Einstellungen zu ändern.

Um Tests für alle Branches global zu deaktivieren, fügen Sie die USER_DISABLE_TESTS Umgebungsvariable mit dem Wert `true` for all Branches hinzu. Der folgende Screenshot zeigt den Abschnitt Umgebungsvariablen in der Amplify-Konsole mit deaktivierten Tests für alle Zweige.



Um Tests für einen bestimmten Zweig zu deaktivieren, fügen Sie die USER_DISABLE_TESTS Umgebungsvariable mit dem Wert `false` für alle Zweige hinzu und fügen Sie dann für jeden Zweig, den Sie deaktivieren möchten, eine Überschreibung mit dem Wert von `true` hinzu. Im folgenden Screenshot sind Tests für den Hauptzweig deaktiviert und für jeden anderen Zweig aktiviert.



Das Deaktivieren von Tests mit dieser Variablen führt dazu, dass der Testschritt während eines Builds vollständig übersprungen wird. Um Tests wieder zu aktivieren, setzen Sie diesen Wert auf `false` oder löschen Sie die Umgebungsvariable.

Verwenden der Schaltfläche „Deploy to Amplify“, um ein GitHub Projekt zu teilen

> **Important**
>
> Die Bereitstellung mit einem Klick über die Schaltfläche Deploy to Amplify Hosting ist nicht mehr verfügbar. Um die Bereitstellung aus einem Repository durchzuführen, erstellen Sie eine neue Anwendung in Amplify Hosting. Detaillierte Anweisungen finden Sie unter Erste Schritte mit der Bereitstellung einer App auf Amplify Hosting.

Mit der Schaltfläche Deploy to Amplify Hosting können Sie GitHub Projekte öffentlich oder innerhalb Ihres Teams teilen. Das Folgende ist ein Bild der Schaltfläche:

Hinzufügen der Schaltfläche Deploy to Amplify Hosting zu einem Repository oder Blog

Fügen Sie die Schaltfläche zu Ihrer GitHub README.md-Datei, Ihrem Blogbeitrag oder einer anderen Markup-Seite hinzu, die HTML rendert. Die Schaltfläche besteht aus den folgenden zwei Komponenten:

1. Ein SVG-Bild befindet sich unter der URL `https://oneclick.amplifyapp.com/button.svg`
2. Die URL der Amplify-Konsole mit einem Link zu Ihrem GitHub Repository. Sie können entweder die URL Ihres Repositorys kopieren, z. B.`https://github.com/username/repository`, oder Sie können einen Deep-Link in einen bestimmten Ordner bereitstellen, z. B. `https://github.com/username/repository/tree/branchname/folder` Amplify Hosting stellt den Standard-Branch in Ihrem Repository bereit. Nach dem Verknüpfen der App können weitere Verzweigungen verknüpft werden.

Verwenden Sie das folgende Beispiel, um die Schaltfläche zu einer Markdown-Datei wie Ihrer GitHub README.md hinzuzufügen. Ersetze es `https://github.com/username/repository` durch die URL zu deinem Repository.

```
[![amplifybutton](https://oneclick.amplifyapp.com/button.svg)](https://
console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/username/
repository)
```

Verwenden Sie das folgende Beispiel, um die Schaltfläche zu einem beliebigen HTML-Dokument hinzuzufügen. `https://github.com/username/repository`Ersetzen Sie es durch die URL zu Ihrem Repository.

```
<a href="https://console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/
username/repository">
    <img src="https://oneclick.amplifyapp.com/button.svg" alt="Deploy to Amplify
 Hosting">
</a>
```

Umleitungen und Umschreibungen für eine Amplify-Anwendung einrichten

Umleitungen ermöglichen einem Webserver das Umleiten der Navigation von einer URL zu einer anderen. Zu den häufigsten Gründen für die Verwendung von Weiterleitungen gehören das Anpassen des Erscheinungsbilds einer URL, die Vermeidung defekter Links, das Verschieben des Hosting-Speicherorts einer App oder Website, ohne deren Adresse zu ändern, und das Ändern einer angeforderten URL in das von einer Web-App benötigte Formular.

Die von Amplify unterstützten Weiterleitungen verstehen

Amplify unterstützt die folgenden Umleitungstypen in der Konsole.

Dauerhafte Umleitung (301)

301-Umleitungen sind für dauerhafte Änderungen am Ziel einer Webadresse vorgesehen. Der Verlauf des Suchmaschinenrankings der ursprünglichen Adresse gilt für die neue Zieladresse. Die Umleitung erfolgt auf der Clientseite, daher wird in einer Browser-Navigationsleiste nach der Umleitung die Zieladresse angezeigt.

Häufige Gründe für die Verwendung von 301-Umleitungen:

- Vermeiden eines fehlerhaften Links beim Ändern der Adresse einer Seite
- Vermeiden eines fehlerhaften Links, wenn ein Benutzer einen vorhersehbare Tippfehler in einer Adresse macht

Temporäre Umleitung (302)

302-Umleitungen sind für temporäre Änderungen am Ziel einer Webadresse vorgesehen. Der Verlauf des Suchmaschinen-Rankings der ursprünglichen Adresse gilt nicht für die neue Zieladresse. Die Umleitung erfolgt auf der Clientseite, daher wird in einer Browser-Navigationsleiste nach der Umleitung die Zieladresse angezeigt.

Häufige Gründe für die Verwendung von 302-Umleitungen:

- Bereitstellen eines Umleitungsziels, während an einer ursprünglichen Adresse Reparaturen vorgenommen werden

- Um Testseiten für den A/B Vergleich einer Benutzeroberfläche bereitzustellen.

> **Note**
>
> Wenn Ihre App eine unerwartete 302-Antwort zurückgibt, wird der Fehler wahrscheinlich durch Änderungen verursacht, die Sie an der Weiterleitung und der benutzerdefinierten Header-Konfiguration Ihrer App vorgenommen haben. Um dieses Problem zu beheben, stellen Sie sicher, dass Ihre benutzerdefinierten Header gültig sind, und aktivieren Sie dann erneut die standardmäßige 404-Rewrite-Regel für Ihre App.

Umschreibung (200)

200-Umleitungen (Umschreibungen) dienen zum Anzeigen von Inhalt an der Zieladresse, als würde er von der ursprünglichen Adresse bereitgestellt. Der Verlauf des Suchmaschinenrankings gilt weiterhin für die ursprüngliche Adresse. Die Umleitung erfolgt auf der Serverseite, daher wird in einer Browser-Navigationsleiste nach der Umleitung die ursprüngliche Adresse angezeigt. Häufige Gründe für die Verwendung von 200-Umleitungen:

- Umleiten einer gesamten Website an einen neuen Hostingstandort, ohne die Adresse der Website zu ändern
- Umleiten des gesamten Datenverkehrs an die Seite „index.html“ einer Single Page Web App (SPA) zur Verarbeitung durch eine clientseitige Routerfunktion

Nicht gefunden (404)

404-Weiterleitungen treten auf, wenn eine Anfrage auf eine Adresse verweist, die nicht existiert. Die Zielseite einer 404-Umleitung wird anstatt der angeforderten Seite angezeigt. Häufige Gründe für 404-Umleitungen:

- Vermeiden einer Meldung vom Typ „Fehlerhafter Link“, wenn ein Benutzer eine ungültige URL eingibt
- Verweisen von Anforderungen an nicht vorhandene Seiten einer Web-App an die Seite „index.html“ zur Verarbeitung durch eine clientseitige Routerfunktion

Die Reihenfolge der Weiterleitungen verstehen

Weiterleitungen werden vom Anfang der Liste nach unten angewendet. Stellen Sie sicher, dass Ihre Reihenfolge das vorgesehene Ergebnis erzielt. Beispiel: Die folgende Reihenfolge von Umleitungen bewirkt, dass alle Anforderungen für einen bestimmten Pfad unter /docs/ an denselben Pfad unter /documents/ umgeleitet werden, mit Ausnahme von /docs/bestimmter-dateiname.html, das an /documents/anderer-dateiname.html umgeleitet wird:

```
/docs/specific-filename.html /documents/different-filename.html 301
/docs/<*> /documents/<*>
```

Bei der folgenden Umleitungsreihenfolge wird die Umleitung von bestimmter-dateiname.html an anderer-dateiname.html ignoriert:

```
/docs/<*> /documents/<*>
/docs/specific-filename.html /documents/different-filename.html 301
```

Verstehen, wie Amplify Abfrageparameter weiterleitet

Sie können Abfrageparameter verwenden, um mehr Kontrolle über Ihre URL-Übereinstimmungen zu haben. Amplify leitet alle Abfrageparameter für 301- und 302-Weiterleitungen an den Zielpfad weiter, mit den folgenden Ausnahmen:

- Wenn die ursprüngliche Adresse eine Abfragezeichenfolge enthält, die auf einen bestimmten Wert festgelegt ist, leitet Amplify keine Abfrageparameter weiter. In diesem Fall gilt die Umleitung nur für Anfragen an die Ziel-URL mit dem angegebenen Abfragewert.
- Wenn die Zieladresse für die Abgleichsregel Abfrageparameter enthält, werden Abfrageparameter nicht weitergeleitet. Wenn die Zieladresse für die Umleitung beispielsweise lautet`https://example-target.com?q=someParam`, werden Abfrageparameter nicht weitergeleitet.

Weiterleitungen in der Amplify-Konsole erstellen und bearbeiten

Sie können Weiterleitungen für eine Anwendung in der Amplify-Konsole erstellen und bearbeiten. Bevor Sie beginnen, benötigen Sie die folgenden Informationen zu den Teilen einer Weiterleitung.

Eine Originaladresse

Die Adresse, die der Benutzer angefordert hat.

Eine Zieladresse

Die Adresse, die tatsächlich den Inhalt bereitstellt, den der Benutzer sieht.

Ein Umleitungstyp

Zu den Typen gehören eine permanente Weiterleitung (301), eine temporäre Weiterleitung (302), eine Umschreibung (200) oder eine nicht gefundene Weiterleitung (404).

Ein aus zwei Buchstaben bestehender Ländercode (optional)

Ein Wert, den Sie angeben können, um die Nutzererfahrung Ihrer App nach geografischer Region zu segmentieren.

Um eine Weiterleitung in der Amplify-Konsole zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie eine Weiterleitung erstellen möchten.
3. Wählen Sie im Navigationsbereich Hosting und anschließend Umschreibungen und Weiterleitungen aus.
4. Wählen Sie auf der Seite Umschreibungen und Weiterleitungen die Option Weiterleitungen verwalten aus.
5. Fügen Sie Weiterleitungen im JSON-Editor für Umschreibungen und Weiterleitungen manuell hinzu oder aktualisieren Sie sie.
   a. Geben Sie für die ursprüngliche Adresse an`source`, die der Benutzer angefordert hat.
   b. Geben Sie für `status` die Art der Weiterleitung an.
   c. Geben Sie für `target` die Zieladresse an, die den Inhalt für den Benutzer wiedergibt.
   d. (Optional) Geben Sie für `condition` einen aus zwei Buchstaben bestehenden Ländercode ein.
6. Wählen Sie Speichern.

Leitet die Beispielreferenz weiter und schreibt sie neu

Dieser Abschnitt enthält Beispiele für eine Vielzahl gängiger Umleitungsszenarien.

Important

Domänenspezifische Weiterleitungen unterstützen keine Pfadkomponenten im Quellfeld. Unterstützt:

- `"source": "https://example.com"` (Pfade werden automatisch angehängt)

Nicht unterstützt:

- `"source": "https://example.com/specific-path"`
- Regeln mit `domain+path` Kombinationen werden derzeit nicht unterstützt.

Alternative Muster
Verwenden Sie für domänenspezifische Pfadweiterleitungen:

1. Separate Regeln nur für Domänen (Pfade werden automatisch angehängt)
2. Reine Pfadregeln mit bedingter Logik
3. Mehrere Regelkombinationen

Sie können diese Beispiele verwenden, um die JSON-Syntax für die Erstellung Ihrer eigenen Weiterleitungen und Umschreibungen im JSON-Editor der Amplify-Konsole zu verstehen.

Note

Beim Domainabgleich mit der ursprünglichen Adresse wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Themen

- Regionsbasierte Weiterleitungen
- Verwendung von Platzhalterausdrücken bei Weiterleitungen und Umschreibungen

Einfache Weiterleitungen und Umschreibungen

Sie können das folgende Beispiel verwenden, um eine bestimmte Seite dauerhaft an eine neue Adresse umzuleiten.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/original.html`	`/destination.html`	`permanent redirect (301)`	

JSON-Format

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

Sie können das folgende Beispiel verwenden, um einen beliebigen Pfad unter einem Ordner auf denselben Pfad in einem anderen Ordner umzuleiten.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

JSON-Format

```
[
```

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

Sie können das folgende Beispiel verwenden, um den gesamten Datenverkehr beim Umschreiben an index.html umzuleiten. In diesem Szenario wird die Seite dem Benutzer so angezeigt, als befände er sich an der ursprünglichen Adresse.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/<*>`	`/index.html`	`rewrite (200)`	

JSON-Format

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

Sie können das folgende Beispiel verwenden, um mit einem Rewrite die Subdomain zu ändern, die dem Benutzer angezeigt wird.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`https://mydomain.com`	`https://www.mydomain.com`	`rewrite (200)`	

JSON-Format

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

Sie können das folgende Beispiel verwenden, um zu einer anderen Domain mit einem Pfadpräfix umzuleiten.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`https://mydomain.com`	`https://www.mydomain.com/documents`	`temporary redirect (302)`	

JSON-Format

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

Sie können das folgende Beispiel verwenden, um Pfade unter einem Ordner, der nicht gefunden werden kann, auf eine benutzerdefinierte 404-Seite umzuleiten.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/<*>`	`/404.html`	`not found (404)`	

JSON-Format

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

> **Important**
>
> Pfadkomponenten in domänenbasierten Quellregeln (wie`"https://domain.com/path"`) werden nicht unterstützt und führen dazu, dass die Regel ohne Fehler ignoriert wird.

Weiterleitungen für Single-Page-Web-Apps (SPA)

Die meisten SPA-Frameworks unterstützen HTML5 history.pushState (), um den Browserstandort zu ändern, ohne eine Serveranfrage auszulösen. Diese Methode funktioniert für Benutzer, die an der Stamm-URL (oder /index.html) beginnen, aber nicht für Benutzer, die direkt zu einer anderen Seite navigieren.

Im folgenden Beispiel werden reguläre Ausdrücke verwendet, um einen 200-Rewrite-Vorgang für alle Dateien in index.html einzurichten, mit Ausnahme der im regulären Ausdruck angegebenen Dateierweiterungen.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`</^[^.]+$|\.(?!(css|gif|ico`	`/index.html`	`200`	

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|webp)$)([^.]+$)/>`			

JSON-Format

```
[
  {
    "source": "</^[^.]+$|\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|
webp)$)([^.]+$)/>",
    "status": "200",
    "target": "/index.html",
    "condition": null
  }
]
```

Reverse-Proxy umschreiben

Im folgenden Beispiel wird ein Rewrite verwendet, um Inhalt von einem anderen Ort aus als Proxy zu verwenden, sodass der Benutzer den Eindruck hat, dass sich die Domain nicht geändert hat. HTTPS ist das einzige Protokoll, das für Reverse-Proxys unterstützt wird.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/images/<*>`	`https://images.otherdomain.com/<*>`	`rewrite (200)`	

JSON-Format

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

Schrägstriche am Ende und sauber URLs

Zum Erstellen bereinigter URL-Strukturen wie about anstelle von about.html generieren Generatoren von statischen Seiten wie z. B. Hugo Verzeichnisse für Seiten mit „index.html“ (/about/index.html). Amplify erstellt automatisch Clean, URLs indem es bei Bedarf einen abschließenden Schrägstrich hinzufügt. In der folgenden Tabelle sind verschiedene Szenarien hervorgehoben:

Benutzereingaben im Browser	URL in der Adressleiste	Bereitgestelltes Dokument
`/about`	`/about`	`/about.html`
`/about (when about.html returns 404)`	`/about/`	`/about/index.html`
`/about/`	`/about/`	`/about/index.html`

Platzhalter

Sie können das folgende Beispiel verwenden, um Pfade in einer Ordnerstruktur zu einer passenden Struktur in einem anderen Ordner umzuleiten.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/docs/<year>/<month>/<date>/<itemid>`	`/documents/<year>/<month>/<date>/<itemid>`	`permanent redirect (301)`	

JSON-Format

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

Abfragen von Zeichenketten und Pfadparametern

> **Warning**
>
> Schließen Sie keine Geheimnisse, Anmeldeinformationen oder vertraulichen Daten URLs als Pfad- oder Abfrageparameter ein. Diese Werte sind im Klartext in den Zugriffsprotokollen Ihrer Amplify-Anwendung sichtbar.

Sie können das folgende Beispiel verwenden, um einen Pfad zu einem Ordner umzuleiten, dessen Name dem Wert eines Abfragezeichenfolgenelements in der ursprünglichen Adresse entspricht:

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/docs?id=<my-blog-id-value`	`/documents/<my-blog-post-id-value>`	`permanent redirect (301)`	

JSON-Format

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

```
]
```

> **Note**
>
> Amplify leitet alle Abfragezeichenfolgenparameter für 301- und 302-Weiterleitungen an den Zielpfad weiter. Wenn die ursprüngliche Adresse jedoch eine Abfragezeichenfolge enthält, die auf einen bestimmten Wert festgelegt ist, wie in diesem Beispiel gezeigt, leitet Amplify keine Abfrageparameter weiter. In diesem Fall gilt die Umleitung nur für Anfragen an die Zieladresse mit dem angegebenen Abfragewert`id`.

Sie können das folgende Beispiel verwenden, um alle Pfade, die auf einer bestimmten Ebene einer Ordnerstruktur nicht gefunden werden können, zu index.html in einem angegebenen Ordner umzuleiten.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/documents/<folder>/<child-folder>/<grand-child-folder>`	`/documents/index.html`	`not found (404)`	

JSON-Format

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

Regionsbasierte Weiterleitungen

Sie können das folgende Beispiel verwenden, um Anfragen basierend auf der Region umzuleiten.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/documents`	`/documents/us/`	`temporary redirect (302)`	`<US>`

JSON-Format

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

Verwendung von Platzhalterausdrücken bei Weiterleitungen und Umschreibungen

Sie können den Platzhalterausdruck,<*>, in der ursprünglichen Adresse für eine Umleitung oder ein Umschreiben verwenden. Sie müssen den Ausdruck am Ende der ursprünglichen Adresse platzieren und er muss eindeutig sein. Amplify ignoriert Originaladressen, die mehr als einen Platzhalterausdruck enthalten, oder verwendet ihn an einer anderen Stelle.

Im Folgenden finden Sie ein Beispiel für eine gültige Weiterleitung mit einem Platzhalterausdruck.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

Die folgenden beiden Beispiele zeigen ungültige Weiterleitungen mit Platzhalterausdrücken.

Ursprüngliche Adresse	Zieladresse	Umleitungsart	Country Code (Ländercode)
`/docs/<*>/content`	`/documents/<*>/content`	`permanent redirect (301)`	
`/docs/<*>/content/<*>`	`/documents/<*>/content/<*>`	`permanent redirect (301)`	

Verwenden von Umgebungsvariablen in einer Amplify-Anwendung

Umgebungsvariablen sind Schlüssel-Wert-Paare, die Sie zu den Einstellungen Ihrer Anwendung hinzufügen können, um sie für Amplify Hosting verfügbar zu machen. Als bewährte Methode können Sie Umgebungsvariablen verwenden, um Anwendungskonfigurationsdaten verfügbar zu machen. Alle Umgebungsvariablen, die Sie hinzufügen, sind verschlüsselt, um unbefugten Zugriff zu verhindern.

Amplify erzwingt die folgenden Einschränkungen für die von Ihnen erstellten Umgebungsvariablen.

- Amplify erlaubt Ihnen nicht, Umgebungsvariablennamen mit einem `AWS` Präfix zu erstellen. Dieses Präfix ist nur für den internen Gebrauch von Amplify reserviert.
- Der Wert einer Umgebungsvariablen darf 5500 Zeichen nicht überschreiten.

> **Important**
>
> Verwenden Sie keine Umgebungsvariablen zum Speichern von Geheimnissen. Verwenden Sie für eine Gen 2-App die Secret-Verwaltungsfunktion in der Amplify-Konsole. Weitere Informationen finden Sie unter Secrets and environment vars in der Amplify-Dokumentation. Speichern Sie für eine Gen-1-App Geheimnisse in einem Umgebungsgeheimnis, das mit dem AWS Systems Manager Parameter Store erstellt wurde. Weitere Informationen finden Sie unter Verwaltung von Umgebungsgeheimnissen.

Umgebungsvariablenreferenz Amplify

Auf die folgenden Umgebungsvariablen kann standardmäßig in der Amplify-Konsole zugegriffen werden.

Variablenname	Description	Beispielwert
_BUILD_TIMEOUT	Die Dauer des Build-Timeouts in Minuten. Der Mindestwert ist 5.	`30`

Variablenname	Description	Beispielwert
	Der Höchstwert ist 120.	
_LIVE_UPDATES	Das Tool wird auf die neueste Version aktualisiert.	`[{"name":"Amplify CLI","pkg":"@aws-amplify/cli","type":"npm","version":"latest"}]`
USER_DISABLE_TESTS	Der Testschritt wird während eines Builds übersprungen. Sie können Tests für alle Zweige oder bestimmte Zweige in einer App deaktivieren. Diese Umgebungsvariable wird für Apps verwendet, die während der Buildphase Tests durchführen. Weitere Informationen zum Setzen dieser Variablen finden Sie unterTests für eine Amplify-Anwendung oder einen Amplify-Zweig ausschalten.	`true`
AWS_APP_ID	Die App-ID des aktuellen Builds	`abcd1234`
AWS_BRANCH	Der Verzweigungsname des aktuellen Builds	`main`, `develop`, `beta`, `v2.0`
AWS_BRANCH_ARN	Der Branch-Amazon-Ressourcenname (ARN) des aktuellen Builds	`aws:arn:amplify:us-west-2:123456789012:appname/branch/...`

Variablenname	Description	Beispielwert
AWS_CLONE_URL	Die Klon-URL zum Abrufen der Git-Repository-Inhalte	`git@github.com:<user-name>/<repo-name>.git`
AWS_COMMIT_ID	Die Commit-ID des aktuellen Builds „HEAD“ für Rebuilds	`abcd1234`
AWS_JOB_ID	Die Auftrags-ID des aktuellen Builds. Dies beinhaltet eine gewisse Polsterung von '0', sodass es immer dieselbe Länge hat.	`0000000001`
AWS_PULL_REQUEST_ID	Die Pull-Request-ID des Pull-Request-Webvorschau-Builds. Diese Umgebungsvariable ist nicht verfügbar, wenn Sie sie AWS CodeCommit als Repository-Anbieter verwenden.	`1`
AWS_PULL_REQUEST_SOURCE_BRANCH	Der Name des Feature-Branches für eine Pull-Request-Vorschau, der an einen Anwendungszweig in der Amplify-Konsole gesendet wird.	`featureA`
AWS_PULL_REQUEST_DESTINATION_BRANCH	Der Name des Anwendungszweigs in der Amplify-Konsole, an den eine Feature-Branch-Pull-Anfrage gesendet wird.	`main`

Variablenname	Description	Beispielwert
AMPLIFY_AMAZON_CLIENT_ID	Die Amazon-Kunden-ID	`123456`
AMPLIFY_AMAZON_CLIENT_SECRET	Das Geheimnis des Amazon-Kunden	`example123456`
AMPLIFY_FACEBOOK_CLIENT_ID	Die Facebook-Kunden-ID	`123456`
AMPLIFY_FACEBOOK_CLIENT_SECRET	Das geheime Facebook-Kundengeheimnis	`example123456`
AMPLIFY_GOOGLE_CLIENT_ID	Die Google-Kunden-ID	`123456`
AMPLIFY_GOOGLE_CLIENT_SECRET	Das Geheimnis des Google-Clients	`example123456`
AMPLIFY_DIFF_DEPLOY	Aktivieren oder deaktivieren Sie die diff-basierte Frontend-Bereitstellung. Weitere Informationen finden Sie unter Konfiguration von diff-basiertem Frontend, Build und Deployment.	`true`
AMPLIFY_DIFF_DEPLOY_ROOT	Der Pfad, der für Vergleichsvergleiche von Frontend-Deployments verwendet werden soll, relativ zum Stammverzeichnis Ihres Repositorys.	`dist`

Variablenname	Description	Beispielwert
AMPLIFY_DIFF_BACKEND	Aktiviert oder deaktiviert diff-basierte Backend-Builds. Dies gilt nur für Apps der 1. Generation. Weitere Informationen finden Sie unter Konfiguration von diff-basierten Backend-Builds für eine Gen-1-App.	`true`
AMPLIFY_BACKEND_PULL_ONLY	Amplify verwaltet diese Umgebungsvariable. Dies gilt nur für Apps der 1. Generation. Weitere Informationen finden Sie unter Bearbeiten Sie ein vorhandenes Frontend so, dass es auf ein anderes Backend verweist.	`true`
AMPLIFY_BACKEND_APP_ID	Amplify verwaltet diese Umgebungsvariable. Dies gilt nur für Apps der 1. Generation. Weitere Informationen finden Sie unter Bearbeiten Sie ein vorhandenes Frontend so, dass es auf ein anderes Backend verweist.	`abcd1234`
AMPLIFY_SKIP_BACKEND_BUILD	Wenn Ihre Build-Spezifikation keinen Backend-Abschnitt enthält und Sie Backend-Builds deaktivieren möchten, setzen Sie diese Umgebungsvariable auf `true` Dies gilt nur für Apps der 1. Generation.	`true`

Variablenname	Description	Beispielwert
AMPLIFY_ENABLE_DEBUG_OUTPUT	Setzen Sie diese Variable auf, `true` um einen Stack-Trace in den Protokollen zu drucken. Dies ist hilfreich beim Debuggen von Backend-Build-Fehlern.	`true`
AMPLIFY_MONOREPO_APP_ROOT	Der Pfad, der verwendet werden soll, um das App-Stammverzeichnis einer Monorepo-App relativ zum Stammverzeichnis Ihres Repositorys anzugeben.	`apps/react-app`
AMPLIFY_USERPOOL_ID	Die ID für den Amazon Cognito Cognito-Benutzerpool, der für die Authentifizierung importiert wurde	`us-west-2_example`
AMPLIFY_WEBCLIENT_ID	Die ID für den App-Client, der von Webanwendungen verwendet werden soll Der App-Client muss mit Zugriff auf den Amazon Cognito Cognito-Benutzerpool konfiguriert sein, der in der Umgebungsvariablen AMPLIFY_USERPOOL_ID angegeben ist.	`123456`

Variablenname	Description	Beispielwert
AMPLIFY_NATIVECLIENT_ID	Die ID für den App-Client, der von nativen Anwendungen verwendet werden soll Der App-Client muss mit Zugriff auf den Amazon Cognito Cognito-Benutzerpool konfiguriert sein, der in der Umgebungsvariablen AMPLIFY_USERPOOL_ID angegeben ist.	`123456`
AMPLIFY_IDENTITYPOOL_ID	Die ID für den Amazon Cognito Cognito-Identitätspool	`example-identitypool-id`
AMPLIFY_PERMISSIONS_BOUNDARY_ARN	Der ARN für die IAM-Richtlinie, der als Berechtigungsgrenze verwendet werden soll und für alle von Amplify erstellten IAM-Rollen gilt.	`arn:aws:iam::123456789012:policy/example-policy`
AMPLIFY_DESTRUCTIVE_UPDATES	Setzen Sie diese Umgebungsvariable auf true, damit eine GraphQL-API mit Schemaoperationen aktualisiert werden kann, die möglicherweise zu Datenverlust führen können.	`true`

> **Note**
>
> Die Umgebungsvariablen `AMPLIFY_AMAZON_CLIENT_ID` und die `AMPLIFY_AMAZON_CLIENT_SECRET` Umgebungsvariablen sind OAuth Token, kein AWS Zugriffsschlüssel und kein geheimer Schlüssel.

Umgebungsvariablen des Frontend-Frameworks

Wenn Sie Ihre App mit einem Frontend-Framework entwickeln, das seine eigenen Umgebungsvariablen unterstützt, ist es wichtig zu verstehen, dass diese nicht mit den Umgebungsvariablen identisch sind, die Sie in der Amplify-Konsole konfigurieren. Mit React (mit dem Präfix REACT_APP) und Gatsby (mit dem Präfix GATSBY) können Sie beispielsweise Laufzeitumgebungsvariablen erstellen, die diese Frameworks automatisch in Ihren Frontend-Produktions-Build bündeln. Informationen zu den Auswirkungen der Verwendung dieser Umgebungsvariablen zum Speichern von Werten finden Sie in der Dokumentation für das von Ihnen verwendete Frontend-Framework.

Das Speichern sensibler Werte, wie API-Schlüssel, in diesen Umgebungsvariablen des Frontend-Frameworks mit Präfix ist keine bewährte Methode und es wird dringend davon abgeraten.

Umgebungsvariablen setzen

Verwenden Sie die folgenden Anweisungen, um Umgebungsvariablen für eine Anwendung in der Amplify-Konsole festzulegen.

> **Note**
>
> Umgebungsvariablen sind nur dann im App-Einstellungsmenü der Amplify-Konsole sichtbar, wenn eine App für die kontinuierliche Bereitstellung eingerichtet und mit einem Git-Repository verbunden ist. Anweisungen zu dieser Art der Bereitstellung finden Sie unter Erste Schritte mit vorhandenem Code.

So legen Sie Umgebungsvariablen fest

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie in der Amplify-Konsole Hosting und dann Umgebungsvariablen aus.
3. Wählen Sie auf der Seite Umgebungsvariablen die Option Variablen verwalten aus.
4. Geben Sie unter Variable Ihren Schlüssel ein. Geben Sie unter Wert Ihren Wert ein. Standardmäßig wendet Amplify die Umgebungsvariablen auf alle Zweige an, sodass Sie Variablen nicht erneut eingeben müssen, wenn Sie einen neuen Zweig verbinden.
5. (Optional) Um eine Umgebungsvariable speziell für einen Zweig anzupassen, fügen Sie wie folgt eine Branch-Override hinzu:

a. Wählen Sie „Aktionen“ und anschließend „Variablenüberschreibung hinzufügen“.

b. Sie verfügen jetzt über einen spezifischen Satz von Umgebungsvariablen für Ihre Verzweigung.

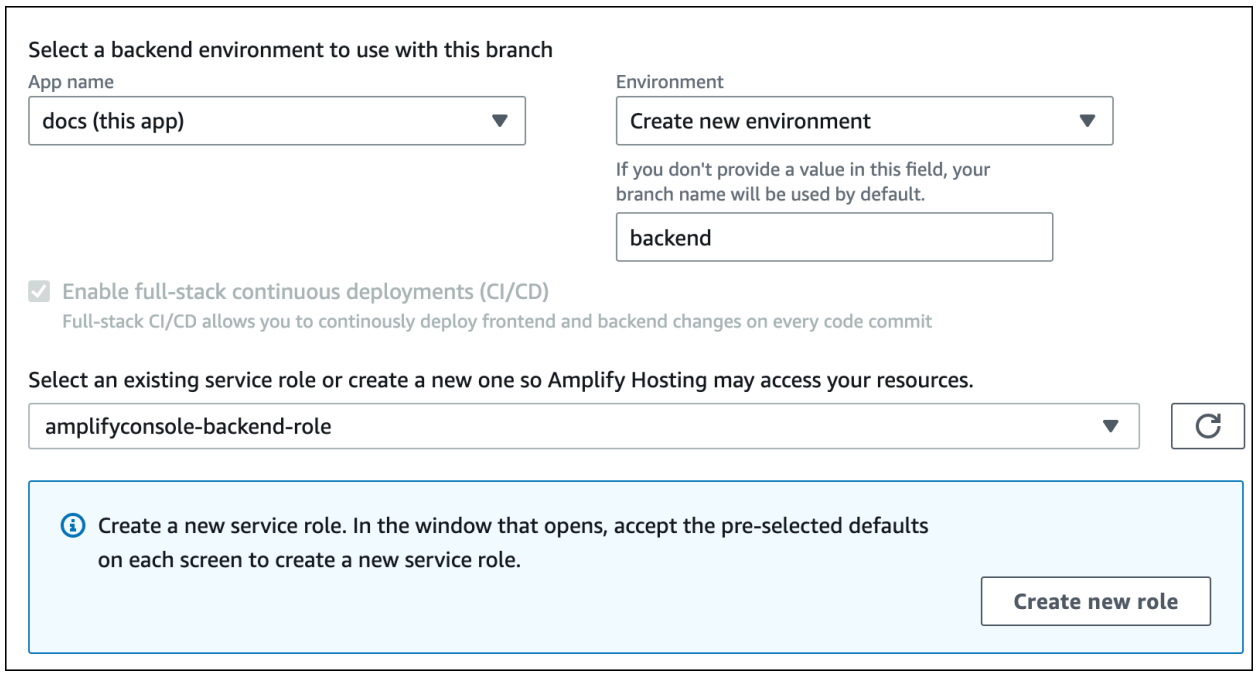
6. Wählen Sie Speichern.

Erstellen Sie eine neue Backend-Umgebung mit Authentifizierungsparametern für die Anmeldung über soziale Netzwerke

Um eine Filiale mit einer App zu verbinden

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Das Verfahren zum Verbinden einer Filiale mit einer App hängt davon ab, ob Sie eine Filiale mit einer neuen App oder einer vorhandenen App verbinden.
   - Eine Filiale mit einer neuen App verbinden

     a. Suchen Sie auf der Seite mit den Build-Einstellungen den Abschnitt Wählen Sie eine Backend-Umgebung aus, die mit diesem Zweig verwendet werden soll. Wählen Sie für Umgebung die Option Neue Umgebung erstellen aus und geben Sie den Namen Ihrer Backend-Umgebung ein. Der folgende Screenshot zeigt den Abschnitt Wählen Sie eine Backend-Umgebung aus, die mit diesem Zweig verwendet werden soll auf der Seite mit den Build-Einstellungen, in dem für die Backend-Umgebung ein Name **`backend`** eingegeben wurde.



b. Erweitern Sie auf der Seite mit den Build-Einstellungen den Abschnitt Erweiterte Einstellungen und fügen Sie Umgebungsvariablen für Social-Anmeldeschlüssel hinzu. **`AMPLIFY_FACEBOOK_CLIENT_SECRET`**Ist beispielsweise eine gültige Umgebungsvariable. Eine Liste der standardmäßig verfügbaren Amplify-Systemumgebungsvariablen finden Sie in Umgebungsvariablenreferenz Amplify der Tabelle unter.

- Einen Zweig mit einer vorhandenen App verbinden

  a. Wenn Sie eine neue Filiale mit einer vorhandenen App verbinden, legen Sie die Umgebungsvariablen für die Anmeldung über soziale Netzwerke fest, bevor Sie die Filiale verbinden. Wählen Sie im Navigationsbereich App-Einstellungen, Umgebungsvariablen aus.

  b. Wählen Sie im Abschnitt Umgebungsvariablen die Option Variablen verwalten aus.

  c. Wählen Sie im Abschnitt Variablen verwalten die Option Variable hinzufügen aus.

  d. Geben Sie unter Variable (Schlüssel) Ihre Client-ID ein. Geben Sie unter Value Ihr Client-Geheimnis ein.

  e. Wählen Sie „Speichern“.

Verwaltung von Umgebungsgeheimnissen

Mit der Veröffentlichung von Amplify Gen 2 wurde der Workflow für Umgebungsgeheimnisse optimiert, um die Verwaltung von Geheimnissen und Umgebungsvariablen in der Amplify-Konsole zu zentralisieren. Anweisungen zum Einrichten und Zugreifen auf Geheimnisse für eine Amplify Gen 2-App finden Sie unter Secrets and environment vars in der Amplify-Dokumentation.

Umgebungsgeheimnisse für eine Gen-1-App ähneln Umgebungsvariablen, es handelt sich jedoch um Schlüsselwertpaare für den AWS Systems Manager Parameterspeicher, die verschlüsselt werden können. Einige Werte müssen verschlüsselt werden, z. B. die Anmeldung mit dem privaten Apple-Schlüssel für Amplify.

Wird verwendet AWS Systems Manager , um Umgebungsgeheimnisse für eine Amplify Gen 1-Anwendung festzulegen

Verwenden Sie die folgenden Anweisungen, um mithilfe der AWS Systems Manager Konsole ein Umgebungsgeheimnis für eine Amplify-App der Generation 1 festzulegen.

Um ein Umgebungsgeheimnis festzulegen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Systems Manager Konsole.
2. Wählen Sie im Navigationsbereich Application Management und dann Parameter Store aus.
3. Wählen Sie auf der Seite AWS Systems Manager Parameter Store die Option Create parameter aus.
4. Gehen Sie auf der Seite Parameter erstellen im Abschnitt Parameterdetails wie folgt vor:
    a. Geben Sie für Name einen Parameter im Format ein**`/amplify/{your_app_id}/{your_backend_environment_name}/{your_parameter_name}`**.
    b. Wählen Sie für Type (Typ) die Option SecureString aus.
    c. Wählen Sie unter KMS-Schlüsselquelle die Option Mein aktuelles Konto aus, um den Standardschlüssel für Ihr Konto zu verwenden.
    d. Geben Sie unter Wert Ihren geheimen Wert für die Verschlüsselung ein.
5. Wählen Sie „Parameter erstellen“.

> **Note**
>
> Amplify hat nur Zugriff auf die Schlüssel unter dem Build `/amplify/{your_app_id}/{your_backend_environment_name}` für die spezifische Umgebung. Sie müssen die Standardeinstellung angeben AWS KMS key , damit Amplify den Wert entschlüsseln kann.

Zugreifen auf Umgebungsgeheimnisse für eine Gen 1-Anwendung

Umgebungsgeheimnisse für eine Gen-1-Anwendung werden `process.env.secrets` als JSON-Zeichenfolge gespeichert.

Referenz zu Umgebungsgeheimnissen verstärken

Geben Sie einen Systems Manager Manager-Parameter im Format an`/amplify/{your_app_id}/{your_backend_environment_name}/AMPLIFY_SIWA_CLIENT_ID`.

Sie können die folgenden Umgebungsgeheimnisse verwenden, auf die standardmäßig in der Amplify-Konsole zugegriffen werden kann.

Variablenname	Description	Beispielwert
AMPLIFY_SIWA_CLIENT_ID	Die Anmeldung mit der Apple-Client-ID	`com.yourapp.auth`
AMPLIFY_SIWA_TEAM_ID	Das Melden Sie sich mit der Apple-Team-ID an	`ABCD123`
AMPLIFY_SIWA_KEY_ID	Die Anmeldung mit der Apple-Schlüssel-ID	`ABCD123`
AMPLIFY_SIWA_PRIVATE_KEY	Die Anmeldung mit dem privaten Apple-Schlüssel	-----BEGINNE MIT DEM PRIVATEN SCHLÜSSEL----- ****..... -----ENDE DES PRIVATEN SCHLÜSSELS-----

Benutzerdefinierte Header für eine Amplify-App einrichten

Mit benutzerdefinierten HTTP-Headern können Sie Header für jede HTTP-Antwort angeben. Response-Header können für Debugging-, Sicherheits- und Informationszwecke verwendet werden. Sie können Header in der Amplify-Konsole angeben oder indem Sie die `customHttp.yml` Datei einer App herunterladen, bearbeiten und im Stammverzeichnis des Projekts speichern. Die detaillierten Schritte finden Sie unter Benutzerdefinierte Header einrichten.

Bisher wurden benutzerdefinierte HTTP-Header für eine App entweder durch Bearbeitung der Build-Spezifikation (Buildspec) in der Amplify-Konsole oder durch Herunterladen und Aktualisieren der `amplify.yml` Datei und Speichern im Stammverzeichnis des Projekts angegeben. Wir empfehlen dringend, auf diese Weise angegebene benutzerdefinierte Header aus der Buildspec und der Datei zu migrieren. `amplify.yml` Detaillierte Anweisungen finden Sie unter Migrieren benutzerdefinierter Header aus der Build-Spezifikation und amplify.yml.

Themen

YAML-Referenz für benutzerdefinierte Header

Geben Sie benutzerdefinierte Header im folgenden YAML-Format an:

```
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'
```

Verwenden Sie für ein Monorepo das folgende YAML-Format:

```
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'
```

Wenn Sie Ihrer App benutzerdefinierte Header hinzufügen, geben Sie Ihre eigenen Werte für Folgendes an:

pattern

Benutzerdefinierte Header werden auf alle URL-Dateipfade angewendet, die dem Muster entsprechen.

Header

Definiert die Header, die dem Dateimuster entsprechen.

Schlüssel

Der Name des benutzerdefinierten Headers.

value

Der Wert des benutzerdefinierten Headers.

Weitere Informationen zu HTTP-Headern finden Sie in Mozillas Liste der HTTP-Header.

Benutzerdefinierte Header einrichten

Es gibt zwei Möglichkeiten, benutzerdefinierte HTTP-Header für eine Amplify-App anzugeben. Sie können Header in der Amplify-Konsole angeben oder Sie können Header angeben, indem Sie

die `customHttp.yml` Datei einer App herunterladen, bearbeiten und im Stammverzeichnis Ihres Projekts speichern.

Um benutzerdefinierte Header für eine App festzulegen und sie in der Konsole zu speichern

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie benutzerdefinierte Header festlegen möchten.
3. Wählen Sie im Navigationsbereich Hosting und anschließend Benutzerdefinierte Header aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Kopfzeilen die Option Bearbeiten aus.
5. Geben Sie im Fenster Benutzerdefinierte Header bearbeiten die Informationen für Ihre benutzerdefinierten Header im YAML-Format für benutzerdefinierte Header ein.

   a. Geben Sie für `pattern` das passende Muster ein.

   b. Geben Sie für `key` den Namen der benutzerdefinierten Kopfzeile ein.

   c. Geben Sie für `value` den Wert des benutzerdefinierten Headers ein.
6. Wählen Sie Speichern.
7. Stellen Sie die App erneut bereit, um die neuen benutzerdefinierten Header anzuwenden.
   - Navigieren Sie für eine CI/CD App zu dem Zweig, den Sie bereitstellen möchten, und wählen Sie Diese Version erneut bereitstellen aus. Sie können auch einen neuen Build von Ihrem Git-Repository aus ausführen.
   - Für eine App zur manuellen Bereitstellung stellen Sie die App erneut in der Amplify-Konsole bereit.

Um benutzerdefinierte Header für eine App festzulegen und sie im Stammverzeichnis Ihres Repositorys zu speichern

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie benutzerdefinierte Header festlegen möchten.
3. Wählen Sie im Navigationsbereich Hosting und anschließend Benutzerdefinierte Header aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Header die Option YML herunterladen aus.
5. Öffnen Sie die heruntergeladene `customHttp.yml` Datei im Code-Editor Ihrer Wahl und geben Sie die Informationen für Ihre benutzerdefinierten Header im YAML-Format für benutzerdefinierte Header ein.

   a. Geben Sie für `pattern` das passende Muster ein.

b. Geben Sie für `key` den Namen der benutzerdefinierten Kopfzeile ein.

c. Geben Sie für `value` den Wert des benutzerdefinierten Headers ein.

6. Speichern Sie die bearbeitete `customHttp.yml` Datei im Stammverzeichnis Ihres Projekts. Wenn Sie mit einem Monorepo arbeiten, speichern Sie die `customHttp.yml` Datei im Stammverzeichnis Ihres Repos.
7. Stellen Sie die App erneut bereit, um die neuen benutzerdefinierten Header anzuwenden.
   - Führen Sie für eine CI/CD App einen neuen Build aus Ihrem Git-Repository aus, der die neue `customHttp.yml` Datei enthält.
   - Für eine App mit manueller Bereitstellung stellen Sie die App erneut in der Amplify-Konsole bereit und fügen Sie die neue `customHttp.yml` Datei mit den Artefakten hinzu, die Sie hochladen.

> **Note**
>
> Benutzerdefinierte Header, die in der `customHttp.yml` Datei festgelegt und im Stammverzeichnis der App bereitgestellt werden, überschreiben benutzerdefinierte Header, die im Abschnitt Benutzerdefinierte Header in der Amplify-Konsole definiert sind.

Beispiel für benutzerdefinierte Sicherheits-Header

Benutzerdefinierte Sicherheitsheader ermöglichen die Durchsetzung von HTTPS, die Verhinderung von XSS-Angriffen und den Schutz Ihres Browsers vor Clickjacking. Verwenden Sie die folgende YAML-Syntax, um benutzerdefinierte Sicherheitsheader auf Ihre App anzuwenden.

```
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'"
```

Benutzerdefinierte Cache-Control-Header einrichten

Apps, die mit Amplify gehostet werden, berücksichtigen die vom Ursprung gesendeten `Cache-Control` Header, es sei denn, Sie überschreiben sie mit benutzerdefinierten Headern, die Sie definieren. Amplify wendet benutzerdefinierte Cache-Control-Header nur für erfolgreiche Antworten mit einem Statuscode an. `200 OK` Dadurch wird verhindert, dass Fehlerantworten zwischengespeichert und anderen Benutzern bereitgestellt werden, die dieselbe Anfrage stellen.

Sie können die `s-maxage` Direktive manuell anpassen, um mehr Kontrolle über die Leistung und Bereitstellungsverfügbarkeit Ihrer App zu haben. Um beispielsweise die Dauer zu verlängern, für die Ihre Inhalte am Edge zwischengespeichert bleiben, können Sie die Gültigkeitsdauer (Time to Live, TTL) manuell erhöhen, indem Sie `s-maxage` auf einen Wert aktualisieren, der länger als die Standardeinstellung 600 Sekunden (10 Minuten) ist.

Verwenden Sie das folgende YAML-Format`s-maxage`, um einen benutzerdefinierten Wert für anzugeben. In diesem Beispiel wird der zugehörige Inhalt 3600 Sekunden (eine Stunde) lang am Edge zwischengespeichert.

```
customHeaders:
  - pattern: '/img/*'
    headers:
      - key: 'Cache-Control'
        value: 's-maxage=3600'
```

Weitere Informationen zur Steuerung der Anwendungsleistung mithilfe von Headern finden Sie unter. Verwenden des Cache-Control-Headers zur Steigerung der App-Leistung

Migrieren benutzerdefinierter Header aus der Build-Spezifikation und amplify.yml

Bisher wurden benutzerdefinierte HTTP-Header für eine App entweder durch Bearbeitung der Build-Spezifikation in der Amplify-Konsole oder durch Herunterladen und Aktualisieren der `amplify.yml` Datei und Speichern im Stammverzeichnis des Projekts angegeben. Es wird dringend empfohlen, dass Sie Ihre benutzerdefinierten Header aus der Build-Spezifikation und der Datei migrieren. `amplify.yml`

Geben Sie Ihre benutzerdefinierten Header im Abschnitt Benutzerdefinierte Header der Amplify-Konsole an oder indem Sie die Datei herunterladen und bearbeiten. `customHttp.yml`

Um benutzerdefinierte Header zu migrieren, die in der Amplify-Konsole gespeichert sind

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die die benutzerdefinierte Header-Migration durchgeführt werden soll.
3. Wählen Sie im Navigationsbereich Hosting, Build-Einstellungen aus. Im Abschnitt App-Build-Spezifikation können Sie die Buildspec Ihrer App überprüfen.
4. Wählen Sie Herunterladen, um eine Kopie Ihrer aktuellen Buildspec zu speichern. Sie können später auf diese Kopie verweisen, wenn Sie Einstellungen wiederherstellen müssen.
5. Wenn der Download abgeschlossen ist, wählen Sie Bearbeiten.
6. Notieren Sie sich die benutzerdefinierten Header-Informationen in der Datei, da Sie sie später in Schritt 9 verwenden werden. Löschen Sie im Bearbeitungsfenster alle benutzerdefinierten Header aus der Datei und wählen Sie Speichern.
7. Wählen Sie im Navigationsbereich Hosting, Benutzerdefinierte Header aus.
8. Wählen Sie auf der Seite Benutzerdefinierte Header die Option Bearbeiten aus.
9. Geben Sie im Fenster Benutzerdefinierte Kopfzeilen bearbeiten die Informationen für Ihre benutzerdefinierten Kopfzeilen ein, die Sie in Schritt 6 gelöscht haben.
10. Wählen Sie Speichern.
11. Stellen Sie alle Branches erneut bereit, auf die die neuen benutzerdefinierten Header angewendet werden sollen.

Um benutzerdefinierte Header von amplify.yml zu CustomHttp.yml zu migrieren

1. Navigieren Sie zu der `amplify.yml` Datei, die derzeit im Stammverzeichnis Ihrer App bereitgestellt ist.
2. Öffnen Sie `amplify.yml` im Code-Editor Ihrer Wahl.
3. Notieren Sie sich die benutzerdefinierten Header-Informationen in der Datei, da Sie sie später in Schritt 8 verwenden werden. Löschen Sie die benutzerdefinierten Header in der Datei. Speichern und schließen Sie die Datei.
4. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
5. Wählen Sie die App aus, für die Sie benutzerdefinierte Header festlegen möchten.

6. Wählen Sie im Navigationsbereich Hosting, Benutzerdefinierte Header aus.
7. Wählen Sie auf der Seite Benutzerdefinierte Header die Option Herunterladen aus.
8. Öffnen Sie die heruntergeladene `customHttp.yml` Datei im Code-Editor Ihrer Wahl und geben Sie die Informationen für Ihre benutzerdefinierten Header ein, aus denen Sie `amplify.yml` in Schritt 3 gelöscht haben.
9. Speichern Sie die bearbeitete `customHttp.yml` Datei im Stammverzeichnis Ihres Projekts. Wenn Sie mit einem Monorepo arbeiten, speichern Sie die Datei im Stammverzeichnis Ihres Repos.
10. Stellen Sie die App erneut bereit, um die neuen benutzerdefinierten Header anzuwenden.
    - Führen Sie für eine CI/CD App einen neuen Build aus Ihrem Git-Repository aus, der die neue `customHttp.yml` Datei enthält.
    - Für eine App mit manueller Bereitstellung stellen Sie die App erneut in der Amplify-Konsole bereit und fügen Sie die neue `customHttp.yml` Datei mit Artefakten hinzu, die Sie hochladen.

> **Note**
>
> Benutzerdefinierte Header, die in der `customHttp.yml` Datei festgelegt und im Stammverzeichnis der App bereitgestellt werden, überschreiben die benutzerdefinierten Header, die im Abschnitt Benutzerdefinierte Header der Amplify-Konsole definiert sind.

Anforderungen an benutzerdefinierte Monorepo-Header

Wenn Sie benutzerdefinierte Header für eine App in einem Monorepo angeben, beachten Sie die folgenden Einrichtungsanforderungen:

- Es gibt ein spezielles YAML-Format für ein Monorepo. Die korrekte Syntax finden Sie unter. YAML-Referenz für benutzerdefinierte Header
- Sie können benutzerdefinierte Header für eine Anwendung in einem Monorepo mithilfe des Abschnitts Benutzerdefinierte Header der Amplify-Konsole angeben. Sie müssen Ihre Anwendung erneut bereitstellen, um die neuen benutzerdefinierten Header anzuwenden.
- Als Alternative zur Verwendung der Konsole können Sie benutzerdefinierte Header für eine App in einem Monorepo in einer Datei angeben. `customHttp.yml` Sie müssen die `customHttp.yml` Datei im Stammverzeichnis Ihres Repos speichern und dann die Anwendung erneut bereitstellen,

um die neuen benutzerdefinierten Header anzuwenden. In der `customHttp.yml` Datei angegebene benutzerdefinierte Header überschreiben alle benutzerdefinierten Header, die im Abschnitt Benutzerdefinierte Header der Amplify-Konsole angegeben wurden.

Verwaltung der Cache-Konfiguration für eine App

Amplify verwendet Amazon CloudFront , um die Caching-Konfiguration für Ihre gehosteten Anwendungen zu verwalten. Auf jede App wird eine Cache-Konfiguration angewendet, um die beste Leistung zu erzielen.

Am 13. August 2024 veröffentlichte Amplify Verbesserungen der Caching-Effizienz für Anwendungen. Weitere Informationen finden Sie unter Verbesserungen des CDN-Cachings für eine bessere App-Leistung beim Hosting. AWS Amplify

Die folgende Tabelle fasst die Amplify-Unterstützung für bestimmte Caching-Verhaltensweisen vor und nach der Veröffentlichung der Caching-Verbesserungen zusammen.

Caching-Verhalten	Bisherige Unterstützung	Mit Verbesserungen beim Zwischenspeichern
Sie können benutzerdefinierte Header für eine App in der Amplify-Konsole oder in einer `customHeaders.yaml` Datei hinzufügen. Einer der Header, die Sie überschreiben können, ist. `Cache-Control` Weitere Informationen finden Sie unter Benutzerdefinierte Header für eine Amplify-App einrichten.	Ja	Ja
Amplify respektiert die `Cache-Control` Header, die Sie in einer `customHeaders.yaml` Datei definieren, und sie haben Vorrang vor den Standard-Cache-Einstellungen von Amplify.	Ja	Ja

Caching-Verhalten	Bisherige Unterstützung	Mit Verbesserungen beim Zwischenspeichern
Amplify respektiert die `Cache-Control` Header, die im Framework einer Anwendung für dynamische Routen festgelegt sind (z. B. Next.js SSR-Routen). Wenn in der Datei der App ein `Cache-Control` Header festgeleg t ist, hat dieser Vorrang vor den Einstellungen in der `customHeaders.yaml` Datei. `next.config.js`	Ja	Ja
Bei jeder neuen CI/CD App-Bereitstellung wird der Cache geleert.	Ja	Ja
Sie können den Leistungs modus für eine App aktivieren.	Ja	Nein Die Einstellung für den Leistungsmodus ist in der Amplify-Konsole nicht mehr verfügbar. Sie können jedoch einen `Cache-Control` Header erstellen, der die `s-maxage` Direktive festlegt. Detaillierte Anweisungen finden Sie unter Verwenden des Cache-Control-Headers zur Steigerung der App-Leist ung.

In der folgenden Tabelle sind die Änderungen an den Standardwerten für bestimmte Cache-Einstellungen aufgeführt.

Cache-Einstellung	Vorheriger Standardwert	Standardwert mit Verbesserungen beim Zwischenspeichern
Cache-Dauer für statische Objekte	Zwei Sekunden	Ein Jahr
Cache-Dauer für Reverse-Proxy-Antworten	Zwei Sekunden	Null Sekunden (kein Caching)
Max. Lebenszeit (TTL)	Zehn Minuten	Ein Jahr

Weitere Informationen darüber, wie Amplify die Caching-Konfiguration bestimmt, die auf eine Anwendung angewendet werden soll, sowie Anweisungen zur Verwaltung der Cache-Schlüsselkonfiguration finden Sie in den folgenden Themen.

Themen

- Wie Amplify die Cache-Konfiguration auf eine App anwendet
- Verwaltung von Cache-Schlüssel-Cookies
- Verwenden des Cache-Control-Headers zur Steigerung der App-Leistung

Wie Amplify die Cache-Konfiguration auf eine App anwendet

Um das Caching für Ihre App zu verwalten, bestimmt Amplify die Art des Inhalts, der bereitgestellt wird, indem es den Plattformtyp der App und die Umschreibungsregeln untersucht. Für `Compute` Apps untersucht Amplify auch die Routing-Regeln im Bereitstellungsmanifest.

> **Note**
>
> Der Plattformtyp der App wird von Amplify Hosting während der Bereitstellung festgelegt. Eine SSG-App (statisch) ist auf den Plattformtyp eingestellt. `WEB` Eine SSR-App (Next.js 12 oder höher) ist auf den Plattformtyp eingestellt. `WEB_COMPUTE`

Amplify identifiziert die folgenden vier Inhaltstypen und wendet die angegebene verwaltete Cache-Richtlinie an.

Statisch

Der Inhalt, der von Apps mit der `WEB` Plattform bereitgestellt wird, oder die statischen Routen in einer `WEB_COMPUTE` App.

Dieser Inhalt verwendet die Amplify-StaticContent Cache-Richtlinie.

Bildoptimierung

Die Bilder, die von den `ImageOptimization` Routen in einer `WEB_COMPUTE` App bereitgestellt werden.

Dieser Inhalt verwendet die Amplify-ImageOptimization Cache-Richtlinie.

Datenverarbeitung

Der Inhalt, der von den `Compute` Routen in einer `WEB_COMPUTE` App bereitgestellt wird. Dies schließt alle serverseitig gerenderten (SSR) Inhalte ein.

Dieser Inhalt verwendet entweder die Amplify-Default oder die Amplify-DefaultNoCookies Cache-Richtlinie, je nachdem`cacheConfig.type`, welcher Wert auf Ihrem Amplify `App` festgelegt ist.

Reverse-Proxy

Der Inhalt, der von Pfaden bereitgestellt wird, die einer benutzerdefinierten Reverse-Proxy-Rewrite-Regel entsprechen. Weitere Informationen zum Erstellen dieser benutzerdefinierten Regel finden Sie Reverse-Proxy umschreiben im Kapitel Weiterleitungen verwenden.

Dieser Inhalt verwendet entweder die Amplify-Default oder die Amplify-DefaultNoCookies Cache-Richtlinie, je nachdem`cacheConfig.type`, welcher Wert auf Ihrem Amplify `App` festgelegt ist.

Grundlegendes zu den verwalteten Cache-Richtlinien von Amplify

Amplify verwendet die folgenden vordefinierten verwalteten Cache-Richtlinien, um die Standard-Cache-Konfiguration für Ihre gehosteten Anwendungen zu optimieren.

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

Amplify-Default Einstellungen für verwaltete Cache-Richtlinien

Diese Richtlinie in der Konsole anzeigen CloudFront

Diese Richtlinie wurde für die Verwendung mit einem Ursprung entwickelt, bei dem es sich um eine AWS Amplify-Web-App handelt.

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden
- Maximale TTL: 31536000 Sekunden (ein Jahr)
- Standard-TTL: 0 Sekunden
- Im Cache-Schlüssel enthaltene Header:
  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`
  - `Host`
- Im Cache-Schlüssel enthaltene Cookies: Alle Cookies sind enthalten.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Alle Abfragezeichenfolgen sind enthalten.
- Einstellung für komprimierte Objekte zwischenspeichern: Gzip und Brotli aktiviert.

Amplify — Einstellungen für DefaultNoCookies verwaltete Cache-Richtlinien

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie wurde für die Verwendung mit einem Ursprung entwickelt, bei dem es sich um eine AWS Amplify-Web-App handelt.

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden
- Maximale TTL: 31536000 Sekunden (ein Jahr)
- Standard-TTL: 0 Sekunden
- Im Cache-Schlüssel enthaltene Header:
  - `Authorization`
  - `Accept`

  - `CloudFront-Viewer-Country`
  - `Host`
- Im Cache-Schlüssel enthaltene Cookies: Es sind keine Cookies enthalten.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Alle Abfragezeichenfolgen sind enthalten.
- Einstellung für komprimierte Objekte zwischenspeichern: Gzip und Brotli aktiviert.

Amplify — Einstellungen für ImageOptimization verwaltete Cache-Richtlinien

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie wurde für die Verwendung mit einem Ursprung entwickelt, bei dem es sich um eine AWS Amplify-Web-App handelt.

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden
- Maximale TTL: 31536000 Sekunden (ein Jahr)
- Standard-TTL: 0 Sekunden
- Im Cache-Schlüssel enthaltene Header:
  - `Authorization`
  - `Accept`
  - `Host`
- Im Cache-Schlüssel enthaltene Cookies: Es sind keine Cookies enthalten.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Alle Abfragezeichenfolgen sind enthalten.
- Einstellung für komprimierte Objekte zwischenspeichern: Gzip und Brotli aktiviert.

Amplify — Einstellungen für StaticContent verwaltete Cache-Richtlinien

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie wurde für die Verwendung mit einem Ursprung entwickelt, bei dem es sich um eine AWS Amplify-Web-App handelt.

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden

- Maximale TTL: 31536000 Sekunden (ein Jahr)
- Standard-TTL: 0 Sekunden
- Im Cache-Schlüssel enthaltene Header:
  - `Authorization`
  - `Host`
- Im Cache-Schlüssel enthaltene Cookies: Es sind keine Cookies enthalten.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Es sind keine Abfragezeichenfolgen enthalten.
- Einstellung für komprimierte Objekte zwischenspeichern: Gzip und Brotli aktiviert.

Verwaltung von Cache-Schlüssel-Cookies

Wenn Sie Ihre App auf Amplify bereitstellen, können Sie wählen, ob Sie Cookies in den Cache-Schlüssel aufnehmen oder ausschließen möchten. In der Amplify-Konsole wird diese Einstellung auf der Seite Benutzerdefinierte Header und Cache mithilfe des Schalters Cache-Schlüsseleinstellungen angegeben. Detaillierte Anweisungen finden Sie unter Cookies in den Cache-Schlüssel einbeziehen oder aus dem Cache-Schlüssel ausschließen.

Cookies in den Cache-Schlüssel einbeziehen

Mit dieser Einstellung wählt Amplify automatisch eine optimale Cache-Konfiguration für Ihre App aus, basierend auf der Art des Inhalts, der bereitgestellt wird. Sie müssen diesen Cache-Konfigurationstyp explizit auswählen.

Wenn Sie das SDKs oder das verwenden AWS CLI, entspricht diese Einstellung der Einstellung `cacheConfig.type` auf `AMPLIFY_MANAGED` mit dem `CreateApp` oder `UpdateApp` APIs.

Schließt Cookies aus dem Cache-Schlüssel aus

Dies ist die Standard-Cache-Konfiguration. Diese Cache-Konfiguration ähnelt der `AMPLIFY_MANAGED` Konfiguration, mit der Ausnahme, dass sie alle Cookies aus dem Cache-Schlüssel ausschließt.

Wenn Sie sich dafür entscheiden, Cookies aus dem Cache-Schlüssel auszuschließen, kann dies zu einer besseren Cache-Leistung führen. Bevor Sie sich für diese Cache-Konfiguration entscheiden, sollten Sie sich jedoch überlegen, ob Ihre App Cookies zur Bereitstellung dynamischer Inhalte verwendet.

Wenn Sie das SDKs oder das verwenden AWS CLI, entspricht diese Einstellung dem Setzen von `cacheConfig.type` auf `AMPLIFY_MANAGED_NO_COOKIES` mit dem `CreateApp` Oder `UpdateApp` APIs.

Weitere Informationen zum Cache-Schlüssel finden Sie unter Den Cache-Schlüssel verstehen im Amazon CloudFront Developer Guide;.

Cookies in den Cache-Schlüssel einbeziehen oder aus dem Cache-Schlüssel ausschließen

Sie können die Cache-Schlüssel-Cookie-Konfiguration für eine App in der Amplify-Konsole oder im AWS CLI festlegen. SDKs

Gehen Sie wie folgt vor, um anzugeben, ob Cookies in den Cache-Schlüssel aufgenommen oder ausgeschlossen werden sollen, wenn Sie eine neue App mithilfe der Amplify-Konsole bereitstellen.

So legen Sie die Cache-Schlüssel-Cookie-Konfiguration fest, wenn Sie eine App auf Amplify bereitstellen

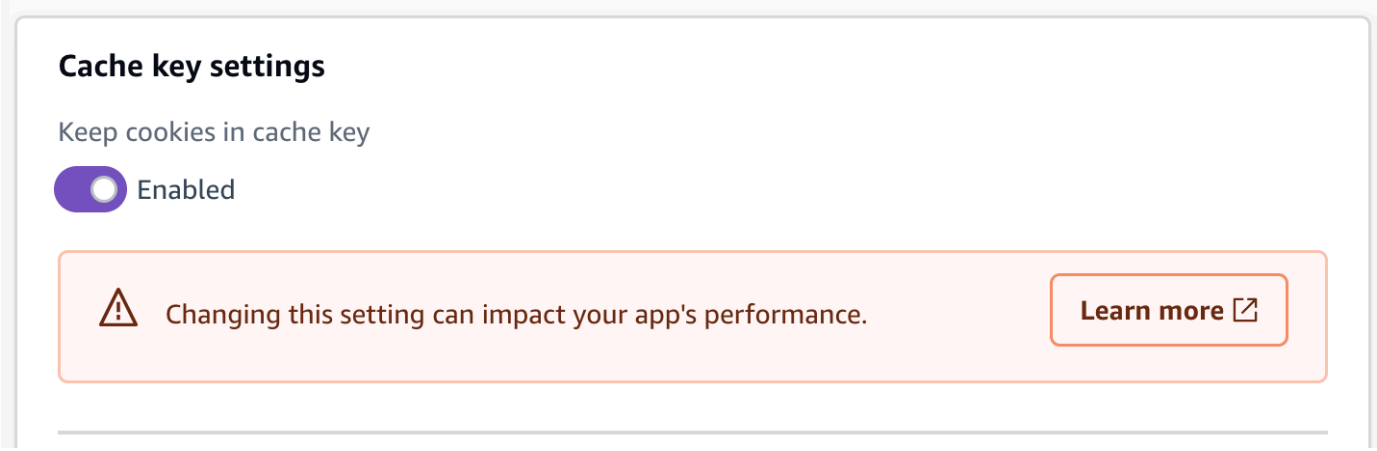
1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite „Alle Apps“ die Option Neue App erstellen aus.
3. Wählen Sie auf der Seite Start building with Amplify Ihren Git-Repository-Anbieter aus und klicken Sie dann auf Weiter.
4. Gehen Sie auf der Seite Repository-Zweig hinzufügen wie folgt vor:
   a. Wählen Sie den Namen des Repositorys aus, zu dem Sie eine Verbindung herstellen möchten.
   b. Wählen Sie den Namen des Repository-Zweigs aus, zu dem eine Verbindung hergestellt werden soll.
   c. Wählen Sie Weiter aus.
5. Wenn die App eine IAM-Servicerolle benötigt, können Sie entweder Amplify Hosting Compute erlauben, automatisch eine Servicerolle für Sie zu erstellen, oder Sie können eine von Ihnen erstellte Rolle angeben.
   - Um Amplify zu ermöglichen, automatisch eine Rolle zu erstellen und sie an Ihre App anzuhängen:
     - Wählen Sie Neue Servicerolle erstellen und verwenden aus.

- Um eine Servicerolle anzuhängen, die Sie zuvor erstellt haben:

  a. Wählen Sie Bestehende Servicerolle verwenden aus.

  b. Wählen Sie die zu verwendende Rolle aus der Liste aus.

6. Wählen Sie Erweiterte Einstellungen und suchen Sie dann den Abschnitt Cache-Schlüsseleinstellungen.
7. Wählen Sie entweder „Cookies im Cache-Schlüssel behalten“ oder „Cookies aus dem Cache-Schlüssel entfernen“. Der folgende Screenshot zeigt den Schalter mit den Einstellungen für den Cache-Schlüssel in der Konsole.



8. Wählen Sie Weiter aus.
9. Wählen Sie auf der Seite „Überprüfen“ die Option „Speichern und bereitstellen“.

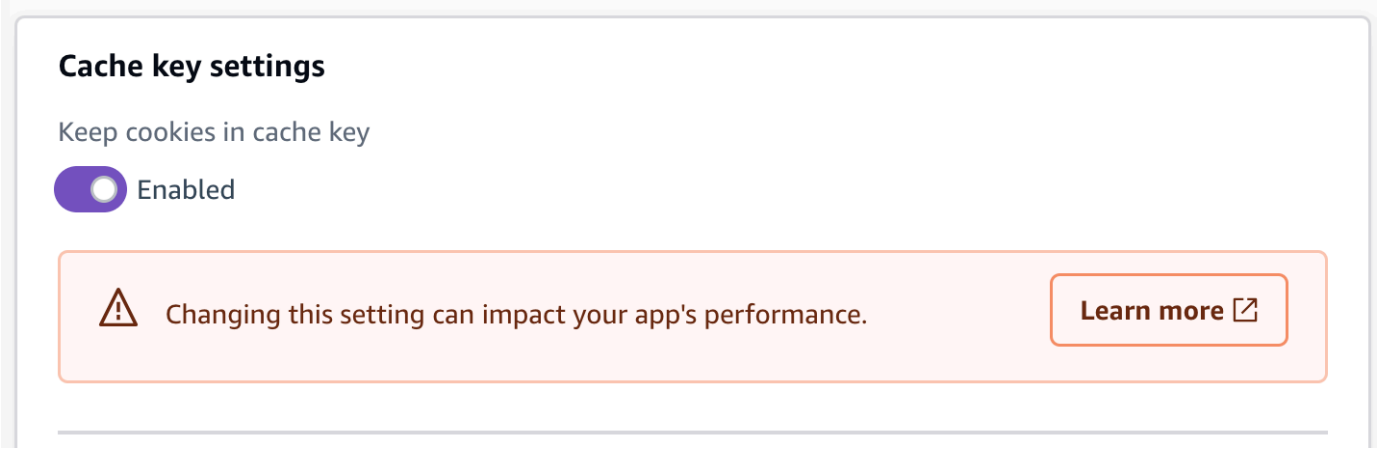
Änderung der Cache-Schlüssel-Cookie-Konfiguration für eine App

Sie können die Cache-Schlüssel-Cookie-Konfiguration für eine App ändern, die bereits für Amplify bereitgestellt wurde. Gehen Sie wie folgt vor, um zu ändern, ob Cookies für eine App mithilfe der Amplify-Konsole in den Cache-Schlüssel aufgenommen oder daraus ausgeschlossen werden sollen.

Um die Cookie-Konfiguration des Cache-Schlüssels für eine bereitgestellte App zu ändern

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite Alle Apps die Anwendung aus, die Sie aktualisieren möchten.
3. Wählen Sie im Navigationsbereich Hosting und dann Benutzerdefinierte Header und Cache aus.
4. Suchen Sie auf der Seite Benutzerdefinierte Header und Cache den Abschnitt Cache-Schlüsseleinstellungen und wählen Sie Bearbeiten aus.

5. Wählen Sie entweder Cookies im Cache-Schlüssel behalten oder Cookies aus Cache-Schlüssel entfernen. Der folgende Screenshot zeigt den Schalter mit den Einstellungen für den Cache-Schlüssel in der Konsole.



6. Wählen Sie Speichern.

Verwenden des Cache-Control-Headers zur Steigerung der App-Leistung

Die Standard-Hosting-Architektur von Amplify optimiert das Gleichgewicht zwischen Hosting-Leistung und Bereitstellungsverfügbarkeit. Für die meisten Kunden empfehlen wir, die Standardarchitektur zu verwenden.

Wenn Sie eine genauere Kontrolle über die Leistung einer App benötigen, können Sie den `Cache-Control` HTTP-Header manuell so einstellen, dass er die Hosting-Leistung optimiert, indem Inhalte für einen längeren Zeitraum am Rand des Content Delivery Network (CDN) zwischengespeichert werden.

Die `Cache-Control` HTTP-Header `max-age` und `s-maxage` -Direktiven wirken sich auf die Dauer des Inhalts-Cachings für Ihre App aus. Die `max-age` Direktive teilt dem Browser mit, wie lange (in Sekunden) Inhalte im Cache verbleiben sollen, bevor sie vom Ursprungsserver aktualisiert werden. Die `s-maxage` Direktive überschreibt `max-age` und ermöglicht es Ihnen, anzugeben, wie lange (in Sekunden) Inhalte am CDN-Edge verbleiben sollen, bevor sie vom Ursprungsserver aktualisiert werden.

Apps, die mit Amplify gehostet werden, berücksichtigen die vom Ursprung gesendeten `Cache-Control` Header, es sei denn, Sie überschreiben sie mit benutzerdefinierten Headern, die Sie definieren. Amplify wendet nur `Cache-Control` benutzerdefinierte Header für erfolgreiche

Antworten mit einem `200 OK` Statuscode an. Dadurch wird verhindert, dass Fehlerantworten zwischengespeichert und anderen Benutzern, die dieselbe Anfrage stellen, zugestellt werden.

Sie können die `s-maxage` Direktive manuell anpassen, um mehr Kontrolle über die Leistung und Bereitstellungsverfügbarkeit Ihrer App zu haben. Um beispielsweise zu ändern, wie lange Ihre Inhalte am Edge zwischengespeichert bleiben, können Sie die Gültigkeitsdauer (Time to Live, TTL) manuell festlegen, indem Sie `s-maxage` auf einen anderen Wert als den Standardwert 31536000 Sekunden (ein Jahr) aktualisieren.

Sie können benutzerdefinierte Header für eine App im Abschnitt Benutzerdefinierte Header der Amplify-Konsole definieren. Ein Beispiel für das YAML Format finden Sie unter. Benutzerdefinierte Cache-Control-Header einrichten

Gehen Sie wie folgt vor, um die `s-maxage` Direktive so einzustellen, dass Inhalte 24 Stunden lang am CDN-Edge zwischengespeichert werden.

So legen Sie einen benutzerdefinierten Header fest Cache-Control

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie benutzerdefinierte Header festlegen möchten.
3. Wählen Sie im Navigationsbereich Hosting, Benutzerdefinierte Header aus.
4. Wählen Sie auf der Seite Benutzerdefinierte Header die Option Bearbeiten aus.
5. Geben Sie im Fenster Benutzerdefinierte Kopfzeilen bearbeiten die Informationen für Ihre benutzerdefinierte Kopfzeile wie folgt ein:

   a. Geben Sie für`pattern`, **`**/*`** für alle Pfade ein.

   b. Geben Sie unter `key` den Wert **`Cache-Control`** ein.

   c. Geben Sie unter `value` den Wert **`s-maxage=86400`** ein.
6. Wählen Sie Speichern.
7. Stellen Sie die App erneut bereit, um den neuen benutzerdefinierten Header anzuwenden.

Skew-Schutz für Amplify-Bereitstellungen

Für Amplify-Anwendungen steht ein Deployment Skew-Schutz zur Verfügung, um Probleme mit Versionsverzerrungen zwischen Client und Servern in Webanwendungen zu vermeiden. Wenn Sie einen Skew-Schutz auf eine Amplify-Anwendung anwenden, können Sie sicherstellen, dass Ihre Clients immer mit der richtigen Version der serverseitigen Ressourcen interagieren, unabhängig davon, wann eine Bereitstellung erfolgt.

Versionsverzerrung ist eine häufige Herausforderung für Webentwickler. Es tritt auf, wenn ein Webbrowser eine veraltete Version einer Anwendung ausführt und auf dem Server eine neue Version ausgeführt wird. Diese Diskrepanz kann zu unvorhersehbarem Verhalten und Fehlern führen und die Benutzererfahrung der Anwendung beeinträchtigen. Die Funktion Amplify Deployment Skew Protection verknüpft Clients, die in Webbrowsern ausgeführt werden, mit einer bestimmten Bereitstellung. Dadurch wird sichergestellt, dass Amplify immer die Ressourcen für diese spezielle Bereitstellung bereitstellt und Client und Server synchronisiert bleiben.

Mit der Skew-Protection-Funktion von Amplify können Fehler für die Benutzer Ihrer Anwendung reduziert werden, wenn Sie neue Bereitstellungen veröffentlichen. Es kann auch das Entwicklererlebnis verbessern, indem der Zeitaufwand für die Behebung von Abwärts- und Vorwärtskompatibilitätsproblemen reduziert wird.

Details zu den Schutzfunktionen verzerren:

Unterstützte Anwendungstypen

Sie können statischen und SSR-Anwendungen, die mit einem beliebigen Framework erstellt wurden, das Amplify unterstützt, einen Schiefschutz hinzufügen. Anwendungen können aus einem Git-Repository oder einer manuellen Bereitstellung bereitgestellt werden.

Sie können einer Anwendung, die auf der `WEB_DYNAMIC` Plattform bereitgestellt wird (Next.js Version 11 oder früher), keinen Schiefschutz hinzufügen.

Dauer

Für statische Anwendungen bietet Amplify Bereitstellungen für eine Woche an. Für SSR-Anwendungen garantieren wir einen Skew-Schutz für bis zu acht frühere Bereitstellungen.

Cost (Kosten)

Es fallen keine zusätzlichen Kosten für das Hinzufügen von Schräglagenschutz zu einer Anwendung an.

Berücksichtigung der Leistung

Wenn der Skew-Schutz für eine Anwendung aktiviert ist, muss Amplify seine CDN-Cache-Konfigurationen aktualisieren. Daher sollten Sie damit rechnen, dass Ihre erste Bereitstellung nach der Aktivierung des Skew-Schutzes bis zu zehn Minuten dauert.

Topics

Konfiguration des Deployment Skew Protection für eine Amplify-Anwendung

Sie können den Deployment Skew-Schutz für eine Anwendung mithilfe der Amplify-Konsole, der oder der AWS Command Line Interface hinzufügen oder entfernen. SDKs Die Funktion wird auf Zweigstellenebene angewendet. Nur neue Bereitstellungen, die nach der Aktivierung des Schieferschutzes für eine Filiale vorgenommen werden, werden durch einen Schiefschutz geschützt.

Verwenden Sie die Felder und, um den Deployment Skew-Schutz mithilfe der Option oder hinzuzufügen AWS CLI oder SDKs zu entfernen. `CreateBranch.enableSkewProtection` `UpdateBranch.enableSkewProtection` Weitere Informationen finden Sie UpdateBranchin CreateBranchund in der Amplify API-Referenzdokumentation.

Wenn Sie ein bestimmtes Deployment entfernen möchten, sodass es nicht mehr bereitgestellt wird, verwenden Sie die `DeleteJob` API. Weitere Informationen finden Sie DeleteJobin der Amplify API-Referenzdokumentation.

Derzeit können Sie den Skew-Schutz nur für eine Anwendung aktivieren, die bereits auf Amplify Hosting bereitgestellt ist. Verwenden Sie die folgenden Anweisungen, um einem Zweig mithilfe der Amplify-Konsole einen Schiefschutz hinzuzufügen.

Aktivieren Sie den Skew-Schutz für den Zweig einer Amplify-Anwendung

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite Alle Apps den Namen der bereitgestellten App aus, für die der Skew-Schutz aktiviert werden soll.

3. Wählen Sie im Navigationsbereich App-Einstellungen und dann Branch-Einstellungen aus.
4. Wählen Sie im Abschnitt Filialen den Namen der Filiale aus, die aktualisiert werden soll.
5. Wählen Sie im Menü Aktionen die Option Schiefschutz aktivieren aus.
6. Wählen Sie im Bestätigungsfenster die Option Bestätigen aus. Der Schiefschutz ist jetzt für die Filiale aktiviert.
7. Stellen Sie Ihren Anwendungszweig erneut bereit. Nur Bereitstellungen, die nach der Aktivierung des Schiefschutzes vorgenommen werden, sind verzerrungsgeschützt.

Verwenden Sie die folgenden Anweisungen, um mithilfe der Amplify-Konsole den Skew-Schutz aus einem Zweig einer Anwendung zu entfernen.

Entfernen Sie den Schiefschutz aus einem Zweig einer Amplify-Anwendung

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite „Alle Apps“ den Namen der bereitgestellten App aus, für die Sie den Schiefschutz entfernen möchten.
3. Wählen Sie im Navigationsbereich App-Einstellungen und dann Branch-Einstellungen aus.
4. Wählen Sie im Abschnitt Filialen den Namen der Filiale aus, die aktualisiert werden soll.
5. Wählen Sie im Menü Aktionen die Option Schiefschutz deaktivieren aus. Der Schräglagenschutz ist jetzt für den Branch deaktiviert und es werden nur die neuesten Inhalte bereitgestellt.

So funktioniert der Skew-Schutz

In den meisten Fällen entspricht das Standardverhalten des _dpl-Cookies Ihren Anforderungen an den Schutz vor Verzerrungen. In den folgenden erweiterten Szenarien ist es jedoch besser, den Skew-Schutz mithilfe der `X-Amplify-Dpl` Header- und Abfrageparameter zu aktivieren. `dpl`

- Gleichzeitiges Laden Ihrer Website in mehreren Browser-Tabs
- Einsatz von Servicemitarbeitern

Amplify bewertet die eingehende Anfrage in der folgenden Reihenfolge, wenn es darum geht, welche Inhalte dem Kunden zur Verfügung gestellt werden sollen:

1. **`X-Amplify-Dpl`**header — Anwendungen können diesen Header verwenden, um Anfragen an eine bestimmte Amplify-Bereitstellung weiterzuleiten. Dieser Anforderungsheader kann mithilfe des Werts von `process.env.AWS_AMPLIFY_DEPLOYMENT_ID` festgelegt werden.
2. **`dpl`**Abfrageparameter — Anwendungen mit Next.js setzen automatisch den Abfrageparameter _dpl für Anfragen an Objekte mit Fingerabdruck (.js- und .css-Dateien).
3. _dpl-Cookie — Dies ist die Standardeinstellung für alle Anwendungen, die durch Schräglagen geschützt sind. Für einen bestimmten Browser wird dasselbe Cookie für jeden Browser-Tab oder jede Browser-Instanz gesendet, die mit einer Domain interagiert.

   Beachten Sie, dass das _dpl-Cookie von allen Tabs gemeinsam genutzt wird, wenn verschiedene Browser-Tabs unterschiedliche Versionen einer Website geladen haben. In diesem Szenario ist es nicht möglich, mit dem _dpl-Cookie einen vollständigen Schutz vor Verzerrungen zu erreichen, und Sie sollten erwägen, den Header für den `X-Amplify-Dpl` Schutz vor Verzerrungen zu verwenden.

X-Amplify-Dpl Beispiel für einen Header

Das folgende Beispiel zeigt den Code für eine SSR-Seite von Next.js, die über den Header auf den Skew-Schutz zugreift. `X-Amplify-Dpl` Die Seite rendert ihren Inhalt auf der Grundlage einer ihrer API-Routen. Das Deployment, das auf der API-Route bereitgestellt werden soll, wird mithilfe des `X-Amplify-Dpl` Headers angegeben, der auf den Wert von `process.env.AWS_AMPLIFY_DEPLOYMENT_ID` gesetzt ist.

```
import { useEffect, useState } from 'react';

export default function MyPage({deploymentId}) {
    const [data, setData] = useState(null);

    useEffect(() => {
        fetch('/api/hello', {
            headers: {
                'X-Amplify-Dpl': process.env.AWS_AMPLIFY_DEPLOYMENT_ID
            },
        })
        .then(res => res.json())
        .then(data => setData(data))
        .catch(error => console.error("error", error))
    }, []);
```

```
    return <div>
        {data ? JSON.stringify(data) : "Loading ... " }
    </div>
}
```

Überwachung einer Amplify-Anwendung

AWS Amplify bietet die folgenden Funktionen zur Überwachung Ihrer gehosteten Anwendungen:

- CloudWatch Metriken — Amplify sendet Metriken über Amazon aus, mit CloudWatch denen Sie Traffic, Fehler, Datenübertragung und Latenz für Ihre Anwendungen überwachen können.
- Zugriffsprotokolle — Amplify bietet Zugriffsprotokolle mit detaillierten Informationen zu Anfragen an Ihre Anwendung.
- CloudTrail Protokollierung — Amplify ist integriert und bietet eine Aufzeichnung der Aktionen, AWS CloudTrail die von einem Benutzer, einer Rolle oder einem AWS Dienst in Amplify ausgeführt wurden. Sie können diese Ereignisse in der CloudTrail Konsole einsehen.

Themen

Überwachung einer Amplify-Anwendung mit Amazon CloudWatch

AWS Amplify ist in Amazon integriert CloudWatch, sodass Sie die Metriken für Ihre Amplify-Anwendungen nahezu in Echtzeit überwachen und Alarme erstellen können, die Benachrichtigungen senden, wenn eine Metrik einen von Ihnen festgelegten Schwellenwert überschreitet. Weitere Informationen zur Funktionsweise des CloudWatch Service finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Unterstützte CloudWatch Metriken

Amplify unterstützt sieben CloudWatch Metriken im `AWS/AmplifyHosting` Namespace zur Überwachung von Verkehr, Fehlern, Datenübertragung, Latenz und Anforderungstoken für Ihre Apps. Diese Metriken werden in Intervallen von einer Minute aggregiert. CloudWatch Die Monitoring-Metriken sind kostenlos und werden nicht auf die CloudWatch Servicequoten angerechnet.

In der folgenden Tabelle werden alle unterstützten Metriken beschrieben und die relevantesten Statistiken aufgeführt. Nicht alle verfügbaren Statistiken gelten für jede Metrik.

Metrik	Description
Anforderungen	Die Gesamtzahl der Zuschaueranfragen, die Ihre App erhalten hat. Die relevanteste Statistik ist`Sum`. Verwenden Sie die `Sum` Statistik, um die Gesamtzahl der Anfragen zu ermitteln.
BytesDownloaded	Die Gesamtmenge der Daten, die von Zuschauern für, und `OPTIONS` Anfragen aus Ihrer App übertragen (heruntergeladen) wurden `GETHEAD`, in Byte. Die relevanteste Statistik ist`Sum`.
BytesUploaded	Die Gesamtmenge der in Ihre App übertrage nen (hochgeladenen) Daten in Byte für jede Anfrage, einschließlich Header. Amplify berechnet Ihnen keine Gebühren für Daten, die in Ihre Anwendungen hochgeladen wurden. Die relevanteste Statistik ist. `Sum`
4xxErrors	Die Anzahl der Anfragen, bei denen ein Fehler im Bereich des HTTP-Statuscodes 400-499 zurückgegeben wurde. Die relevanteste Statistik ist. `Sum` Verwenden Sie die `Sum` Statistik, um die Gesamthäufigkeit dieser Fehler zu ermitteln.
5xxErrors	Die Anzahl der Anfragen, bei denen ein Fehler im Bereich des HTTP-Statuscodes 500-599 zurückgegeben wurde.

Metrik	Description
	Die relevanteste Statistik ist. `Sum` Verwenden Sie die `Sum` Statistik, um die Gesamthäufigkeit dieser Fehler zu ermitteln.
Latenz	Die Zeit bis zum ersten Byte in Sekunden. Dies ist die Gesamtzeit zwischen dem Empfang einer Anfrage durch Amplify Hosting und der Rückgabe einer Antwort an das Netzwerk. Dies beinhaltet nicht die Netzwerklatenz, die auftritt, wenn eine Antwort das Gerät des Betrachters erreicht. Die relevantesten Statistiken sind `Average`,`Maximum`,`Minimum`,`p10`,`p50`,`p90`,`p95`, und`p100`. Verwenden Sie die `Average` Statistik, um die erwarteten Latenzen zu bewerten.

Metrik	Description
TokensConsumed	Die von Ihrer App verbrauchten Anforderungstoken. Die `Sum` Statistik stellt den gesamten Verbrauch von Anforderungstoken dar. Sie können diese Statistik mit Ihrem aktuellen `Request tokens per second` Servicekontingent vergleichen, um festzustellen, ob Sie eine Erhöhung des Kontingents beantragen müssen, um eine mögliche Drosselung bei einem future Ereignis mit hohem Datenverkehr zu vermeiden. Die `Average` Statistik zeigt den Verbrauch von Anforderungstoken in normalen Zeiten und zu Spitzenzeiten. Ein höherer Token-Verbrauch führt in der Regel zu einer längeren Zeit bis zum ersten Byte (TTFB). Daher können Sie diese Statistik bei der Bewertung der Latenz Ihrer Anwendung verwenden. Wenn Ihre Latenz gering ist, können Sie Ihren Downstream verbessern, APIs um Ihren Token-Verbrauch zu reduzieren und die Drosselung zu vermeiden, die auftreten kann, wenn der Token-Verbrauch die Servicequote Ihrer Anwendung überschreitet. `Request tokens per second` Weitere Informationen zum `Request tokens per second` Dienstkontingent finden Sie unter. Amplify Kontingente für Hosting-Dienste

Amplify bietet die folgenden CloudWatch metrischen Abmessungen.

Dimension	Description
App	Metrische Daten werden per App bereitgestellt.
AWS-Konto	Metrische Daten werden für alle Apps in der bereitgestellt AWS-Konto.

Zugriff auf CloudWatch Metriken

Mit dem folgenden Verfahren können Sie direkt von der Amplify-Konsole aus auf CloudWatch Metriken zugreifen.

> **Note**
>
> Sie können auch AWS-Managementkonsole unter https://console.aws.amazon.com/cloudwatch/auf CloudWatch Metriken zugreifen.

So greifen Sie in der Amplify-Konsole auf Metriken zu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie Metriken anzeigen möchten.
3. Wählen Sie im Navigationsbereich Monitoring und dann Metrics aus.

CloudWatch Alarme erstellen

Sie können in der Amplify-Konsole CloudWatch Alarme erstellen, die Benachrichtigungen senden, wenn bestimmte Kriterien erfüllt sind. Ein Alarm überwacht eine einzelne CloudWatch Metrik und sendet eine Amazon Simple Notification Service-Benachrichtigung, wenn die Metrik den Schwellenwert für eine bestimmte Anzahl von Bewertungszeiträumen überschreitet.

Sie können erweiterte Alarme erstellen, die metrische mathematische Ausdrücke verwenden, in der CloudWatch Konsole oder mit dem CloudWatch APIs. Sie können beispielsweise einen Alarm erstellen, der Sie benachrichtigt, wenn der Prozentsatz in drei aufeinanderfolgenden Perioden 15% 4xxErrors überschreitet. Weitere Informationen finden Sie unter Erstellen eines CloudWatch Alarms auf der Grundlage eines metrischen mathematischen Ausdrucks im CloudWatch Amazon-Benutzerhandbuch.

Für Alarme gelten die CloudWatch Standardpreise. Weitere Informationen finden Sie unter CloudWatchAmazon-Preise.

Gehen Sie wie folgt vor, um einen Alarm in der Amplify-Konsole zu erstellen.

Um einen CloudWatch Alarm für eine Amplify-Metrik zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie einen Alarm einrichten möchten.
3. Wählen Sie im Navigationsbereich Überwachung und dann Alarme aus.
4. Wählen Sie auf der Seite Alarme die Option Alarm erstellen aus.
5. Konfigurieren Sie Ihren Alarm im Fenster „Alarm erstellen“ wie folgt:

   a. Wählen Sie unter Metrik den Namen der zu überwachenden Metrik aus der Liste aus.

   b. Geben Sie unter Name des Alarms einen aussagekräftigen Namen für den Alarm ein. Wenn Sie beispielsweise Anfragen überwachen, könnten Sie dem Alarm einen Namen geben**`HighTraffic`**. Der Name darf nur ASCII-Zeichen enthalten.

   c. Gehen Sie für Benachrichtigungen einrichten wie folgt vor:

      - i. Wählen Sie Neu, um ein neues Amazon SNS SNS-Thema einzurichten.
        ii. Geben Sie unter E-Mail-Adresse die E-Mail-Adresse des Empfängers der Benachrichtigungen ein.
        iii. Wählen Sie Neue E-Mail-Adresse hinzufügen, um weitere Empfänger hinzuzufügen.
      - i. Wählen Sie Existing, um ein Amazon SNS SNS-Thema wiederzuverwenden.
        ii. Wählen Sie unter SNS-Thema den Namen eines vorhandenen Amazon SNS SNS-Themas aus der Liste aus.

   d. Stellen Sie für Whenever the Statistic of Metric die Bedingungen für Ihren Alarm wie folgt ein:

      i. Geben Sie an, ob die Metrik größer, kleiner oder gleich dem Schwellenwert sein muss.
      ii. Geben Sie den Schwellenwert an.
      iii. Geben Sie die Anzahl der aufeinanderfolgenden Evaluierungsperioden an, die sich im Alarmstatus befinden müssen, um den Alarm auszulösen.
      iv. Geben Sie die Dauer des Evaluierungszeitraums an.

   e. Wählen Sie Bestätigen aus.

> **Note**
>
> Jeder Amazon SNS SNS-Empfänger, den Sie angeben, erhält eine Bestätigungs-E-Mail von AWS Notifications. Die E-Mail enthält einen Link, dem der Empfänger folgen muss, um sein Abonnement zu bestätigen und Benachrichtigungen zu erhalten.

Zugreifen auf CloudWatch Protokolle für SSR-Apps

Amplify sendet Informationen über Ihre SSR-Laufzeit an Amazon CloudWatch Logs in Ihrem. AWS-Konto Wenn Sie eine SSR-App für Amplify Hosting Compute bereitstellen, benötigt die App eine IAM-Servicerolle, die Amplify übernimmt, wenn Sie andere Dienste in Ihrem Namen aufrufen. Sie können entweder Amplify Hosting Compute erlauben, automatisch eine Servicerolle für Sie zu erstellen, oder Sie können eine Rolle angeben, die Sie erstellt haben.

Wenn Sie Amplify erlauben, eine IAM-Rolle für Sie zu erstellen, verfügt die Rolle bereits über die Berechtigungen zum Erstellen CloudWatch von Protokollen. Wenn Sie Ihre eigene IAM-Rolle erstellen, müssen Sie Ihrer Richtlinie die folgenden Berechtigungen hinzufügen, damit Amplify auf Amazon CloudWatch Logs zugreifen kann.

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

Weitere Informationen zum Hinzufügen einer Servicerolle finden Sie unter. Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen Weitere Informationen zur Bereitstellung serverseitig gerenderter Apps finden Sie unterBereitstellung serverseitig gerenderter Anwendungen mit Amplify Hosting.

Sie können die Amplify Hosting-Rechenprotokolle für eine SSR-Anwendung in der CloudWatch Konsole oder in der Amplify-Konsole anzeigen. Verwenden Sie die folgenden Anweisungen, um die Protokolle in der Amplify-Konsole anzuzeigen.

Um CloudWatch Protokolle für eine SSR-Anwendung in der Amplify-Konsole anzuzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die SSR-App aus, für die Sie die CloudWatch Protokolle anzeigen möchten.

3. Wählen Sie im Navigationsbereich Monitoring und dann Hosting Compute Logs aus.
4. Suchen Sie auf der Seite Hosting-Rechenprotokolle nach einer CloudWatch Protokollgruppe für einen bestimmten Zweig und wählen Sie sie aus.

Abrufen und Analysieren von Zugriffsprotokollen für eine Amplify-Anwendung

Amplify speichert Zugriffsprotokolle für alle Apps, die Sie in Amplify hosten. Zugriffsprotokolle enthalten Informationen über Anfragen, die an Ihre gehosteten Apps gestellt werden. Amplify speichert alle Zugriffsprotokolle für eine App, bis Sie die App löschen. Alle Zugriffsprotokolle für eine App sind in der Amplify-Konsole verfügbar. Jede einzelne Anfrage nach Zugriffsprotokollen ist jedoch auf einen von Ihnen angegebenen Zeitraum von zwei Wochen begrenzt.

> **Warning**
>
> Geben Sie keine Geheimnisse, Anmeldeinformationen oder vertraulichen Daten URLs als Pfad- oder Abfrageparameter an. Diese Werte sind im Klartext in den Zugriffsprotokollen Ihrer Amplify-Anwendung sichtbar.

Amplify verwendet niemals CloudFront Distributionen zwischen Kunden wieder. Amplify erstellt CloudFront Distributionen im Voraus, sodass Sie bei der Bereitstellung einer neuen App nicht darauf warten müssen, dass eine CloudFront Distribution erstellt wird. Bevor diese Distributionen einer Amplify-App zugewiesen werden, erhalten sie möglicherweise Traffic von Bots. Sie sind jedoch so konfiguriert, dass sie immer als Nicht gefunden antworten, bevor sie zugewiesen werden. Wenn die Zugriffsprotokolle Ihrer App Einträge für einen Zeitraum enthalten, bevor Sie Ihre App erstellt haben, beziehen sich diese Einträge auf diese Aktivität.

> **Important**
>
> Wir empfehlen, dass Sie die Protokolle verwenden, um die Art der Anfragen für Ihre Inhalte zu verstehen, nicht als eine vollständige Buchführung aller Anfragen. Amplify liefert Zugriffsprotokolle nach bestem Wissen. Der Protokolleintrag für eine bestimmte Anfrage wird möglicherweise viel später übermittelt, als die Anfrage tatsächlich verarbeitet wurde; in seltenen Fällen kann es auch sein, dass ein Protokolleintrag gar nicht übermittelt wird. Wenn ein Protokolleintrag in den Zugriffsprotokollen weggelassen wird, entspricht die Anzahl der

Einträge in den Zugriffsprotokollen nicht der Nutzung, die in den AWS Abrechnungs- und Nutzungsberichten angegeben ist.

Die Zugriffsprotokolle einer App werden abgerufen

Gehen Sie wie folgt vor, um Zugriffsprotokolle für eine Amplify-App abzurufen.

Um Zugriffsprotokolle anzuzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, für die Sie die Zugriffsprotokolle anzeigen möchten.
3. Wählen Sie im Navigationsbereich Monitoring und anschließend Access Logs aus.
4. Wählen Sie „Zeitraum bearbeiten“.
5. Gehen Sie im Fenster Zeitraum bearbeiten wie folgt vor.
   a. Geben Sie als Startdatum den ersten Tag des zweiwöchigen Intervalls an, für das Protokolle abgerufen werden sollen.
   b. Wählen Sie unter Startzeit die Uhrzeit am ersten Tag aus, an dem der Protokollabruf gestartet werden soll.
   c. Wählen Sie Bestätigen aus.
6. Die Amplify-Konsole zeigt die Protokolle für den angegebenen Zeitraum im Abschnitt Zugriffsprotokolle an. Wählen Sie Herunterladen, um die Protokolle im CSV-Format zu speichern.

Analyse von Zugriffsprotokollen

Um Zugriffsprotokolle zu analysieren, können Sie die CSV-Dateien in einem Amazon S3 S3-Bucket speichern. Eine Möglichkeit, Ihre Zugriffsprotokolle zu analysieren, ist die Verwendung von Athena. Athena ist ein interaktiver Abfragedienst, mit dem Sie Daten für AWS Dienste analysieren können. Sie können den step-by-step Anweisungen hier folgen, um eine Tabelle zu erstellen. Sobald Ihre Tabelle erstellt wurde, können Sie Daten wie folgt abfragen.

```
SELECT SUM(bytes) AS total_bytes
FROM logs
WHERE "date" BETWEEN DATE '2018-06-09' AND DATE '2018-06-11'
LIMIT 100;
```

Protokollieren von Amplify-API-Aufrufen mit AWS CloudTrail

AWS Amplify ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS Dienstes in Amplify bereitstellt. CloudTrail erfasst alle API-Aufrufe für Amplify als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amplify-Konsole und Code-Aufrufe der Amplify-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amplify. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der CloudTrail gesammelten Informationen können Sie die Anfrage, die an Amplify gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

Amplify Sie Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto standardmäßig aktiviert. Wenn in Amplify Aktivitäten auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS -Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen für Amplify, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS -Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie in folgenden Themen im AWS CloudTrail -Benutzerhandbuch:

- Einen Trail für dein AWS Konto erstellen
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle Amplify-Operationen werden von der AWS Amplify Console API Reference, der AWS Amplify Admin UI Reference CloudTrail und der Amplify UI Builder API Reference protokolliert und dokumentiert. Beispielsweise generieren Aufrufe von `DeleteApp` und `DeleteBackendEnvironment` Operationen Einträge in den CloudTrail Protokolldateien. `CreateApp`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Wurde die Anfrage mit Root- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt?
- Wurde die Anfrage mit temporären Sicherheitsanmeldedaten für eine Rolle oder einen Verbundbenutzer gestellt?
- Wurde die Anfrage von einem anderen AWS Dienst gestellt.

Weitere Informationen finden Sie unter dem Element CloudTrail userIdentity im AWS CloudTrail Benutzerhandbuch.

Verstehen von Amplify-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der den `ListApps`Vorgang „ AWS Amplify Console API Reference“ demonstriert.

```
{
    "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"
}
```

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der den `ListBackendJobs`Vorgang AWS Amplify Admin UI API Reference demonstriert.

```
{
    "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"
}
```

Verwenden von IAM-Rollen mit Amplify-Anwendungen

Eine IAM-Rolle ist eine IAM-Identität mit bestimmten Berechtigungen. Die Berechtigungen der Rolle bestimmen, was die Identität tun kann und was nicht. AWS Sie können in Ihrem IAM-Rollen erstellen AWS-Konto und diese verwenden, um Berechtigungen an Amplify Hosting zu delegieren. Weitere Informationen zu Rollen finden Sie unter IAM-Rollen im IAM-Benutzerhandbuch.

Sie können die folgenden Arten von IAM-Rollen verwenden, um Amplify Hosting die erforderlichen Berechtigungen zu gewähren, um Aktionen in Ihrem Namen auszuführen oder Rechencode auszuführen, der auf andere Ressourcen zugreift. AWS

IAM-Servicerolle

: Amplify übernimmt diese Rolle, um Aktionen in Ihrem Namen durchzuführen. Diese Rolle ist für Anwendungen mit Backend-Ressourcen erforderlich.

IAM SSR-Rechenrolle

: Ermöglicht einer serverseitig gerenderten (SSR) Anwendung den sicheren Zugriff auf bestimmte Ressourcen. AWS

Rolle „IAM SSR-Protokolle“ CloudWatch

: Wenn Sie eine SSR-App bereitstellen, benötigt die App eine IAM-Servicerolle, die Amplify annimmt, um Amplify den Zugriff auf Amazon Logs zu ermöglichen. CloudWatch

Themen

Hinzufügen einer Servicerolle mit Berechtigungen zur Bereitstellung von Backend-Ressourcen

Amplify benötigt Berechtigungen, um Backend-Ressourcen mit Ihrem Frontend bereitzustellen. Dazu verwenden Sie eine Servicerolle. Eine Servicerolle ist die AWS Identity and Access Management (IAM) -Rolle, die Amplify Hosting die Erlaubnis gibt, Backends in Ihrem Namen bereitzustellen, zu erstellen und zu verwalten.

Wenn Sie eine neue App erstellen, für die eine IAM-Servicerolle erforderlich ist, können Sie entweder Amplify Hosting erlauben, automatisch eine Servicerolle für Sie zu erstellen, oder Sie können eine IAM-Rolle auswählen, die Sie bereits erstellt haben. In diesem Abschnitt erfahren Sie, wie Sie eine Amplify-Servicerolle erstellen, die über Kontoverwaltungsrechte verfügt und explizit direkten Zugriff auf Ressourcen ermöglicht, die Amplify-Anwendungen für die Bereitstellung, Erstellung und Verwaltung von Backends benötigen.

Eine Amplify-Servicerolle in der IAM-Konsole erstellen

So erstellen Sie eine Servicerolle

1. Öffnen Sie die IAM-Konsole und wählen Sie in der linken Navigationsleiste Rollen und anschließend Rolle erstellen aus.
2. Wählen Sie auf der Seite Vertrauenswürdige Entitäten auswählen die Option AWS -Service aus. Wählen Sie als Anwendungsfall Amplify — Backend Deployment und dann Weiter aus.
3. Wählen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) die Option Next (Weiter) aus.
4. Geben Sie auf der Seite Name, Ansicht und Erstellung für Rollenname einen aussagekräftigen Namen ein, z. B. **`AmplifyConsoleServiceRole-AmplifyRole`**
5. Akzeptieren Sie alle Standardeinstellungen und wählen Sie Create role aus.
6. Kehren Sie zur Amplify-Konsole zurück, um die Rolle an Ihre App anzuhängen.
   - Wenn Sie gerade dabei sind, eine neue App bereitzustellen, gehen Sie wie folgt vor:
     a. Aktualisieren Sie die Liste der Servicerollen.
     b. Wählen Sie die Rolle aus, die Sie gerade erstellt haben. In diesem Beispiel sollte sie wie AmplifyConsoleServiceRole- aussehenAmplifyRole.
     c. Wählen Sie Weiter und folgen Sie den Schritten, um Ihre App-Bereitstellung abzuschließen.
   - Wenn Sie bereits eine App haben, gehen Sie wie folgt vor:
     a. Wählen Sie im Navigationsbereich App-Einstellungen und dann IAM-Rollen aus.
     b. Wählen Sie auf der Seite mit den IAM-Rollen im Abschnitt Servicerolle die Option Bearbeiten aus.
     c. Wählen Sie auf der Seite „Servicerolle“ die Rolle aus, die Sie gerade erstellt haben, aus der Liste der Servicerollen.
     d. Wählen Sie Speichern.

7. Amplify hat jetzt die Erlaubnis, Backend-Ressourcen für Ihre App bereitzustellen.

Bearbeiten Sie die Vertrauensrichtlinie einer Servicerolle, um zu verhindern, dass der Stellvertreter verwirrt wird

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. Weitere Informationen finden Sie unter Serviceübergreifende Confused-Deputy-Prävention.

Derzeit erzwingt die standardmäßige Vertrauensrichtlinie für die `Amplify-Backend Deployment` Servicerolle die Schlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Kontextbedingungen, um zu verhindern, dass der Stellvertreter verwirrt wird. Wenn Sie jedoch zuvor eine `Amplify-Backend Deployment` Rolle in Ihrem Konto erstellt haben, können Sie die Vertrauensrichtlinie der Rolle aktualisieren, um diese Bedingungen hinzuzufügen, um vor verwirrtem Stellvertreter zu schützen.

Verwenden Sie das folgende Beispiel, um den Zugriff auf Apps in Ihrem Konto einzuschränken. Ersetzen Sie die Region und die Anwendungs-ID im Beispiel durch Ihre eigenen Informationen.

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
```

Anweisungen zum Bearbeiten der Vertrauensrichtlinie für eine Rolle mithilfe von finden Sie unter Ändern einer Rolle (Konsole) im IAM-Benutzerhandbuch. AWS-Managementkonsole

Hinzufügen einer SSR-Compute-Rolle, um den Zugriff auf Ressourcen zu ermöglichen AWS

Diese Integration ermöglicht es Ihnen, dem Amplify SSR Compute-Dienst eine IAM-Rolle zuzuweisen, damit Ihre serverseitig gerenderte (SSR) -Anwendung auf der Grundlage der Rollenberechtigungen sicher auf bestimmte AWS Ressourcen zugreifen kann. Beispielsweise

können Sie den SSR-Rechenfunktionen Ihrer App den sicheren Zugriff auf andere AWS Dienste oder Ressourcen wie Amazon Bedrock einen Amazon S3 S3-Bucket ermöglichen, basierend auf den in der zugewiesenen IAM-Rolle definierten Berechtigungen.

Die IAM-SSR-Rechenrolle stellt temporäre Anmeldeinformationen bereit, sodass langlebige Sicherheitsanmeldedaten in Umgebungsvariablen nicht fest codiert werden müssen. Die Verwendung der IAM SSR Compute-Rolle entspricht den bewährten AWS Sicherheitsmethoden, d. h. der Gewährung von Berechtigungen mit den geringsten Rechten und der Verwendung von kurzfristigen Anmeldeinformationen, wenn möglich.

In den Anweisungen weiter unten in diesem Abschnitt wird beschrieben, wie Sie eine Richtlinie mit benutzerdefinierten Berechtigungen erstellen und die Richtlinie einer Rolle zuordnen. Wenn Sie die Rolle erstellen, müssen Sie eine benutzerdefinierte Vertrauensrichtlinie anhängen, die Amplify die Erlaubnis erteilt, die Rolle zu übernehmen. Wenn die Vertrauensbeziehung nicht korrekt definiert ist, wird beim Versuch, die Rolle hinzuzufügen, eine Fehlermeldung angezeigt. Die folgende benutzerdefinierte Vertrauensrichtlinie gewährt Amplify die Erlaubnis, die Rolle zu übernehmen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Sie können eine IAM-Rolle in Ihrer AWS-Konto mit einer vorhandenen SSR-Anwendung verknüpfen, indem Sie die Amplify-Konsole verwenden AWS SDKs, oder die. AWS CLI Die Rolle, die Sie zuordnen, wird automatisch dem Amplify SSR-Rechendienst zugeordnet und gewährt ihm die von Ihnen angegebenen Berechtigungen für den Zugriff auf andere AWS Ressourcen. Da sich die

Anforderungen Ihrer Anwendung im Laufe der Zeit ändern, können Sie die zugeordnete IAM-Rolle ändern, ohne Ihre Anwendung erneut bereitstellen zu müssen. Dies bietet Flexibilität und reduziert Ausfallzeiten von Anwendungen.

> **Important**
>
> Sie sind dafür verantwortlich, Ihre Anwendung so zu konfigurieren, dass sie Ihre Sicherheits- und Compliance-Ziele erfüllt. Dazu gehört die Verwaltung Ihrer SSR-Compute-Rolle, die so konfiguriert sein sollte, dass sie über die Mindestberechtigungen verfügt, die zur Unterstützung Ihres Anwendungsfalls erforderlich sind. Weitere Informationen finden Sie unter Verwaltung der Sicherheit von IAM SSR Compute-Rollen.

Erstellen einer SSR-Compute-Rolle in der IAM-Konsole

Bevor Sie einer Amplify-Anwendung eine IAM SSR Compute-Rolle hinzufügen können, muss die Rolle bereits in Ihrer vorhanden sein. AWS-Konto In diesem Abschnitt erfahren Sie, wie Sie eine IAM-Richtlinie erstellen und sie einer Rolle zuordnen, die Amplify für den Zugriff auf bestimmte AWS Ressourcen übernehmen kann.

Wir empfehlen Ihnen, bei der Erstellung einer AWS IAM-Rolle die bewährten Methoden zur Gewährung von Berechtigungen mit den geringsten Rechten zu befolgen. Die IAM-SSR-Compute-Rolle wird nur von SSR-Rechenfunktionen aus aufgerufen und sollte daher nur die Berechtigungen gewähren, die für die Ausführung des Codes erforderlich sind.

Sie können das AWS-Managementkonsole, oder verwenden AWS CLI, um Richtlinien SDKs in IAM zu erstellen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien.

Die folgenden Anweisungen zeigen, wie Sie mit der IAM-Konsole eine IAM-Richtlinie erstellen, die die Berechtigungen definiert, die dem Amplify Compute-Dienst gewährt werden sollen.

Um den JSON-Richtlinien-Editor der IAM-Konsole zum Erstellen einer Richtlinie zu verwenden

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich auf der linken Seite Policies (Richtlinien).
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.

5. Geben oder fügen Sie ein JSON-Richtliniendokument ein.
6. Wenn Sie mit dem Hinzufügen von Berechtigungen zur Richtlinie fertig sind, wählen Sie Next (Weiter) aus.
7. Geben Sie auf der Seite Review and create (Überprüfen und erstellen) unter Name einen Namen und unter Description (Beschreibung) (optional) eine Beschreibung für die Richtlinie ein, die Sie erstellen. Überprüfen Sie Permissions defined in this policy (In dieser Richtlinie definierte Berechtigungen), um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden.
8. Wählen Sie Create policy (Richtlinie erstellen) aus, um Ihre neue Richtlinie zu speichern.

Nachdem Sie eine Richtlinie erstellt haben, folgen Sie den folgenden Anweisungen, um die Richtlinie einer IAM-Rolle zuzuordnen.

Um eine Rolle zu erstellen, die Amplify-Berechtigungen für bestimmte AWS Ressourcen gewährt

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie den Rollentyp Custom trust policy (Benutzerdefinierte Vertrauensrichtlinie).
4. Geben Sie im Abschnitt Benutzerdefinierte Vertrauensrichtlinie die benutzerdefinierte Vertrauensrichtlinie für die Rolle ein. Eine Rollenvertrauensrichtlinie ist erforderlich und definiert die Prinzipale, denen Sie vertrauen, dass sie die Rolle übernehmen.

   Kopieren Sie die folgende Vertrauensrichtlinie und fügen Sie sie ein, um dem Amplify-Dienst die Erlaubnis zu erteilen, diese Rolle zu übernehmen.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
```

```
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

5. Beheben Sie alle Sicherheitswarnungen, Fehler oder allgemeinen Warnungen, die während der Richtlinien-Validierung erzeugt wurden, und wählen Sie dann Next (Weiter) aus.
6. Suchen Sie auf der Seite „Berechtigungen hinzufügen“ nach dem Namen der Richtlinie, die Sie im vorherigen Verfahren erstellt haben, und wählen Sie sie aus. Klicken Sie anschließend auf Weiter.
7. Geben Sie für Rollenname einen Rollennamen ein. Rollennamen müssen innerhalb Ihres Unternehmens eindeutig sein AWS-Konto. Sie werden nicht nach Groß- und Kleinschreibung unterschieden. z. B. können Sie keine Rollen erstellen, die **`PRODROLE`** bzw. **`prodrole`** heißen. Da andere AWS Ressourcen möglicherweise auf die Rolle verweisen, können Sie den Namen der Rolle nicht bearbeiten, nachdem sie erstellt wurde.
8. (Optional) Geben Sie unter Role description (Rollenbeschreibung) eine Beschreibung für die neue Rolle ein.
9. (Optional) Wählen Sie Bearbeiten im Abschnitt Schritt 1: Vertrauenswürdige Entitäten auswählen oder Schritt 2: Berechtigungen hinzufügen, um die benutzerdefinierte Richtlinie und die Berechtigungen für die Rolle zu bearbeiten.
10. Prüfen Sie die Rolle und klicken Sie dann auf Rolle erstellen.

Hinzufügen einer IAM SSR Compute-Rolle zu einer Amplify-App

Nachdem Sie in Ihrem eine IAM-Rolle erstellt haben AWS-Konto, können Sie sie einer App in der Amplify-Konsole zuordnen.

Um einer App in der Amplify-Konsole eine SSR-Compute-Rolle hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite Alle Apps den Namen der App aus, zu der Sie eine Rechenrolle hinzufügen möchten.
3. Wählen Sie im Navigationsbereich App-Einstellungen und dann IAM-Rollen aus.
4. Wählen Sie im Abschnitt Rechenrolle die Option Bearbeiten aus.

5. Suchen Sie in der Liste Standardrolle nach dem Namen der Rolle, die Sie anhängen möchten, und wählen Sie sie aus. In diesem Beispiel können Sie den Namen der Rolle wählen, die Sie im vorherigen Verfahren erstellt haben. Standardmäßig wird die Rolle, die Sie auswählen, allen Zweigen Ihrer App zugeordnet.

   Wenn die Vertrauensbeziehung der Rolle nicht korrekt definiert ist, erhalten Sie eine Fehlermeldung und Sie können die Rolle nicht hinzufügen.
6. (optional) Wenn sich Ihre Anwendung in einem öffentlichen Repository befindet und die automatische Branch-Erstellung verwendet oder Webvorschauen für Pull-Requests aktiviert sind, empfehlen wir nicht, eine Rolle auf App-Ebene zu verwenden. Ordnen Sie die Rolle Compute stattdessen nur Branches zu, die Zugriff auf bestimmte Ressourcen benötigen. Gehen Sie wie folgt vor, um das Standardverhalten auf App-Ebene zu überschreiben und einer bestimmten Branche eine Rolle zuzuweisen:

   a. Wählen Sie unter Branch den Namen des Branches aus, den Sie verwenden möchten.
   b. Wählen Sie unter Rechenrolle den Namen der Rolle aus, die dem Zweig zugeordnet werden soll.
7. Wählen Sie „Speichern“.

Verwaltung der Sicherheit von IAM SSR Compute-Rollen

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Sie sind dafür verantwortlich, Ihre Anwendung so zu konfigurieren, dass sie Ihre Sicherheits- und Compliance-Ziele erfüllt. Dazu gehört die Verwaltung Ihrer SSR-Compute-Rolle, die so konfiguriert sein sollte, dass sie über die Mindestberechtigungen verfügt, die zur Unterstützung Ihres Anwendungsfalls erforderlich sind. Die Anmeldeinformationen für die von Ihnen angegebene SSR-Compute-Rolle sind sofort während der Laufzeit Ihrer SSR-Funktion verfügbar. Wenn Ihr SSR-Code diese Anmeldeinformationen entweder absichtlich, aufgrund eines Fehlers oder durch die Zulassung von Remote Code Execution (RCE) preisgibt, kann sich ein nicht autorisierter Benutzer Zugriff auf die SSR-Rolle und ihre Berechtigungen verschaffen.

Wenn eine Anwendung in einem öffentlichen Repository eine SSR-Compute-Rolle und automatische Branch-Erstellung oder Webvorschauen für Pull-Requests verwendet, müssen Sie sorgfältig verwalten, welche Branches auf die Rolle zugreifen können. Wir empfehlen, keine Rolle auf App-Ebene zu verwenden. Stattdessen sollten Sie eine Rechenrolle auf Filialebene anhängen. Auf diese Weise können Sie nur den Branchen Berechtigungen gewähren, die Zugriff auf bestimmte Ressourcen benötigen.

Wenn die Anmeldeinformationen Ihrer Rolle offengelegt werden, ergreifen Sie die folgenden Maßnahmen, um jeglichen Zugriff auf die Anmeldeinformationen der Rolle zu entfernen.

1. Widerrufen Sie alle Sitzungen

   Anweisungen zum sofortigen Widerrufen aller Berechtigungen für die Anmeldeinformationen der Rolle finden Sie unter Temporäre Sicherheitsanmeldeinformationen für die IAM-Rolle widerrufen.

2. Löschen Sie die Rolle aus der Amplify-Konsole

   Diese Aktion wird sofort wirksam. Sie müssen Ihre Anwendung nicht erneut bereitstellen.

Um eine Compute-Rolle in der Amplify-Konsole zu löschen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole unter https://console.aws.amazon.com/amplify/.
2. Wählen Sie auf der Seite Alle Apps den Namen der App aus, aus der Sie die Compute-Rolle entfernen möchten.
3. Wählen Sie im Navigationsbereich App-Einstellungen und dann IAM-Rollen aus.
4. Wählen Sie im Abschnitt Rechenrolle die Option Bearbeiten aus.
5. Um die Standardrolle zu löschen, wählen Sie das X rechts neben dem Namen der Rolle.
6. Wählen Sie Speichern.

Hinzufügen einer Servicerolle mit Berechtigungen für den Zugriff auf CloudWatch Protokolle

Amplify sendet Informationen über Ihre SSR-Laufzeit an Amazon CloudWatch Logs in Ihrem. AWS-Konto Wenn Sie eine SSR-App bereitstellen, benötigt die App eine IAM-Dienstrolle, die Amplify übernimmt, wenn es andere Dienste in Ihrem Namen aufruft. Sie können entweder Amplify Hosting Compute erlauben, automatisch eine Servicerolle für Sie zu erstellen, oder Sie können eine Rolle angeben, die Sie erstellt haben.

Wenn Sie Amplify erlauben, eine IAM-Rolle für Sie zu erstellen, verfügt die Rolle bereits über die Berechtigungen zum Erstellen CloudWatch von Protokollen. Wenn Sie Ihre eigene IAM-Rolle erstellen, müssen Sie Ihrer Richtlinie die folgenden Berechtigungen hinzufügen, damit Amplify auf Amazon CloudWatch Logs zugreifen kann.

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

Vereinheitlichte Webhooks für Git-Repositorys

Amplify Hosting verwendet Webhooks, um nach einem neuen Commit in Ihr Git-Repository automatisch einen Build zu initiieren. Die vereinheitlichte Webhooks-Funktion verbessert die Integrationen von Amplify mit Git-Anbietern und ermöglicht es Ihnen, mehr Amplify-Anwendungen mit einem einzigen Repository zu verbinden. Mit vereinheitlichten Webhooks verwendet Amplify jetzt einen einzigen Webhook pro Region für alle zugehörigen Anwendungen in Ihrem Repository. Wenn Ihr Repository beispielsweise mit Anwendungen in den Regionen USA Ost (Nord-Virginia) und USA West (Oregon) verbunden ist, haben Sie zwei einheitliche Webhooks.

Vor dieser Version hat Amplify für jede App, die einem Repository zugeordnet ist, einen neuen Webhook erstellt. Wenn Sie mehrere Apps in einem einzigen Repository hätten, könnten Sie die von einzelnen Git-Anbietern auferlegten Webhook-Grenzwerte erreichen und daran gehindert werden, weitere Apps hinzuzufügen. Dies war besonders schwierig für Teams, die in Monorepos arbeiteten, wo mehrere Projekte in einem einzigen Repository existieren.

Vereinheitlichte Webhooks bieten die folgenden Vorteile:

- Überwinden Sie die Webhook-Limits des Git-Anbieters: Sie können so viele Amplify-Apps, wie Sie benötigen, mit einem einzigen Repository verbinden.
- Verbesserte Monorepo-Unterstützung: Sie haben mehr Flexibilität und Effizienz bei der Arbeit mit Monorepos, bei denen sich mehrere Projekte ein einziges Repository teilen.
- Vereinfachtes Management: Die Verwaltung mehrerer Amplify-Apps mit einem einzigen Repository-Webhook reduziert die Komplexität und reduziert potenzielle Fehlerquellen.
- Verbesserte Workflow-Integration: Sie können die von Ihrem Git-Anbieter zugewiesenen Webhooks für andere wichtige Workflows in Ihrem Entwicklungsprozess verwenden.

Erste Schritte mit vereinheitlichten Webhooks

Eine neue App erstellen

Wenn Sie eine neue Anwendung aus einem Git-Repository auf Amplify Hosting bereitstellen, wird die einheitliche Webhooks-Funktion automatisch für Ihr Repository implementiert. Anweisungen zum Erstellen einer neuen Anwendung finden Sie unter. Erste Schritte mit der Bereitstellung einer App auf Amplify Hosting

Eine bestehende App aktualisieren

Für bestehende Amplify-Anwendungen müssen Sie Ihr Git-Repository erneut mit Ihrer Anwendung verbinden, um die vorhandenen Webhooks durch einen einheitlichen Webhook zu ersetzen. Wenn du die maximale Anzahl von Webhooks, die dein Git-Anbieter zulässt, bereits erreicht hast, ist die Migration zum vereinheitlichten Webhook möglicherweise nicht erfolgreich. Entfernen Sie in diesem Fall manuell mindestens einen vorhandenen Webhook, bevor Sie die Verbindung erneut herstellen.

Sie können mehrere Anwendungen in einem Repository haben, die in verschiedenen AWS Regionen bereitgestellt werden. Da Amplify-Operationen regionsbezogen sind, erfolgt die Migration zu einem einheitlichen Webhook nur für die Webhooks in der Region, in der Sie Ihre Amplify-App erneut verbunden haben. Infolgedessen werden in Ihrem Repository möglicherweise sowohl auf Anwendungs-ID basierende Webhooks als auch regionsbasierte vereinheitlichte Webhooks angezeigt.

Verwenden Sie die folgenden Anweisungen, um eine bestehende Amplify-App auf einen einheitlichen Webhook zu migrieren.

Um eine bestehende Amplify-App auf einen einheitlichen Webhook zu migrieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, die Sie zu einem einheitlichen Webhook migrieren möchten.
3. Wählen Sie im Navigationsbereich App-Einstellungen und dann Branch-Einstellungen aus.
4. Wählen Sie auf der Seite mit den Branch-Einstellungen die Option Repository erneut verbinden aus.
5. Um zu überprüfen, ob die Migration zum vereinheitlichten Webhook erfolgreich war, navigieren Sie zu den Webhook-Einstellungen in Ihrem Git-Repository. Sie sollten eine einzelne Webhook-URL im Format sehen. `https://amplify-webhooks.`*`Region`*`.amazonaws.com/`*`git-provider`*

Sicherheit in Amplify

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS Amplify, finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amplify anwenden können. Die folgenden Themen zeigen Ihnen, wie Sie Amplify konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Amplify-Ressourcen überwachen und sichern können.

Topics

Identity and Access Management für Amplify

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), Amplify-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Fehlerbehebung bei Amplify Identity and Access).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe So funktioniert Amplify mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Beispiele für identitätsbasierte Richtlinien für Amplify).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden

finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind)

sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Richtlinien zur Dienstkontrolle (SCPs) — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter Richtlinien zur Ressourcenkontrolle (RCPs).
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

So funktioniert Amplify mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amplify zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen mit Amplify verwendet werden können.

IAM-Funktionen, die Sie mit Amplify verwenden können

IAM-Feature	Unterstützung Amplify
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Forward Access Sessions (FAS)	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie Amplify und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für Amplify

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amplify

Beispiele für identitätsbasierte Richtlinien von Amplify finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amplify

Ressourcenbasierte Richtlinien innerhalb von Amplify

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Politische Maßnahmen für Amplify

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der Amplify-Aktionen finden Sie unter Actions defined by AWS Amplify in der Service Authorization Reference.

Richtlinienaktionen in Amplify verwenden das folgende Präfix vor der Aktion:

```
amplify
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Beispiele für identitätsbasierte Richtlinien von Amplify finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amplify

Politische Ressourcen für Amplify

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der Amplify-Ressourcentypen und ihrer ARNs Typen finden Sie unter Ressourcentypen definiert von AWS Amplify in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von AWS Amplify definierte Aktionen.

Beispiele für identitätsbasierte Richtlinien von Amplify finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amplify

Schlüssel zur Richtlinienbedingung für Amplify

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der Amplify-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für AWS Amplify in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Aktionen definiert von AWS Amplify.

Beispiele für identitätsbasierte Richtlinien von Amplify finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amplify

Zugriffskontrolllisten (ACLs) in Amplify

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit Amplify

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, um Operationen zu ermöglichen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Temporäre Anmeldeinformationen mit Amplify verwenden

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie den Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Weiterleitungszugriffssitzungen für Amplify

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für Amplify

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern

und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann die Amplify-Funktionalität beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amplify Sie dazu anleitet.

Serviceverknüpfte Rollen für Amplify

Unterstützt serviceverknüpfte Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen finden Sie im AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Ja, um die Dokumentation zu serviceverknüpften Rollen für diesen Service aufzurufen.

Beispiele für identitätsbasierte Richtlinien für Amplify

Standardmäßig haben Benutzer und Rollen keine Berechtigung, Amplify-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von Amplify definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Amplify in der Service Authorization Reference.

Themen

- Best Practices für Richtlinien

- Verwenden der Amplify-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amplify-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Wenn du identitätsbasierte Richtlinien erstellst oder bearbeitest, befolge diese Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als

100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.

- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amplify-Konsole

Um auf die AWS Amplify Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amplify-Ressourcen in Ihrem AWS-Konto aufzulisten und anzuzeigen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Mit der Veröffentlichung von Amplify Studio sind für das Löschen einer App oder eines Backends beides `amplify` und `amplifybackend` Berechtigungen erforderlich. Wenn eine IAM-Richtlinie nur `amplify` Berechtigungen vorsieht, erhält ein Benutzer beim Versuch, eine App zu löschen, einen Berechtigungsfehler. Wenn Sie ein Administrator sind, der Richtlinien schreibt, sollten Sie die richtigen Berechtigungen für Benutzer festlegen, die Löschaktionen ausführen müssen.

Um sicherzustellen, dass Benutzer und Rollen die Amplify-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die Amplify `ConsoleAccess` - oder `ReadOnly` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer

Benutzeridentität angefügt sind. Diese Richtlinie beinhaltet Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der OR-API. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete Richtlinien für AWS Amplify

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird. AWS AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle

bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

Von AWS verwaltete Richtlinie: AdministratorAccess -Amplify

Sie können die `AdministratorAccess-Amplify`-Richtlinie an Ihre IAM-Identitäten anfügen. Amplify verknüpft diese Richtlinie auch mit einer Servicerolle, die es Amplify ermöglicht, Aktionen in Ihrem Namen durchzuführen.

Wenn Sie ein Backend in der Amplify-Konsole bereitstellen, müssen Sie eine `Amplify-Backend Deployment` Servicerolle erstellen, die Amplify zum Erstellen und Verwalten von Ressourcen verwendet. AWS IAM fügt die `AdministratorAccess-Amplify` verwaltete Richtlinie der Servicerolle hinzu. `Amplify-Backend Deployment`

Diese Richtlinie gewährt Kontoverwaltungsberechtigungen und ermöglicht gleichzeitig ausdrücklich den direkten Zugriff auf Ressourcen, die Amplify-Anwendungen zum Erstellen und Verwalten von Backends benötigen.

Details zu Berechtigungen

Diese Richtlinie ermöglicht den Zugriff auf mehrere AWS Dienste, einschließlich IAM-Aktionen. Diese Aktionen ermöglichen es Identitäten, für die diese Richtlinie gilt, zu verwenden, AWS Identity and Access Management um andere Identitäten mit beliebigen Berechtigungen zu erstellen. Dies ermöglicht eine Eskalation von Berechtigungen, und diese Richtlinie sollte als ebenso wirksam angesehen werden wie die Richtlinie. `AdministratorAccess`

Diese Richtlinie erteilt der `iam:PassRole` Aktion die Genehmigung für alle Ressourcen. Dies ist erforderlich, um die Konfiguration von Amazon Cognito Cognito-Benutzerpools zu unterstützen.

Die Berechtigungen für diese Richtlinie finden Sie unter AdministratorAccess-Amplify in der Referenz für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmplifyBackendDeployFullAccess

Sie können die `AmplifyBackendDeployFullAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Amplify volle Zugriffsberechtigungen für die Bereitstellung von Amplify-Backend-Ressourcen mithilfe von. AWS Cloud Development Kit (AWS CDK) Berechtigungen werden auf die AWS CDK Rollen übertragen, die über die erforderlichen Richtlinienberechtigungen verfügen. `AdministratorAccess`

Details zu Berechtigungen

Diese Richtlinie umfasst Berechtigungen für die folgenden Aktionen.

- `Amplify`— Rufen Sie Metadaten über bereitgestellte Anwendungen ab.
- `CloudFormation`— Erstellen, aktualisieren und löschen Sie von Amplify verwaltete Stacks.
- `SSM`— Erstellen, aktualisieren und löschen Sie den von Amplify verwalteten SSM-Parameterspeicher `String` und `SecureString` die Parameter.
- `AWS AppSync`— AWS AppSync Schema-, Resolver- und Funktionsressourcen aktualisieren und abrufen. Der Zweck besteht darin, die Sandbox-Hotswapping-Funktionalität der zweiten Generation zu unterstützen.
- `Lambda`— Aktualisieren Sie die Konfiguration für von Amplify verwaltete Funktionen und rufen Sie sie ab. Der Zweck besteht darin, die Sandbox-Hotswapping-Funktionalität der zweiten Generation zu unterstützen.

  Ruft die Tags einer Lambda-Funktion ab. Der Zweck besteht darin, von Kunden definierte Lambda-Funktionen zu unterstützen.
- `Amazon S3`— Rufen Sie die Implementierungsressourcen von Amplify ab.
- `AWS -Security-Token-Service`— Ermöglicht der AWS Cloud Development Kit (AWS CDK) CLI, die Bereitstellungsrolle zu übernehmen.
- `Amazon RDS`— Liest Metadaten von DB-Instances, Clustern und Proxys.
- `Amazon EC2`— Lesen Sie die Informationen zur Verfügbarkeitszone für ein Subnetz.
- `CloudWatch Logs`— Rufen Sie die Protokolle für die Lambda-Funktion eines Kunden ab. Der Zweck besteht darin, einer Amplify Cloud-Entwicklungssandbox-Umgebung zu ermöglichen, die Protokolle einer Lambda-Funktion an das Terminal eines Kunden zu streamen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AmplifyBackendDeployFullAccess in der Referenz zu von AWS verwalteten Richtlinien.

Amplify Sie die Aktualisierungen AWS verwalteter Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amplify an, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen. Um automatische Warnungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Dokumenthistorie für AWS Amplify-Seite.

Änderungen	Beschreibung	Datum
AmplifyBackendDeployFullAccess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie der `logs:FilterLogEvents` Ressource Lesezugriff hinzu, damit Amplify Protokolle von Funktionen streamen kann, für die eine benutzerdefinierte Protokollgruppe erstellt wurde. Dies ist eine Erweiterung der bestehenden Fähigkeit, die Protokolle einer Lambda-Funktion zu streamen.	14. November 2024
AmplifyBackendDeployFullAccess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie Lesezugriff auf die `lambda:ListTags` und `logs:FilterLogEvents` -Ressourcen hinzu, um von Kunden definierte Lambda-Funktionen zu unterstützen. Diese Berechtigungen ermöglichen es einer Amplify Cloud-Entwicklungssandbox-Umgebung, die Protokolle einer Lambda-Funktion an	18. Juli 2024

Änderungen	Beschreibung	Datum
	das Terminal eines Kunden zu streamen.	
AmplifyBackendDeployFullAccess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie der `arn:aws:ssm:*:*:parameter/cdk-bootstrap/*` Ressource Lesezugriff hinzu, damit Amplify die CDK-Bootstrap-Version im Konto eines Kunden erkennen kann.	31. Mai 2024

Änderungen	Beschreibung	Datum
AmplifyBackendDeployFullAccess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie eine neue `AmplifyDiscoverRDSVpcConfig` Richtlinienerklärung mit Leseberechtigungen für Amazon RDS und Amazon EC2 hinzu, die sowohl von den Ressourcen- als auch von den Kontobedingungen abhängig sind. Diese Berechtigungen unterstützen den `npx amplify generate schema-from-database` Befehl Amplify Gen 2, mit dem Kunden ein Typescript-Datenschema aus einer vorhandenen SQL-Datenbank generieren können. Fügen Sie die Berechtigungen`rds:DescribeDBProxies` ,`rds:DescribeDBInstances` , `rds:DescribeDBClusters` `rds:DescribeDBSubnetGroups` , und hinzu. `ec2:DescribeSubnets` Der `npx amplify generate schema-from-database` Befehl benötigt diese Berechtigungen, um zu überprüfen, ob ein bestimmter DB-Host in Amazon RDS gehostet wird, und um	17. April 2024

Änderungen	Beschreibung	Datum
	automatisch die Amazon VPC-Konfiguration zu generiere n, die für die Bereitstellung der anderen Ressourcen erforderlich ist, die für die Einrichtung einer von einer SQL-Datenbank gestützten AWS AppSync API erforderl ich sind.	
AmplifyBackendDeployFullAcc ess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie die `cloudform ation:DeleteStack` Richtlinienaktion hinzu, um das Löschen von Stacks zu unterstützen, wenn die `DeleteBranch` API aufgerufen wird. Fügen Sie die `lambda:Ge tFunction` Richtlini enaktion hinzu, um Hotswappi ng-Funktionen zu unterstüt zen. Fügen Sie die `lambda:Up dateFunctionConfig uration` Richtlinienaktion hinzu, um Updates für die Lambda-Funktion zu unterstüt zen.	5. April 2024

Änderungen	Beschreibung	Datum
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie die `cloudformation:UnTagResource` Berechtigungen `cloudformation:TagResource` und hinzu, an die Anrufe unterstützt zt werden sollen. CloudFormation APIs	4. April 2024
AmplifyBackendDeployFullAccess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie die `lambda:InvokeFunction` Richtlinienaktion zur Unterstützung von AWS Cloud Development Kit (AWS CDK) Hotswapping hinzu. Der AWS CDK ruft direkt eine Lambda-Funktion auf, um Amazon S3-Asset-Hotswapping durchzuführen. Fügen Sie die `lambda:UpdateFunctionCode` Richtlinienaktion zur Unterstützung von Hotswapping-Funktionen hinzu.	02. Januar 2024
AmplifyBackendDeployFullAccess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie Richtlinienaktionen hinzu, um den `UpdateApiKey` Vorgang zu unterstützen. Dies ist erforderlich, um nach dem Beenden und Neustarten der Sandbox eine erfolgreiche App-Bereitstellung zu ermöglichen, ohne Ressourcen zu löschen.	17. November 2023

Änderungen	Beschreibung	Datum
AmplifyBackendDeployFullAccess – Aktualisierung auf eine bestehende Richtlinie	Fügen Sie die `amplify:GetBackendEnvironment` Erlaubnis hinzu, die Bereitstellung von Amplify-Apps zu unterstützen.	6. November 2023
AmplifyBackendDeployFullAccess – Neue Richtlinie	Amplify hat eine neue Richtlinie mit den Mindestberechtigungen hinzugefügt, die für die Bereitstellung von Amplify-Backend-Ressourcen erforderlich sind.	8. Oktober 2023
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie die `ecr:DescribeRepositories` Berechtigung hinzu, die für die Amplify Command Line Interface (CLI) erforderlich ist.	01. Juni 2023

Änderungen	Beschreibung	Datum
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie eine Richtlini enaktion hinzu, um das Entfernen von Tags aus einer AWS AppSync Ressource zu unterstützen. Fügen Sie eine Richtlini enaktion zur Unterstützung der Amazon Polly Polly-Ressource hinzu. Fügen Sie eine Richtlini enaktion hinzu, um die Aktualisierung der OpenSearc h Domain-Konfiguration zu unterstützen. Fügen Sie eine Richtlini enaktion hinzu, um das Entfernen von Tags aus einer AWS Identity and Access Management Rolle zu unterstützen. Fügen Sie eine Richtlini enaktion hinzu, um das Entfernen von Tags aus einer Amazon DynamoDB DynamoDB-Ressource zu unterstützen. Fügen Sie dem `CLISDKCal ls` Anweisungsblock die `cloudfront:GetClou dFrontOriginAccess IdentityConfig`	24. Februar 2023

Änderungen	Beschreibung	Datum
	Berechtigungen `cloudfront:GetCloudFrontOriginAccessIdentity` und hinzu, um die Veröffentlichungs- und Hosting-Workflows von Amplify zu unterstützen. Fügen Sie dem `s3:PutBucketPublicAccessBlock CLIManageviaCFNPolicy` Anweisungsblock die Erlaubnis hinzu, damit der die bewährte Amazon S3 S3-Sicherheitsmethode AWS CLI zur Aktivierung der Amazon S3 S3-Funktion Block Public Access für interne Buckets unterstützen kann. Fügen Sie dem `cloudformation:DescribeStacks CLISDKCalls` Anweisungsblock die Berechtigung hinzu, um das Abrufen von CloudFormation Kundenstapeln bei Wiederholungsversuchen im Amplify-Backend-Prozessor zu unterstützen, um doppelte Ausführungen zu vermeiden, wenn ein Stack aktualisiert wird. Fügen Sie die Erlaubnis zum Anweisungsblock hinzu. `cloudformation:Lis`	

Änderungen	Beschreibung	Datum
	`tStacks CLICloudformationPolicy` Diese Erlaubnis ist erforderlich, um die CloudFormation DescribeStacks Aktion vollständig zu unterstützen.	
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie Richtlinienaktionen hinzu, damit die serversei tige Rendering-Funktion von Amplify Anwendungsmetriken CloudWatch in die eines Kunden übertragen kann. AWS-Konto	30. August 2022
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie Richtlinienaktionen hinzu, um den öffentlichen Zugriff auf den Amazon S3 S3-Bucket für die Amplify-Bereitstellung zu blockieren.	27. April 2022

Änderungen	Beschreibung	Datum
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie eine Aktion hinzu, mit der Kunden ihre serversei tig gerenderten Apps (SSR) löschen können. Dadurch kann auch die entsprech ende CloudFront Distribution erfolgreich gelöscht werden. Fügen Sie eine Aktion hinzu, damit Kunden mithilfe der Amplify-CLI eine andere Lambda-Funktion angeben können, um Ereignisse aus einer vorhandenen Ereignisq uelle zu verarbeiten. Mit diesen Änderungen kann AWS Lambda die Aktion ausgeführt werden. UpdateEventSourceM apping	17. April 2022
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie eine Richtlini enaktion hinzu, um Amplify UI Builder-Aktionen für alle Ressourcen zu aktivieren.	2. Dezember 2021

Änderungen	Beschreibung	Datum
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie Richtlinienaktionen hinzu, um die Amazon Cognito Cognito-Authentifizierungsfunktion zu unterstützen, die Anbieter sozialer Identitäten verwendet. Fügen Sie eine Richtlinienaktion zur Unterstützung von Lambda-Schichten hinzu. Fügen Sie eine Richtlinienaktion zur Unterstützung der Amplify Storage-Kategorie hinzu.	8. November 2021

Änderungen	Beschreibung	Datum
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Fügen Sie Amazon Lex Lex-Aktionen hinzu, um die Kategorie Amplify Interactions zu unterstützen. Fügen Sie Amazon Rekogniti on Rekognition-Aktionen hinzu, um die Kategorie Amplify Predictions zu unterstützen. Fügen Sie eine Amazon Cognito Cognito-Aktion hinzu, um die MFA-Konfiguration in Amazon Cognito Cognito-B enutzerpools zu unterstützen. Fügen Sie CloudFormation Aktionen zur Unterstützung hinzu. CloudFormation StackSets Fügen Sie Amazon Location Service Service-Aktionen hinzu, um die Amplify Geo-Kategorie zu unterstützen. Fügen Sie eine Lambda-Ak tion hinzu, um Lambda-Sc hichten in Amplify zu unterstüt zen. Fügen Sie CloudWatch Log-Aktionen zur Unterstüt zung von Ereignissen hinzu. CloudWatch	27. September 2021

Änderungen	Beschreibung	Datum
	Fügen Sie Amazon S3 S3-Aktionen hinzu, um die Kategorie Amplify Storage zu unterstützen. Fügen Sie Richtlinienaktionen hinzu, um serverseitig gerenderte Apps (SSR) zu unterstützen.	

Änderungen	Beschreibung	Datum
AdministratorAccess-Amplify — Aktualisierung einer bestehenden Richtlinie	Konsolidieren Sie alle Amplify-Aktionen in einer einzigen `amplify:*` Aktion. Fügen Sie eine Amazon S3 S3-Aktion hinzu, um die Verschlüsselung von Amazon S3 S3-Buckets von Kunden zu unterstützen. Fügen Sie IAM-Berec htigungsgrenzaktionen hinzu, um Amplify-Apps zu unterstützen, für die Berechtig ungsgrenzen aktiviert sind. Fügen Sie Amazon SNS SNS-Aktionen hinzu, um das Anzeigen von Ausgangst elefonnummern und das Anzeigen, Erstellen, Überprüfe n und Löschen von Zieltelef onnummern zu unterstützen. Amplify Studio: Fügen Sie Amazon Cognito, AWS Lambda IAM und CloudForm ation Richtlinienaktionen hinzu, um die Verwaltung von Backends in der Amplify-K onsole und Amplify Studio zu ermöglichen. Fügen Sie eine AWS Systems Manager (SSM) -Richtlin ienerklärung hinzu, um	28. Juli 2021

Änderungen	Beschreibung	Datum
	Amplify-Umgebungsg eheimnisse zu verwalten. Fügen Sie eine CloudForm ation `ListResources` Aktion hinzu, um Lambda-Ebenen für Amplify-Apps zu unterstützen.	
Amplify hat begonnen, Änderungen zu verfolgen	Amplify begann, Änderunge n für seine AWS verwalteten Richtlinien nachzuverfolgen.	28. Juli 2021

Fehlerbehebung bei Amplify Identity and Access

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amplify und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion in Amplify durchzuführen
- Ich bin nicht berechtigt, IAM auszuführen: PassRole
- Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amplify-Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion in Amplify durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `amplify:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplify:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `amplify:GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Mit der Veröffentlichung von Amplify Studio sind für das Löschen einer App oder eines Backends beides `amplify` und `amplifybackend` Berechtigungen erforderlich. Wenn ein Administrator eine IAM-Richtlinie geschrieben hat, die nur `amplify` Berechtigungen bereitstellt, wird beim Versuch, eine App zu löschen, ein Berechtigungsfehler angezeigt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM-Benutzer versucht, mit der Konsole eine fiktive `example-amplify-app` Ressource zu löschen, aber nicht über die entsprechenden Berechtigungen verfügt. `amplifybackend:RemoveAllBackends`

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplifybackend;:RemoveAllBackends on resource: example-amplify-app
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `example-amplify-app` auf die Ressource `amplifybackend:RemoveAllBackends` zugreifen zu können.

Ich bin nicht berechtigt, IAM auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amplify übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amplify auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amplify-Ressourcen ermöglichen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amplify diese Funktionen unterstützt, finden Sie unterSo funktioniert Amplify mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Datenschutz in Amplify

AWS Amplify entspricht dem Modell der AWS geteilten Verantwortung mit der , das Vorschriften und Richtlinien für den Datenschutz beinhaltet. AWS ist verantwortlich für den Schutz der globalen Infrastruktur, die alle AWS Dienste betreibt. AWS behält die Kontrolle über die auf dieser Infrastruktur gehosteten Daten, einschließlich der Sicherheitskonfigurationen für den Umgang mit

Kundeninhalten und personenbezogenen Daten. AWS Kunden und APN-Partner, die entweder als Datenverantwortliche oder als Datenverarbeiter agieren, sind für alle personenbezogenen Daten verantwortlich, die sie in die AWS Cloud stellen.

Aus Datenschutzgründen empfehlen wir, dass Sie Ihre AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen innerhalb der AWS Dienste.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu sichern.

Wir empfehlen dringend, in Freitextfeldern wie z. B. im Feld Name keine sensiblen, identifizierenden Informationen wie Kontonummern von Kunden einzugeben. Dies gilt auch, wenn Sie mit Amplify oder anderen AWS Diensten über die Konsole AWS CLI, API oder AWS SDKs arbeiten. Alle Daten, die Sie in Amplify oder andere Dienste eingeben, werden möglicherweise zur Aufnahme in Diagnoseprotokolle aufgenommen. Wenn Sie eine URL für einen externen Server bereitstellen, schließen Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL ein.

Weitere Informationen zum Datenschutz enthält der Blog-Beitrag AWS Shared Responsibility Model and GDPR im AWS -Sicherheitsblog.

Verschlüsselung im Ruhezustand

„Verschlüsselung im Ruhezustand“ bezieht sich auf den Schutz Ihrer Daten vor unbefugtem Zugriff durch deren Verschlüsselung während der Speicherung. Amplify verschlüsselt standardmäßig die Build-Artefakte einer App mithilfe AWS KMS keys von Amazon S3, die von der verwaltet werden. AWS Key Management Service

Amplify verwendet Amazon CloudFront , um Ihre App für Ihre Kunden bereitzustellen. CloudFront verwendet SSDs , die für Edge-Standorte (POPs) verschlüsselt sind, und verschlüsselte EBS-

Volumes für regionale Edge-Caches (). RECs Der Funktionscode und die Konfiguration in CloudFront Functions werden immer in einem verschlüsselten Format auf dem am Edge verschlüsselten Standort und SSDs an anderen Speicherorten gespeichert POPs, die von verwendet werden. CloudFront

Verschlüsselung während der Übertragung

Der Begriff „Verschlüsselung während der Übertragung“ bedeutet, dass Ihre Daten davor geschützt werden, abgefangen zu werden, während sie zwischen Kommunikationsendpunkten verschoben werden. Amplify Hosting bietet standardmäßig Verschlüsselung für Daten während der Übertragung. Die gesamte Kommunikation zwischen Kunden und Amplify sowie zwischen Amplify und seinen nachgelagerten Abhängigkeiten ist durch TLS-Verbindungen geschützt, die mit dem Signature Version 4-Signaturprozess signiert wurden. Alle Amplify Hosting-Endpunkte verwenden SHA-256-Zertifikate, die von verwaltet werden. AWS Private Certificate AuthorityWeitere Informationen finden Sie unter Signaturprozess für Signature Version 4 und Was ist. AWS Private Certificate Authority

Verwaltung von Verschlüsselungsschlüsseln

AWS Key Management Service (KMS) ist ein verwalteter Dienst zur Erstellung und Steuerung der Verschlüsselungsschlüssel AWS KMS keys, die zur Verschlüsselung von Kundendaten verwendet werden. AWS Amplify generiert und verwaltet kryptografische Schlüssel zur Verschlüsselung von Daten im Auftrag von Kunden. Es gibt keine Verschlüsselungsschlüssel, die Sie verwalten müssen.

Konformitätsprüfung für AWS Amplify

Externe Prüfer bewerten die Sicherheit und Einhaltung von Vorschriften im AWS Amplify Rahmen mehrerer AWS Compliance-Programme. Dazu gehören SOC, PCI, ISO, HIPAA, MTCS, C5, K-ISMS, ENS High, OSPAR, HITRUST CSF und FINMA.

Informationen darüber, ob ein in den Geltungsbereich bestimmter Compliance-Programme AWS-Service fällt, finden Sie unter Umfang nach Compliance-Programm unter AWS-Services Umfang nach Compliance-Programm. Wählen Sie dort das Compliance-ProgrammAWS-Services aus, an sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen

und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der AWS Sicherheitsdokumentation.

Sicherheit der Infrastruktur in AWS Amplify

Als verwalteter Dienst AWS Amplify ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amplify zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Protokollierung und Überwachung von Sicherheitsereignissen in Amplify

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amplify und Ihren anderen AWS Lösungen. AWS bietet die folgenden Überwachungstools, um Amplify zu beobachten, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht in Echtzeit Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen AWS. Sie können Kennzahlen sammeln und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme einrichten, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer Amazon Elastic Compute Cloud (Amazon EC2) -Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen zur Verwendung von CloudWatch Metriken und Alarmen mit Amplify finden Sie unterÜberwachung einer Amplify-Anwendung.
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien von Amazon EC2 EC2-Instances und anderen Quellen überwachen AWS CloudTrail, speichern und darauf zugreifen. CloudWatch

Logs können Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

- AWS CloudTrailerfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt wurden, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon Simple Storage Service (Amazon S3) -Bucket. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Aufrufe getätigt wurden und wann die Aufrufe erfolgten. Weitere Informationen finden Sie unter Protokollieren von Amplify-API-Aufrufen mit AWS CloudTrail.
- Amazon EventBridge ist ein serverloser Event-Bus-Service, der es einfach macht, Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen zu verbinden. EventBridge stellt einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, Software-as-a-Service (SaaS) -Anwendungen und AWS Diensten bereit und leitet diese Daten an Ziele weiter, wie AWS Lambda z. Auf diese Weise können Sie Ereignisse überwachen, die in Diensten auftreten, und ereignisgesteuerte Architekturen erstellen. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In AWS kann ein dienstübergreifendes Identitätswechsels zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn`und die `aws:SourceAccount`globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die der AWS Amplify Ressource einen anderen Dienst gewähren. Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Der Wert von `aws:SourceArn` muss der Branch-ARN der Amplify-App sein. Geben Sie diesen Wert im Format `arn:`*`Partition`*`:amplify:`*`Region`*`:`*`Account`*`:apps/`*`AppId`*`/branches/`*`BranchName`* an.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. Beispiel, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Das folgende Beispiel zeigt eine Rollenvertrauensrichtlinie, die Sie anwenden können, um den Zugriff auf jede Amplify-App in Ihrem Konto einzuschränken und das Problem mit dem verwirrten Stellvertreter zu verhindern. Um diese Richtlinie zu verwenden, ersetzen Sie den rot kursiv geschriebenen Text in der Beispielrichtlinie durch Ihre eigenen Informationen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": [
          "amplify.me-south-1.amazonaws.com",
          "amplify.eu-south-1.amazonaws.com",
          "amplify.ap-east-1.amazonaws.com",
          "amplifybackend.amazonaws.com",
          "amplify.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
```

```
  }
}
```

Das folgende Beispiel zeigt eine Rollenvertrauensrichtlinie, die Sie anwenden können, um den Zugriff auf eine bestimmte Amplify-App in Ihrem Konto einzuschränken und das Problem mit dem verwirrten Stellvertreter zu verhindern. Um diese Richtlinie zu verwenden, ersetzen Sie den rot kursiv gedruckten Text in der Beispielrichtlinie durch Ihre eigenen Informationen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": [
          "amplify.me-south-1.amazonaws.com",
          "amplify.eu-south-1.amazonaws.com",
          "amplify.ap-east-1.amazonaws.com",
          "amplifybackend.amazonaws.com",
          "amplify.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/d123456789/
branches/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Bewährte Sicherheitsmethoden für Amplify

Amplify bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden stellen allgemeine Richtlinien und keine vollständige Sicherheitslösung dar. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Empfehlungen und nicht als bindend ansehen.

Verwendung von Cookies mit der Amplify-Standarddomain

Wenn Sie Amplify verwenden, um eine Web-App bereitzustellen, hostet Amplify sie für Sie auf der `amplifyapp.com` Standarddomain. Sie können Ihre App auf einer URL anzeigen, die als formatiert ist. `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`

Um die Sicherheit Ihrer Amplify-Anwendungen zu erhöhen, ist die Domain amplifyapp.com in der Public Suffix List (PSL) registriert. Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host-` Präfix zu verwenden, falls Sie jemals sensible Cookies im Standard-Domainnamen für Ihre Amplify-Anwendungen einrichten müssen. Diese Vorgehensweise trägt dazu bei, Ihre Domain vor CSRF-Versuchen (Cross-Site Request Forgery Attempts, Anforderungsfälschung zwischen Websites) zu schützen. Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

Amplify Kontingente für Hosting-Dienste

Im Folgenden sind die Servicekontingente für das AWS Amplify Hosting aufgeführt. Servicekontingenten (früher als Limits bezeichnet) sind die maximale Anzahl von Serviceressourcen oder Vorgängen für Sie AWS-Konto.

Neu AWS-Konten sind die Kontingente für Apps und gleichzeitige Jobs reduziert. AWS erhöht diese Kontingente automatisch auf der Grundlage Ihrer Nutzung. Sie können auch eine Kontingenterhöhung beantragen.

Die Service-Quotas-Konsole enthält Informationen zu Kontingenten für Ihr Konto. Sie können die Service Quotas-Konsole verwenden, um Standard-Kontingente anzuzeigen und Kontingent-Erhöhungen für einstellbare Kontingente anzufordern. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service Quotas-Benutzerhandbuch.

Name	Standard	Anpas	Description
Apps	Jede unterstützte Region: 25	Ja	Die maximale Anzahl von Apps, die Sie in AWS Amplify Console in diesem Konto in der aktuellen Region erstellen können.
Branches pro App	Jede unterstützte Region: 50	Nein	Die maximale Anzahl von Branches pro App, die Sie in diesem Konto in der aktuellen Region erstellen können.
Größe des Build-Artefakts	Jede unterstützte Region: 5 Gigabyte	Nein	Die maximale Größe (in GB) eines App-Build-Artefakts. Ein Build-Artefakt wird nach einem Build von AWS Amplify Console bereitgestellt.

Name	Standard	Anpas	Description
Cache-Artifaktgröße	Jede unterstützte Region: 5 Gigabyte	Nein	Die maximale Größe (in GB) eines Cache-Artefakts.
Gleichzeitige Aufträge	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von gleichzeitigen Aufträgen, die Sie in diesem Konto in der aktuellen Region erstellen können.
Domains pro App	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Domains pro App, die Sie in diesem Konto in der aktuellen Region erstellen können.
Umgebungscache-Artifaktgröße	Jede unterstützte Region: 5 Gigabyte	Nein	Die maximale Größe (in GB) des Umgebungs-Cache-Artefakts.
Größe der manuell bereitgestellten ZIP-Datei	Jede unterstützte Region: 5 Gigabyte	Nein	Die maximale Größe (in GB) einer manuell bereitgestellten ZIP-Datei.
Maximale Anzahl von App-Erstellungen pro Stunde	Jede unterstützte Region: 25	Nein	Die maximale Anzahl von Apps, die Sie in AWS Amplify Console pro Stunde in diesem Konto in der aktuellen Region erstellen können.

Name	Standard	Anpas	Description
Tokens pro Sekunde anfordern	Jede unterstützte Region: 20 000	Ja	Die maximale Anzahl von Anforderungstoken pro Sekunde für eine App. Amplify Hosting weist Anfragen Token zu, basierend auf der Menge der Ressourcen (Verarbeitungszeit und Datenübertragung), die sie verbrauchen.
Subdomains pro Domain	Jede unterstützte Region: 50	Nein	Die maximale Anzahl von Subdomains pro Domain, die Sie in diesem Konto in der aktuellen Region erstellen können.
Webhooks pro App	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Webhooks pro App, die Sie in diesem Konto in der aktuellen Region erstellen können.

Weitere Informationen zu Amplify-Servicekontingenten finden Sie unter AWS Amplify Endpunkte und Kontingente in der. Allgemeine AWS-Referenz

Fehlerbehebung bei Amplify Hosting

Wenn Sie bei der Arbeit mit Amplify Hosting auf Fehler oder Bereitstellungsprobleme stoßen, lesen Sie die Themen in diesem Abschnitt.

Themen

Behebung allgemeiner Amplify-Probleme

Die folgenden Informationen können Ihnen bei der Behebung allgemeiner Probleme mit Amplify Hosting helfen.

Themen

HTTP 429-Statuscode (Zu viele Anfragen)

Amplify steuert die Anzahl der Anfragen pro Sekunde (RPS) an Ihre Website auf der Grundlage der Verarbeitungszeit und der Datenübertragung, die eingehende Anfragen verbrauchen. Wenn Ihre

Anwendung einen HTTP-Statuscode 429 zurückgibt, überschreiten eingehende Anfragen die für Ihre Anwendung vorgesehene Bearbeitungszeit und Datenübertragung. Dieses Anwendungslimit wird durch die Servicequote von Amplify verwaltet. `REQUEST_TOKENS_PER_SECOND` Weitere Informationen zu Kontingenten finden Sie unter Amplify Kontingente für Hosting-Dienste.

Um dieses Problem zu beheben, empfehlen wir, Ihre Anwendung zu optimieren, um die Anforderungsdauer und die Datenübertragung zu reduzieren und so den RPS der App zu erhöhen. Bei denselben 20.000 Tokens kann beispielsweise eine hochoptimierte SSR-Seite, die innerhalb von 100 Millisekunden reagiert, höhere RPS unterstützen als eine Seite mit einer Latenz von mehr als 200 Millisekunden.

In ähnlicher Weise verbraucht eine Anwendung, die eine Antwortgröße von 1 MB zurückgibt, mehr Token als eine Anwendung, die eine Antwortgröße von 250 KB zurückgibt.

Wir empfehlen Ihnen außerdem, den CloudFront Amazon-Cache zu nutzen, indem Sie Cache-Control Header so konfigurieren, dass die Zeit, in der eine bestimmte Antwort im Cache aufbewahrt wird, maximiert wird. Anfragen, die vom CloudFront Cache aus bedient werden, werden nicht auf das Ratenlimit angerechnet. Jede CloudFront Distribution kann bis zu 250.000 Anfragen pro Sekunde verarbeiten, sodass Sie Ihre App mithilfe des Caches sehr stark skalieren können. Weitere Informationen zum CloudFront Cache finden Sie unter Optimizing Caching and Availability im Amazon CloudFront Developer Guide.

In der Amplify-Konsole werden der Build-Status und die Uhrzeit der letzten Aktualisierung für meine App nicht angezeigt

Wenn Sie in der Amplify-Konsole zur Seite Alle Apps navigieren, wird für jede Ihrer Apps in der aktuellen Region eine Kachel angezeigt. Wenn der Build-Status, wie z. B. Bereitgestellt, und die Uhrzeit der letzten Aktualisierung für eine App nicht angezeigt werden, ist der App kein `Production` Phase-Branch zugeordnet.

Um die Apps in der Konsole aufzulisten, verwendet Amplify die `ListApps` API. Amplify verwendet das `ProductionBranch.status` Attribut, um den Build-Status anzuzeigen, und das `ProductionBranch.lastDeployTime` Attribut, um den Zeitpunkt der letzten Aktualisierung anzuzeigen. Weitere Informationen zu dieser API finden Sie ProductionBranchin der Amplify Hosting API-Dokumentation.

Verwenden Sie die folgenden Anweisungen, um dem Branch Ihrer App eine `Production` Phase zuzuordnen.

1. Melden Sie sich bei der Amplify-Konsole an.
2. Wählen Sie auf der Seite Alle Apps die App aus, die Sie aktualisieren möchten.
3. Wählen Sie im Navigationsbereich App-Einstellungen und dann Branch-Einstellungen aus.
4. Wählen Sie im Abschnitt Filialeinstellungen die Option Bearbeiten aus.
5. Wählen Sie unter Produktionszweig den Namen der Filiale aus, die Sie verwenden möchten.
6. Wählen Sie Speichern.
7. Kehren Sie zur Seite „Alle Apps“ zurück. Der Build-Status und die Uhrzeit der letzten Aktualisierung sollten jetzt für Ihre App angezeigt werden.

Für neue Pull-Requests werden keine Webvorschauen erstellt

Mit der Webvorschau-Funktion können Sie eine Vorschau von Änderungen aus Pull-Requests anzeigen, bevor Sie sie zu einem Integrationszweig zusammenführen. Eine Webvorschau stellt jede an dein Repository gerichtete Pull-Anfrage an eine eindeutige Vorschau-URL bereit, die sich von der URL unterscheidet, die deine Hauptseite verwendet.

Wenn Sie Webvorschauen für Ihre App aktiviert haben, diese aber nicht für neue Apps erstellt werden PRs, untersuchen Sie, ob eine der folgenden Ursachen die Ursache für Ihr Problem ist.

1. Prüfen Sie, ob Ihre App das maximale `Branches per app` Dienstkontingent erreicht hat. Weitere Informationen zu Kontingenten finden Sie unter Amplify Kontingente für Hosting-Dienste.

   Um innerhalb des Standardkontingents von 50 Branches pro App zu bleiben, sollten Sie das auto Löschen von Filialen in Ihrer App aktivieren. Dadurch wird verhindert, dass du in deinem Konto Branches anhäufst, die in deinem Repository nicht mehr existieren.
2. Wenn Sie ein öffentliches GitHub Repository verwenden und Ihrer Amplify-App eine IAM-Servicerolle zugewiesen ist, erstellt Amplify aus Sicherheitsgründen keine Vorschauen. Beispielsweise benötigen Apps mit Backends und Apps, die auf der `WEB_COMPUTE` Hosting-Plattform bereitgestellt werden, eine IAM-Servicerolle. Daher können Sie für diese Arten von Apps keine Webvorschauen aktivieren, wenn ihr Repository öffentlich ist.

   Damit Webvorschauen für Ihre App funktionieren, können Sie entweder die Zuordnung der Dienstrolle aufheben (falls die App kein Backend hat oder keine `WEB_COMPUTE` App ist), oder Sie können das Repository privat machen. GitHub

Meine manuelle Bereitstellung bleibt in der Amplify-Konsole mit dem Status „Ausstehend“ hängen

Manuelle Bereitstellungen ermöglichen es Ihnen, Ihre Web-App mit Amplify Hosting zu veröffentlichen, ohne einen Git-Anbieter zu verbinden. Sie können eine der folgenden vier Bereitstellungsoptionen verwenden.

1. Ziehen Sie Ihren Anwendungsordner per Drag & Drop in die Amplify-Konsole.
2. Ziehen Sie eine ZIP-Datei (die die Build-Artefakte Ihrer Site enthält) per Drag & Drop in die Amplify-Konsole.
3. Laden Sie eine .zip-Datei (die die Build-Artefakte Ihrer Site enthält) in einen Amazon S3 S3-Bucket hoch und verbinden Sie den Bucket mit einer App in der Amplify-Konsole.
4. Verwenden Sie eine öffentliche URL, die auf eine ZIP-Datei (die die Build-Artefakte Ihrer Site enthält) in der Amplify-Konsole verweist.

Wir sind uns der Probleme mit der Drag-a-Drop-Funktionalität bewusst, wenn ein Anwendungsordner für eine manuelle Bereitstellung in der Amplify-Konsole verwendet wird. Diese Bereitstellungen können aus den folgenden Gründen fehlschlagen.

- Vorübergehende Netzwerkprobleme treten auf.
- Während des Uploads gibt es eine lokale Änderung der Dateien.
- Die Browsersitzung versucht, eine große Menge statischer Assets gleichzeitig hochzuladen.

Wir arbeiten zwar daran, die Zuverlässigkeit unserer Drag-and-Drop-Uploads zu verbessern, empfehlen jedoch, eine ZIP-Datei zu verwenden, anstatt die Anwendungsordner per Drag-and-Drop zu verschieben.

Wir empfehlen dringend, eine .zip-Datei in einen Amazon S3 S3-Bucket hochzuladen, da dadurch Dateiuploads von der Amplify-Konsole vermieden werden und eine höhere Zuverlässigkeit bei manuellen Bereitstellungen gewährleistet wird. Die Integration von Amplify mit Amazon S3 vereinfacht diesen Prozess. Weitere Informationen finden Sie unter Bereitstellung einer statischen Website für Amplify aus einem Amazon S3 S3-Bucket.

Ich muss die Version Node.js meiner Anwendung aktualisieren

Die Amplify-Unterstützung für Apps, die die Versionen 14, 16 und 18 von Node.js verwenden, endet am 15. September 2025. Das Verhalten nach diesem Datum hängt von Ihrem Anwendungstyp ab:

- SSR-Anwendungen: Build-Fehler treten auf, wenn veraltete Versionen von Node.js verwendet werden. Sie können keine Updates bereitstellen, bis Sie ein Upgrade auf Node.js 20 oder höher durchgeführt haben.
- Nicht-SSR-Anwendungen: Sie können weiterhin veraltete Versionen von Node.js verwenden, wenn Sie sie manuell über Buildspec oder Live-Paket-Updates installieren.

Anwendungen, die bereits bereitgestellt wurden, werden unabhängig von der Version Node.js weiter ausgeführt.

Wenn Sie das Amazon Linux 2023 Build-Image verwenden, wird Node.js Version 20 standardmäßig unterstützt. Ab dem 15. September 2025 unterstützt das AL2023 Image automatisch Node.js 22 und ändert seine Standardversion Node.js von 18 auf 22.

Amazon Linux 2 (AL2) unterstützt Node.js Version 20 oder höher nicht automatisch. Wenn Sie derzeit verwenden AL2, empfehlen wir Ihnen, zu zu wechseln AL2023. Sie können das Build-Image in der Amplify-Konsole ändern. Sie können auch ein benutzerdefiniertes Build-Image verwenden, das die von Ihnen angegebene Version von Node.js unterstützt.

Wir empfehlen Ihnen, Ihre Anwendung vor dem Upgrade in einem neuen Branch zu testen, um sicherzustellen, dass sie ordnungsgemäß funktioniert.

Upgrade-Optionen

Konsole Amplify

Sie können die Funktion für Live-Paketaktualisierungen in der Amplify-Konsole verwenden, um die zu verwendende Version von Node.js anzugeben. Detaillierte Anweisungen finden Sie unter Verwenden bestimmter Paket- und Abhängigkeitsversionen im Build-Image.

Benutzerdefiniertes Build-Image

Wenn Sie ein benutzerdefiniertes Build-Image verwenden und NVM auf Ihrem Image installiert ist, können Sie es `nvm install 20` zu Ihrem Dockerfile hinzufügen. Weitere Informationen zu den Anforderungen und Konfigurationsanweisungen für ein benutzerdefiniertes Build-Image finden Sie unter. Anpassen des Build-Images

Einstellungen erstellen

Sie können die zu verwendende Version von Node.js in den `amplify.yml` Build-Einstellungen Ihrer Anwendung angeben, indem Sie den `nvm use` Befehl zum Abschnitt PreBuild-Befehle hinzufügen. Anweisungen zum Aktualisieren der Build-Einstellungen einer Anwendung finden Sie unterKonfiguration der Build-Einstellungen für eine Amplify-Anwendung.

Das folgende Beispiel zeigt, wie Sie die Buildeinstellungen anpassen, um die Standardversion von Node.js auf Node.js 18 festzulegen und ein Upgrade auf Node.js Version 20 in einem Testzweig mit dem Namen durchzuführen`node-20`.

```
frontend:
  phases:
    preBuild:
      commands:
        - nvm use 18
        - if [ "${AWS_BRANCH}" = "node-20" ]; then nvm use 20; fi
```

> **Warning**
>
> Beachten Sie, dass `preBuild` Befehle nach Live-Paketaktualisierungen ausgeführt werden. Die durch den `nvm use` Befehl angegebene Version von Node.js überschreibt die durch Live-Paketaktualisierungen festgelegte Version von Node.js.

Behebung von Problemen mit dem Amazon Linux 2023 Build Image

Die folgenden Informationen können Ihnen bei der Behebung von Problemen mit dem Build-Image von Amazon Linux 2023 (AL2023) helfen.

Themen

- Ich möchte Amplify-Funktionen mit der Python-Laufzeit ausführen
- Ich möchte Befehle ausführen, für die Superuser- oder Root-Rechte erforderlich sind

Ich möchte Amplify-Funktionen mit der Python-Laufzeit ausführen

Amplify Hosting verwendet jetzt standardmäßig das Amazon Linux 2023 Build-Image, wenn Sie eine neue Anwendung bereitstellen. AL2023 ist mit den Python-Versionen 3.8, 3.9, 3.10 und 3.11 vorinstalliert.

Aus Gründen der Abwärtskompatibilität mit dem Amazon Linux 2-Image sind auf dem AL2023 Build-Image Symlinks für ältere Versionen von Python vorinstalliert.

Standardmäßig wird Python Version 3.10 global verwendet. Um Ihre Funktionen mit einer bestimmten Python-Version zu erstellen, führen Sie die folgenden Befehle in der Build-Spezifikationsdatei Ihrer Anwendung aus.

```
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
```

Ich möchte Befehle ausführen, für die Superuser- oder Root-Rechte erforderlich sind

Wenn Sie das Amazon Linux 2023 Build-Image verwenden und bei der Ausführung von Systembefehlen, die Superuser- oder Root-Rechte erfordern, eine Fehlermeldung erhalten, müssen Sie diese Befehle mit dem `sudo` Linux-Befehl ausführen. Wenn bei der Ausführung beispielsweise ein Fehler auftritt`yum install -y gcc`, verwenden Sie`sudo yum install -y gcc`.

Das Amazon Linux 2-Build-Image verwendete den Root-Benutzer, aber das AL2023 Image von Amplify führt Ihren Code mit einem benutzerdefinierten `amplify` Benutzer aus. Amplify gewährt

diesem Benutzer Rechte, Befehle mit dem `sudo` Linux-Befehl auszuführen. Es hat sich bewährt, es `sudo` für Befehle zu verwenden, für die Superuser-Rechte erforderlich sind.

Behebung von Build-Problemen

Wenn Sie beim Erstellen oder Erstellen einer Amplify-Anwendung auf Probleme stoßen, finden Sie in den Themen in diesem Abschnitt Hilfe.

Themen

Neue Commits in meinem Repository lösen keine Amplify-Builds aus

Wenn neue Commits in deinem Git-Repository keine Amplify-Builds auslösen, vergewissere dich, dass dein Webhook noch in deinem Repository vorhanden ist. Wenn er vorhanden ist, überprüfe den Verlauf der Webhook-Anfragen, um festzustellen, ob Fehler aufgetreten sind. Amplify hat eine Nutzlastgrößenbeschränkung von 256 KB für eingehende Webhooks. Wenn Sie einen Commit in Ihr Repository übertragen, das eine große Anzahl geänderter Dateien enthält, könnten Sie dieses Limit überschreiten und dazu führen, dass Builds nicht ausgelöst werden.

Mein Repository-Name wird beim Erstellen einer neuen Anwendung nicht in der Amplify-Konsole aufgeführt

Wenn Sie eine neue Anwendung in der Amplify-Konsole erstellen, können Sie auf der Seite Repository und Zweig hinzufügen aus den verfügbaren Repositorys Ihrer Organisation auswählen. Ihr Ziel-Repository wird möglicherweise nicht in der Liste angezeigt, wenn es nicht kürzlich aktualisiert wurde. Dies kann der Fall sein, wenn Ihre Organisation über eine große Anzahl von Repositorys verfügt. Um dieses Problem zu beheben, übertragen Sie einen Commit in das Repository und aktualisieren Sie dann die Repository-Liste in der Konsole. Dadurch sollte das Repository angezeigt werden.

Mein Build schlägt mit dem **`Cannot find module aws-exports`** Fehler fehl (nur Gen 1-Apps)

Wenn Ihre App die `aws-exports.js` Datei während eines Builds nicht finden kann, wird der folgende Fehler zurückgegeben.

```
TS2307: Cannot find module 'aws-exports'
```

Die Amplify-Befehlszeilenschnittstelle (CLI) generiert die `aws-exports.js` Datei während Ihres Backend-Builds. Um diesen Fehler zu beheben, müssen Sie eine `aws-exports.js` Datei zur Verwendung im Build erstellen. Fügen Sie Ihrer Build-Spezifikation den folgenden Code hinzu, um die Datei zu erstellen:

```
backend:
  phases:
    build:
      commands:
        - "# Execute Amplify CLI with the helper script"
        - amplifyPush --simple
```

Ein vollständiges Beispiel für die Build-Spezifikationseinstellungen für eine Amplify-App finden Sie unterYAML-Syntaxreferenz für die Build-Spezifikation.

Ich möchte ein Build-Timeout überschreiben

Das Standard-Build-Timeout beträgt 30 Minuten. Sie können das Standard-Build-Timeout mithilfe der `_BUILD_TIMEOUT` Umgebungsvariablen überschreiben. Das minimale Build-Timeout beträgt 5 Minuten. Das maximale Build-Timeout beträgt 120 Minuten.

Anweisungen zum Einstellen einer Umgebungsvariablen für eine App in der Amplify-Konsole finden Sie unterUmgebungsvariablen setzen.

Fehlerbehebung bei benutzerdefinierten Domains

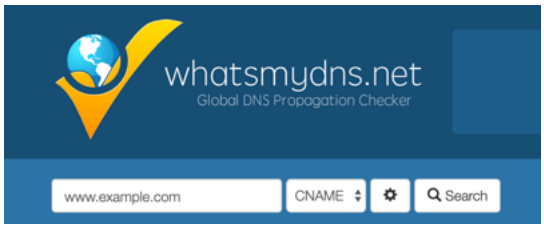
Wenn beim Verbinden einer benutzerdefinierten Domain mit Ihrer Amplify-Anwendung Probleme auftreten, finden Sie in den Themen in diesem Abschnitt Hilfe.

Wenn Sie hier keine Lösung für Ihr Problem finden, wenden Support Sie sich an. Weitere Informationen finden Sie unter Erstellen eines Support-Falls im Benutzerhandbuch von AWS Support .

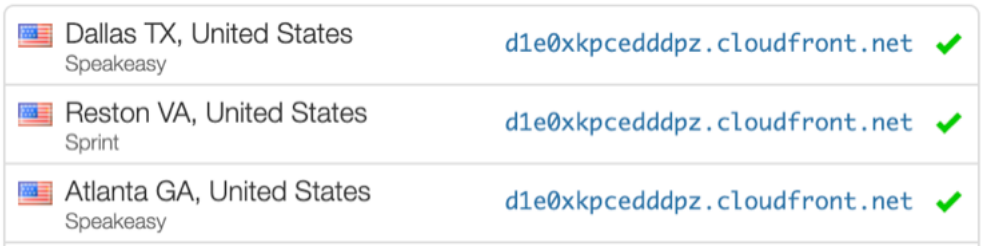
Themen

Ich muss überprüfen, ob mein CNAME behoben wird

1. Nachdem Sie Ihre DNS-Einträge bei Ihrem Drittanbieter für Domains aktualisiert haben, können Sie ein Tool wie dig oder eine kostenlose Website wie https://www.whatsmydns.net/ verwenden, um zu überprüfen, ob Ihr CNAME-Eintrag korrekt aufgelöst wird. Der folgende Screenshot zeigt, wie Sie whatsmydns.net verwenden, um Ihren CNAME-Eintrag für die Domain www.example.com zu überprüfen.



2. Wählen Sie Suchen und whatsmydns.net zeigt die Ergebnisse für Ihren CNAME an. Der folgende Screenshot ist ein Beispiel für eine Ergebnisliste, die bestätigt, dass der CNAME korrekt in eine cloudfront.net-URL aufgelöst wird.



Meine bei einem Drittanbieter gehostete Domain befindet sich im Status „Ausstehende Überprüfung“

1. Wenn bei Ihrer benutzerdefinierten Domain der Status „Überprüfung ausstehend“ nicht mehr angezeigt wird, überprüfen Sie, ob CNAME Ihre Daten behoben werden. Anweisungen zur Durchführung dieser Aufgabe finden Sie im vorherigen Thema zur CNAMEProblembehebung, Wie überprüfe ich, ob meine Probleme behoben sind?
2. Wenn Ihre CNAME Datensätze nicht aufgelöst werden, überprüfen Sie bei Ihrem Domain-Anbieter, ob der CNAME Eintrag in Ihren DNS-Einstellungen vorhanden ist.

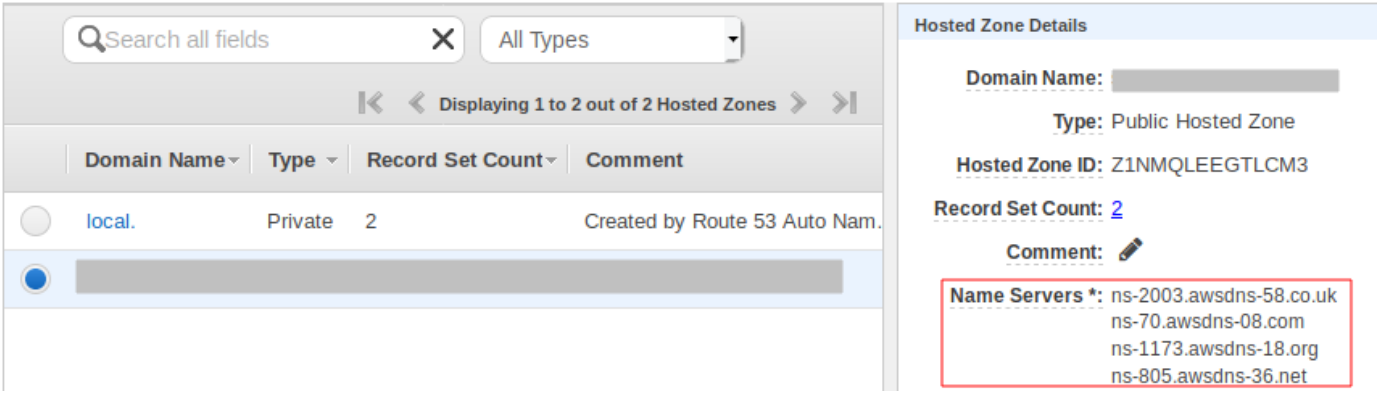
   > **Important**
   >
   > Es ist wichtig, dass Sie Ihre CNAME Einträge aktualisieren, sobald Sie Ihre benutzerdefinierte Domain erstellt haben. Nachdem Ihre App in der Amplify-Konsole erstellt wurde, wird Ihr CNAME Datensatz alle paar Minuten überprüft, um festzustellen, ob er aufgelöst wird. Wenn es nach einer Stunde nicht behoben wird, wird die Überprüfung alle paar Stunden durchgeführt, was zu einer Verzögerung bei der Einsatzbereitschaft Ihrer Domain führen kann. Wenn Sie Ihre CNAME Datensätze einige Stunden nach der Erstellung Ihrer App hinzugefügt oder aktualisiert haben, ist dies der wahrscheinlichste Grund dafür, dass Ihre App im Status „Ausstehende Überprüfung“ hängen bleibt.

3. Wenn Sie überprüft haben, dass der CNAME Eintrag existiert, liegt möglicherweise ein Problem mit Ihrem DNS-Anbieter vor. Sie können sich entweder an den DNS-Anbieter wenden, um zu diagnostizieren, warum die DNS-Überprüfung CNAME nicht funktioniert, oder Sie können Ihr DNS auf Route 53 migrieren. Weitere Informationen finden Sie unter Amazon Route 53 zum DNS-Service für eine bestehende Domain machen.

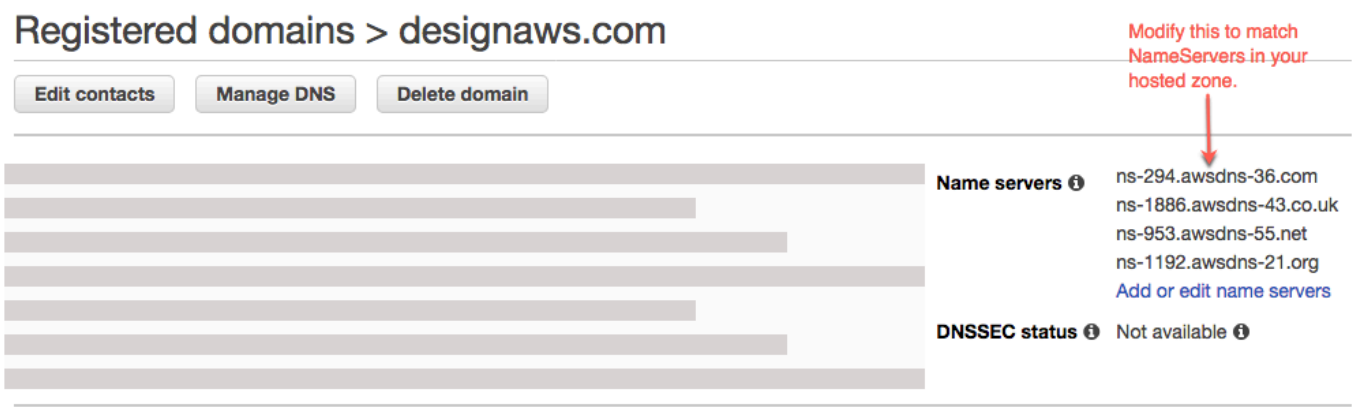
Meine mit Amazon Route 53 gehostete Domain befindet sich im Status „Ausstehende Überprüfung“

Wenn Sie Ihre Domain zu Amazon Route 53 übertragen haben, ist es möglich, dass Ihre Domain andere Nameserver hat als die, die von Amplify bei der Erstellung Ihrer App ausgestellt wurden. Führen Sie die folgenden Schritte aus, um die Ursache des Fehlers zu diagnostizieren.

1. Melden Sie sich bei der Amazon Route 53 53-Konsole an
2. Wählen Sie im Navigationsbereich Hosted Zones und dann den Namen der Domain aus, mit der Sie eine Verbindung herstellen.
3. Notieren Sie sich die Nameserver-Werte aus dem Abschnitt Hosted Zone Details. Sie benötigen diese Werte, um den nächsten Schritt abzuschließen. Der folgende Screenshot der Route 53-Konsole zeigt die Position der Nameserver-Werte in der unteren rechten Ecke.



4. Klicken Sie im Navigationsbereich auf Registered domains (Registrierte Domains). Stellen Sie sicher, dass die im Abschnitt Registrierte Domänen angezeigten Nameserver mit den Nameserver-Werten übereinstimmen, die Sie im vorherigen Schritt im Abschnitt Details zur gehosteten Zone aufgezeichnet haben. Wenn sie nicht übereinstimmen, bearbeiten Sie die Nameserver-Werte so, dass sie mit den Werten in Ihrer Hostzone übereinstimmen. Der folgende Screenshot der Route 53-Konsole zeigt die Position der Nameserver-Werte auf der rechten Seite.



5. Wenn das Problem dadurch nicht behoben wird, wenden Sie sich an Support. Weitere Informationen finden Sie unter Erstellen eines Support-Falls im Benutzerhandbuch von AWS Support .

Meine App mit mehrstufigen Subdomains befindet sich im Status „Ausstehende Überprüfung“

Wenn eine App mit mehrstufigen Subdomains beim Herstellen einer Verbindung zu einem DNS-Drittanbieter im Status „Ausstehende Überprüfung“ hängen bleibt, liegt möglicherweise ein Problem mit dem Format Ihrer DNS-Einträge vor. Einige DNS-Anbieter fügen Ihren Einträgen automatisch die Domainsuffixe Second-Level-Domain (SLD) und Top-Level-Domain (TLD) hinzu. Wenn Sie die Domain auch in dem Format angeben, das SLD und TLD enthält, kann dies zu Problemen bei der Domainverifizierung führen.

Wenn Sie eine Domain verbinden, versuchen Sie zunächst, den Domainnamen im vollständigen Format anzugeben, das beispielsweise `_hash.docs.backend.example.com` von Amplify bereitgestellt wird. Wenn die SSL-Konfiguration im Status „Ausstehende Überprüfung“ hängen bleibt, versuchen Sie, die TLD und SLD aus den Datensätzen zu entfernen. Wenn das vollständige Format beispielsweise lautet`_hash.docs.backend.example.com`, geben Sie an. `_hash.docs.backend` Warten Sie 15 bis 30 Minuten, bis sich die Datensätze verbreiten können. Verwenden Sie dann ein Tool wie MX Toolbox, um zu überprüfen, ob der Überprüfungsprozess funktioniert.

Mein DNS-Anbieter unterstützt keine A-Einträge mit vollständig qualifizierten Domainnamen

Einige DNS-Anbieter unterstützen keine A-Einträge mit einem vollqualifizierten Domainnamen (FQDN), wie `example.cloudfront.net` z. Cloudflare A records kann beispielsweise nur IPv4

Adressen schreiben und unterstützt dies nicht. FQDNs Um diese Einschränkung zu umgehen, empfehlen wir, CNAME Datensätze anstelle von Datensätzen A records in Ihrer DNS Konfiguration zu verwenden.

Als Beispiel verwendet die folgende DNS Konfiguration eineA record.

```
A     | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Ändern Sie sie auf die folgende DNS Konfiguration, um nur CNAME Datensätze zu verwenden.

```
CNAME | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Diese Problemumgehung ermöglicht es Ihnen, Ihre Apex-Domain (@-Eintrag) korrekt auf Dienste wie CloudFront zu verweisen und gleichzeitig die Beschränkung „ IPv4Nur“ A records im Cloudflare-System zu umgehen.

Ich erhalte eine Fehlermeldung CNAMEAlready ExistsException

Wenn Sie eine CNAMEAlreadyExistsExceptionFehlermeldung erhalten, bedeutet dies, dass einer der Hostnamen, mit denen Sie versucht haben, eine Verbindung herzustellen (eine Subdomain oder die Apex-Domain), bereits für eine andere CloudFront Amazon-Distribution bereitgestellt ist. Die Ursache Ihres Fehlers hängt von Ihren aktuellen Hosting- und DNS-Anbietern ab.

Ein CNAME Alias wie `example.com` oder `sub.example.com` kann jeweils nur einer CloudFront Distribution zugeordnet werden. Das CNAMEAlreadyExistsExceptionbedeutet, dass Ihre Domain bereits mit einer anderen CloudFront Distribution verknüpft ist, entweder innerhalb desselben AWS-Konto oder möglicherweise in einem anderen Konto. Die Domain muss von der vorherigen CloudFront Distribution getrennt werden, bevor die von Amplify Hosting erstellte neue Distribution funktioniert. Möglicherweise müssen Sie mehr als ein Konto überprüfen, wenn Sie oder Ihre Organisation mehrere AWS-Konto s besitzen.

Führen Sie die folgenden Schritte aus, um die Ursache des CNAMEAlreadyExistsExceptionFehlers zu diagnostizieren.

1. Melden Sie sich bei der CloudFront Amazon-Konsole an und vergewissern Sie sich, dass Sie diese Domain nicht für eine andere Distribution bereitgestellt haben. Ein einzelner CNAME Datensatz kann jeweils an eine CloudFront Distribution angehängt werden.

2. Wenn Sie die Domain zuvor für eine CloudFront Distribution bereitgestellt haben, müssen Sie sie entfernen.

   a. Wählen Sie im linken Navigationsmenü Distributionen aus.

   b. Wählen Sie den Namen der Distribution aus, die bearbeitet werden soll.

   c. Wählen Sie die Registerkarte Allgemein. Wählen Sie im Abschnitt Settings (Einstellungen) die Option Edit (Bearbeiten) aus.

   d. Entfernen Sie den Domainnamen aus dem alternativen Domainnamen (CNAME). Wählen Sie dann „Änderungen speichern“.

3. Vergewissern Sie sich, dass keine andere CloudFront Distribution existiert, die diese Domain in der aktuellen AWS-Konto oder anderen verwendet AWS-Konten. Wenn dadurch keine aktuell laufenden Dienste gestört werden, versuchen Sie, die gehostete Zone zu löschen und neu zu erstellen.

4. Überprüfe, ob diese Domain mit einer anderen Amplify-App verbunden ist, die du besitzt. Stellen Sie in diesem Fall sicher, dass Sie nicht versuchen, einen der Hostnamen wiederzuwenden. Wenn Sie eine andere App verwenden`www.example.com`, können Sie sie nicht `www.example.com` mit der App verwenden, mit der Sie gerade eine Verbindung herstellen. Sie können andere Subdomains verwenden, wie `blog.example.com` z.

5. Wenn diese Domain erfolgreich mit einer anderen App verbunden und dann innerhalb der letzten Stunde gelöscht wurde, versuchen Sie es nach mindestens einer Stunde erneut. Wenn Sie diese Ausnahme nach 6 Stunden immer noch sehen, wenden Sie sich an Support. Weitere Informationen finden Sie unter Erstellen eines Support-Falls im Benutzerhandbuch von AWS Support .

6. Wenn Sie Ihre Domain über Route 53 verwalten, stellen Sie sicher, dass Sie alle gehosteten Zonen CNAME oder ALIAS Datensätze bereinigen, die auf die alte CloudFront Distribution verweisen.

7. Nachdem Sie die vorherigen Schritte abgeschlossen haben, entfernen Sie die benutzerdefinierte Domain aus Amplify Hosting und beginnen Sie erneut mit dem Workflow, um eine benutzerdefinierte Domain in der Amplify-Konsole zu verbinden.

Ich erhalte die Fehlermeldung „Zusätzliche Überprüfung erforderlich“

Wenn Sie die Fehlermeldung „Zusätzliche Überprüfung erforderlich“ erhalten, bedeutet dies, dass AWS Certificate Manager (ACM) zusätzliche Informationen benötigt, um diese Zertifikatsanforderung zu bearbeiten. Dies kann als Betrugsschutzmaßnahme geschehen, z. B. wenn die Domäne unter

den Top 1000 Websites von Alexa rangiert. Um diese Informationen bereitzustellen, kontaktieren Sie über das Support-Center Support. Wenn Sie über keinen Supportplan verfügen, veröffentlichen Sie einen neuen Thread im ACM-Diskussionsforum.

> **Note**
>
> Sie können kein Zertifikat für Amazon-eigene Domainnamen, wie solche, die mit amazonaws.com, cloudfront.net oder elasticbeanstalk.com enden, anfordern.

Ich erhalte eine 404-Fehlermeldung für die URL CloudFront

Um Traffic bereitzustellen, verweist Amplify Hosting über einen CNAME-Record auf eine CloudFront URL. Beim Verbinden einer App mit einer benutzerdefinierten Domain zeigt die Amplify-Konsole die CloudFront URL für die App an. Über diese CloudFront URL können Sie jedoch nicht direkt auf Ihre Anwendung zugreifen. Es wird ein 404-Fehler zurückgegeben. Ihre Anwendung wird nur mithilfe der Amplify-App-URL (z. B.`https://main.d5udybEXAMPLE.amplifyapp.com`) oder Ihrer benutzerdefinierten Domain (zum Beispiel`www.example.com`) aufgelöst.

Amplify muss Anfragen an den richtigen bereitgestellten Branch weiterleiten und verwendet dazu den Hostnamen. Sie können beispielsweise die Domain `www.example.com` konfigurieren, die auf den Mainline-Zweig einer App verweist, aber auch konfigurieren, `dev.example.com` dass sie auf den Entwicklungszweig derselben App verweist. Daher müssen Sie Ihre Anwendung auf der Grundlage der konfigurierten Subdomains aufrufen, damit Amplify die Anfragen entsprechend weiterleiten kann.

Ich erhalte beim Besuch meiner Domain SSL-Zertifikat- oder HTTPS-Fehler

Wenn Sie DNS-Einträge für Certificate Authority Authorization (CAA) bei Ihrem DNS-Drittanbieter konfiguriert haben, kann AWS Certificate Manager (ACM) möglicherweise keine Zwischenzertifikate für Ihr benutzerdefiniertes Domain-SSL-Zertifikat aktualisieren oder neu ausstellen. Um dieses Problem zu lösen, müssen Sie einen CAA-Eintrag hinzufügen, um mindestens einer der Zertifizierungsstellen-Domains von Amazon zu vertrauen. Das folgende Verfahren beschreibt die Schritte, die Sie ausführen müssen.

So fügen Sie einen CAA-Eintrag hinzu, um einer Amazon-Zertifizierungsstelle zu vertrauen

1. Konfigurieren Sie einen CAA-Eintrag bei Ihrem Domain-Anbieter, um mindestens einer der Amazon-Zertifizierungsstellen-Domains zu vertrauen. Weitere Informationen zur Konfiguration

des CAA-Eintrags finden Sie unter Probleme mit der Zertifizierungsstellen-Autorisierung (CAA) im AWS Certificate Manager Benutzerhandbuch.

2. Verwenden Sie eine der folgenden Methoden, um Ihr SSL-Zertifikat zu aktualisieren:

   - Manuelles Update mit der Amplify-Konsole.

     > **Note**
     >
     > Diese Methode führt zu Ausfallzeiten für Ihre benutzerdefinierte Domain.

     a. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
     b. Wählen Sie Ihre App aus, zu der Sie einen CAA-Eintrag hinzufügen möchten.
     c. Wählen Sie im Navigationsbereich App-Einstellungen, Domainverwaltung aus.
     d. Löschen Sie auf der Seite Domainverwaltung die benutzerdefinierte Domain.
     e. Connect deine App erneut mit der benutzerdefinierten Domain. Durch diesen Vorgang wird ein neues SSL-Zertifikat ausgestellt, und die Zwischenzertifikate können jetzt von ACM verwaltet werden.

        Verwenden Sie eines der folgenden Verfahren, das dem von Ihnen verwendeten Domain-Anbieter entspricht, um Ihre App wieder mit Ihrer benutzerdefinierten Domain zu verbinden.

        - Hinzufügen einer von Amazon Route 53 verwalteten benutzerdefinierten Domain.
        - Hinzufügen einer benutzerdefinierten Domain, die von einem DNS-Drittanbieter verwaltet wird.
        - Aktualisierung von DNS-Einträgen für eine Domain, die verwaltet wird von GoDaddy.

   - Wenden Sie sich Support an uns, um Ihr SSL-Zertifikat erneut ausstellen zu lassen.

Pfadkomponenten werden in Domainumleitungen nicht unterstützt

Domainweiterleitungen stimmen nur mit dem Teil des Hostnamens überein. Pfadkomponenten in domänenbasierten Quellregeln (z. B. `"https://domain.com/path"`) werden nicht unterstützt und führen dazu, dass die Regel fehlerfrei ignoriert wird. Weitere Informationen finden Sie unter Leitet die Beispielreferenz weiter und schreibt sie neu.

Ich erhalte die Fehlermeldung 400 für die kontoübergreifende Domänenzuweisung

Wenn DomainAssociation Sie eine Anfrage für eine Amplify-App mit einer Domain initiieren, die bereits oder zuvor mit verschiedenen Amplify-Apps in anderen AWS-Konten in derselben Region verknüpft ist oder war, wird dies als kontoübergreifende Domain-Zuordnung betrachtet. Wenn Sie diesen Fehler erhalten, bedeutet dies, dass Sie versuchen, eine kontoübergreifende Domainverknüpfung zu verknüpfen, was eine manuelle Überprüfung erfordert. Wenn Sie mit einer kontoübergreifenden Domain-Zuordnung fortfahren möchten, wenden Sie sich bitte an den AWS-Support, um Unterstützung zu erhalten.

Fehlerbehebung bei serverseitig gerenderten Anwendungen

Wenn bei der Bereitstellung einer SSR-App mit Amplify Hosting Compute unerwartete Probleme auftreten, lesen Sie die folgenden Themen zur Fehlerbehebung. Wenn Sie hier keine Lösung für Ihr Problem finden, lesen Sie den Leitfaden zur Fehlerbehebung bei SSR-Webcomputern im Repository Amplify Hosting GitHub Issues.

Topics

Ich benötige Hilfe bei der Verwendung eines Framework-Adapters

Wenn Sie Probleme bei der Bereitstellung einer SSR-App haben, die einen Framework-Adapter verwendet, finden Sie weitere Informationen unterVerwendung von Open-Source-Adaptern für jedes SSR-Framework.

Edge-API-Routen führen dazu, dass mein Build Next.js fehlschlägt

Derzeit unterstützt Amplify keine Next.js Edge-API-Routen. Sie müssen Non-Edge APIs - und Middleware verwenden, wenn Sie Ihre App mit Amplify hosten.

Die inkrementelle statische Regeneration auf Abruf funktioniert für meine App nicht

Ab Version 12.2.0 unterstützt Next.js die inkrementelle statische Regeneration (ISR), um den Next.js Cache für eine bestimmte Seite manuell zu leeren. Amplify unterstützt derzeit jedoch kein On-Demand-ISR. Wenn Ihre App die On-Demand-Revalidierung von Next.js verwendet, funktioniert diese Funktion nicht, wenn Sie Ihre App auf Amplify bereitstellen.

Die Build-Ausgabe meiner Anwendung überschreitet die maximal zulässige Größe

Derzeit beträgt die maximale Build-Ausgabegröße, die Amplify für SSR-Apps unterstützt, 220 MB. Wenn Sie eine Fehlermeldung erhalten, die besagt, dass die Größe der Build-Ausgabe Ihrer App die maximal zulässige Größe überschreitet, müssen Sie Maßnahmen ergreifen, um sie zu reduzieren.

Um die Größe der Build-Ausgabe einer App zu reduzieren, können Sie die Build-Artefakte der App überprüfen und alle großen Abhängigkeiten identifizieren, die aktualisiert oder entfernt werden müssen. Laden Sie zunächst die Build-Artefakte auf Ihren lokalen Computer herunter. Überprüfen Sie dann die Größe der Verzeichnisse. Das `node_modules` Verzeichnis kann beispielsweise Binärdateien wie die Server-Runtime-Dateien von Next.js enthalten `@swc` und auf `@esbuild` die von diesen verwiesen wird. Da diese Binärdateien in der Laufzeit nicht erforderlich sind, können Sie sie nach dem Build löschen.

Verwenden Sie die folgenden Anweisungen, um die Build-Ausgabe einer App herunterzuladen und die Größe der Verzeichnisse mithilfe der AWS Command Line Interface (CLI) zu überprüfen.

Um die Build-Ausgabe für eine Next.js App herunterzuladen und zu überprüfen

1. Öffnen Sie ein Terminalfenster und führen Sie den folgenden Befehl aus. Ändern Sie die App-ID, den Zweignamen und die Job-ID nach Ihren eigenen Angaben. Verwenden Sie für die Job-ID die Buildnummer für den fehlgeschlagenen Build, den Sie untersuchen.

   ```
   aws amplify get-job --app-id abcd1234 --branch-name main --job-id 2
   ```

2. Suchen Sie in der Terminalausgabe im `stepName: "BUILD"` Abschnitt,, nach der URL für vorsignierte Artefakte. `job steps` Die URL ist in der folgenden Beispielausgabe rot hervorgehoben.

```
"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. Kopieren Sie die URL und fügen Sie sie in ein Browserfenster ein. Eine `artifacts.zip` Datei wird auf Ihren lokalen Computer heruntergeladen. Dies ist Ihre Build-Ausgabe.

4. Führen Sie den Befehl du disk usage aus, um die Größe der Verzeichnisse zu überprüfen. Der folgende Beispielbefehl gibt die Größe der `static` Verzeichnisse `compute` und zurück.

```
du -csh compute static
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe mit Größeninformationen für die `static` Verzeichnisse `compute` und.

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

5. Öffnen Sie das `compute` Verzeichnis und suchen Sie den `node_modules` Ordner. Suchen Sie in Ihren Abhängigkeiten nach Dateien, die Sie aktualisieren oder entfernen können, um die Größe des Ordners zu verringern.
6. Wenn Ihre App Binärdateien enthält, die in der Runtime nicht benötigt werden, löschen Sie sie nach dem Build, indem Sie die folgenden Befehle zum Build-Abschnitt der `amplify.yml` App-Datei hinzufügen.

```
- 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
```

Im Folgenden finden Sie ein Beispiel für den Abschnitt mit Build-Befehlen in einer `amplify.yml` Datei, in dem diese Befehle nach der Ausführung eines Produktions-Builds hinzugefügt wurden.

```
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
```

Mein Build schlägt mit einem Fehler wegen unzureichenden Speichers fehl

Next.js ermöglicht es Ihnen, Build-Artefakte zwischenzuspeichern, um die Leistung bei nachfolgenden Builds zu verbessern. Darüber hinaus komprimiert der AWS CodeBuild Container von Amplify diesen Cache und lädt ihn in Ihrem Namen auf Amazon S3 hoch, um die nachfolgende Build-Leistung zu verbessern. Dies könnte dazu führen, dass Ihr Build mit einem Fehler wegen unzureichenden Speichers fehlschlägt.

Führen Sie die folgenden Aktionen aus, um zu verhindern, dass Ihre App das Speicherlimit während der Erstellungsphase überschreitet. Entfernen Sie es zunächst `.next/cache/**/*` aus dem Abschnitt cache.paths Ihrer Build-Einstellungen. Als Nächstes entfernen Sie die `NODE_OPTIONS` Umgebungsvariable aus Ihrer Build-Einstellungsdatei. Stellen Sie stattdessen die `NODE_OPTIONS` Umgebungsvariable in der Amplify-Konsole ein, um das maximale Speicherlimit für den Knoten zu definieren. Weitere Informationen zum Einstellen von Umgebungsvariablen mit der Amplify-Konsole finden Sie unterUmgebungsvariablen setzen.

Nachdem Sie diese Änderungen vorgenommen haben, versuchen Sie es erneut mit Ihrem Build. Wenn dies erfolgreich ist, fügen Sie `.next/cache/**/*` dem Abschnitt cache.paths Ihrer Build-Einstellungsdatei etwas hinzu.

Weitere Informationen zur Cache-Konfiguration von Next.js zur Verbesserung der Build-Leistung finden Sie unter AWS CodeBuild auf der Website Next.js.

Die Größe der HTTP-Antwort meiner Anwendung ist zu groß

Derzeit beträgt die maximale Antwortgröße, die Amplify für Apps von Next.js 12 und höher unterstützt, die die Web Compute-Plattform verwenden, 5,72 MB. Antworten, die dieses Limit überschreiten, geben 504 Fehler ohne Inhalt an die Clients zurück.

Wie messe ich die Startzeit meiner Compute-App lokal?

Verwenden Sie die folgenden Anweisungen, um die lokale initialization/start Betriebszeit für Ihre Compute-App Next.js 12 oder höher zu ermitteln. Sie können die Leistung Ihrer App lokal mit der Leistung von Amplify Hosting vergleichen und die Ergebnisse verwenden, um die Leistung Ihrer App zu verbessern.

Um die Initialisierungszeit einer Next.js Compute-App lokal zu messen

1. Öffnen Sie die `next.config.js` Datei der App und stellen Sie die `output` Option `standalone` wie folgt ein.

   ```
   ** @type {import('next').NextConfig} */
   const nextConfig = {
     // Other options
     output: "standalone",
   };

   module.exports = nextConfig;
   ```

2. Öffnen Sie ein Terminalfenster und führen Sie den folgenden Befehl aus, um die App zu erstellen.

   ```
   next build
   ```

3. Führen Sie den folgenden Befehl aus, um den `.next/static` Ordner zu kopieren`.next/standalone/.next/static`.

```
cp -r .next/static .next/standalone/.next/static
```

4. Führen Sie den folgenden Befehl aus, um den `public` Ordner zu kopieren`.next/standalone/public`.

```
cp -r public .next/standalone/public
```

5. Führen Sie den folgenden Befehl aus, um den Server Next.js zu starten.

```
node .next/standalone/server.js
```

6. Beachten Sie, wie lange es zwischen der Ausführung des Befehls in Schritt 5 und dem Starten des Servers dauert. Wenn der Server einen Port abhört, sollte er die folgende Meldung ausgeben.

```
Listening on port 3000
```

7. Beachten Sie, wie lange es dauert, bis alle anderen Module nach dem Start des Servers in Schritt 6 geladen sind. Zum Beispiel `bugsnag` dauert das Laden von Bibliotheken wie 10—12 Sekunden. Nach dem Laden wird die Bestätigungsmeldung angezeigt`[bugsnag] loaded`.
8. Addieren Sie die Zeitdauern aus Schritt 6 und Schritt 7. Dieses Ergebnis ist die lokale initialization/start Betriebszeit Ihrer Compute-App.

Mein Build schlägt mit einem veralteten Versionsfehler von Node.js fehl

Problem: Der Build Ihrer SSR-Anwendung schlägt fehl und der Fehler Node.js Version wird nicht unterstützt.

```
# NODE.JS VERSION NOT SUPPORTED
================================================================================
Your application uses Node.js v18.x.x, which is no longer supported.
AWS Amplify Console has ended support for Node.js 14, Node.js 16 and Node.js 18.

To deploy your application, please upgrade to Node.js 20 or later.

For detailed migration guidelines, visit: https://docs.aws.amazon.com/amplify/latest/
userguide/troubleshooting-general.html#update-node-version
================================================================================
```

Ursache: Ihre SSR-Anwendung wurde mit einer veralteten Version von Node.js (14.x, 16.x oder 18.x) erstellt. Mit Wirkung zum 15. September 2025 blockiert Amplify die Bereitstellung von SSR-Anwendungen, die diese veralteten Versionen während des Build-Prozesses verwenden.

Aktualisieren Sie Ihre Build-Umgebung, um Node.js 2.0 oder höher zu verwenden. Detaillierte Anweisungen finden Sie unter Ich muss die Version Node.js meiner Anwendung aktualisieren.

Problembehandlung bei Umleitungen und Umschreibungen

Wenn Sie beim Einrichten von Weiterleitungen und Umschreibungen für eine Amplify-Anwendung auf Probleme stoßen, finden Sie in den Themen in diesem Abschnitt Hilfe.

Topics

Der Zugriff wird für bestimmte Routen auch mit der SPA-Umleitungsregel verweigert.

Wenn für bestimmte Routen mit einer SPA-Umleitungsregel die Fehlermeldung „Zugriff verweigert“ angezeigt wird, ist diese `baseDirectory` möglicherweise in den Build-Einstellungen der App nicht korrekt festgelegt. Wenn das Frontend Ihrer App beispielsweise für das `build` Verzeichnis erstellt wurde, müssen Ihre Build-Einstellungen auch auf das `build` Verzeichnis verweisen. Das folgende Beispiel für eine Build-Spezifikation veranschaulicht diese Einstellung.

```
frontend:
  artifacts:
    baseDirectory: build
    files:
        - "**/*"
```

Ein vollständiges Beispiel für die Build-Spezifikationseinstellungen für eine Amplify-App finden Sie unter YAML-Syntaxreferenz für die Build-Spezifikation

Ich möchte einen Reverse-Proxy für eine API einrichten

Sie können den folgenden JSON-Code verwenden, um einen Reverse-Proxy für einen dynamischen Endpunkt einzurichten.

```
[
  {
    "source": "/documents/<*>",
    "target": "https://otherdomain/resource/<*>",
    "status": "200",
    "condition": null
  }
]
```

Ein grundlegendes Beispiel für die Erstellung eines Reverse-Proxys für Ihre Amplify-App zu einer Drittanbieter-API finden Sie unterReverse-Proxy umschreiben.

Fehlerbehebung beim Caching

Wenn Sie Probleme mit dem Zwischenspeichern für eine Amplify-Anwendung haben, finden Sie in den Themen in diesem Abschnitt Hilfe.

Topics

- Ich möchte die Größe des Caches für eine App reduzieren
- Ich möchte das Lesen aus dem Cache für eine App deaktivieren

Ich möchte die Größe des Caches für eine App reduzieren

Wenn Sie den Cache verwenden, zwischenspeichern Sie möglicherweise Zwischendateien, die zwischen den Builds nicht bereinigt wurden. Durch das Zwischenspeichern dieser selten verwendeten Dateien wird die Größe Ihres Caches erhöht. Um dies zu verhindern, können Sie bestimmte Ordner vom Zwischenspeichern ausschließen, indem Sie die `!` Anweisung im `cache` Abschnitt der Build-Spezifikation Ihrer App verwenden.

Das folgende Beispiel für Build-Einstellungen zeigt, wie Sie die `!` Direktive verwenden, um einen Ordner anzugeben, den Sie nicht zwischenspeichern möchten.

```
cache:
  paths:
    - node_modules/**/*
    - "!node_modules/path/not/to/cache"
```

Wenn Sie den `node_modules` Ordner zwischenspeichern, `node_modules/.cache` wird dieser Wert standardmäßig weggelassen.

Ein vollständiges Beispiel für die Build-Spezifikationseinstellungen für eine Amplify-App finden Sie unter YAML-Syntaxreferenz für die Build-Spezifikation

Ich möchte das Lesen aus dem Cache für eine App deaktivieren

Wenn Sie das Lesen aus dem Cache für eine App deaktivieren möchten, entfernen Sie den Cache-Abschnitt aus der Build-Spezifikation Ihrer App.

Amplify-Zugriff auf Repositorys einrichten GitHub

Amplify verwendet jetzt die GitHub Apps-Funktion, um Amplify den schreibgeschützten Zugriff auf Repositorys zu autorisieren. GitHub Mit der Amplify GitHub App werden die Berechtigungen genauer abgestimmt, sodass Sie Amplify nur Zugriff auf die von Ihnen angegebenen Repositorys gewähren können. Weitere Informationen zu GitHub Apps finden Sie auf der Website unter Über GitHub Apps. GitHub

Wenn Sie eine neue App verbinden, die in einem GitHub Repo gespeichert ist, verwendet Amplify standardmäßig die GitHub App, um auf das Repo zuzugreifen. Bestehende Amplify-Apps, die Sie zuvor über GitHub Repos verbunden haben, werden jedoch OAuth für den Zugriff verwendet. CI/CD wird weiterhin für diese Apps funktionieren, aber wir empfehlen Ihnen dringend, sie zu migrieren, um die neue Amplify GitHub App zu verwenden.

Wenn Sie eine neue App bereitstellen oder eine bestehende App mithilfe der Amplify-Konsole migrieren, werden Sie automatisch zum Installationsort für die GitHub Amplify-App weitergeleitet. Um manuell auf die Installations-Landingpage für die App zuzugreifen, öffnen Sie einen Webbrowser und navigieren Sie nach Region zur App. Verwenden Sie das Format `https://github.com/apps/aws-amplify-`*`REGION`* und *`REGION`* ersetzen Sie es durch die Region, in der Sie Ihre Amplify-App bereitstellen werden. Um beispielsweise die Amplify GitHub App in der Region USA West (Oregon) zu installieren, navigieren Sie zu https://github.com/apps/ aws-amplify-us-west -2.

Topics

- Installation und Autorisierung der Amplify GitHub App für eine neue Bereitstellung
- Migration einer vorhandenen OAuth App zur GitHub Amplify App
- Einrichtung der Amplify GitHub App für CLI CloudFormation- und SDK-Bereitstellungen

- Webvorschauen mit der GitHub Amplify App einrichten

Installation und Autorisierung der Amplify GitHub App für eine neue Bereitstellung

Wenn Sie aus vorhandenem Code in einem GitHub Repo eine neue App für Amplify bereitstellen, verwenden Sie die folgenden Anweisungen, um die App zu installieren und zu autorisieren. GitHub

Um die GitHub Amplify App zu installieren und zu autorisieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie auf der Seite Alle Apps die Option Neue App und dann Host-Web-App aus.
3. Wählen Sie auf der Seite Erste Schritte mit Amplify Hosting die Option GitHubund anschließend Weiter aus.
4. Wenn Sie zum ersten Mal eine Verbindung zu einem GitHub Repository herstellen, wird in Ihrem Browser auf GitHub .com eine neue Seite geöffnet, auf der Sie um Erlaubnis zur Autorisierung AWS Amplify in Ihrem GitHub Konto gebeten werden. Klicken Sie auf Authorize.
5. Als Nächstes müssen Sie die Amplify GitHub App in Ihrem GitHub Konto installieren. Auf GitHub.com wird eine Seite geöffnet, auf der um Erlaubnis zur Installation und Autorisierung AWS Amplify in Ihrem Konto gebeten wird. GitHub
6. Wählen Sie das GitHub Konto aus, auf dem Sie die Amplify GitHub App installieren möchten.
7. Führen Sie eine der folgenden Aktionen aus:
   - Um die Installation auf alle Repositorys anzuwenden, wählen Sie Alle Repositorys.
   - Um die Installation auf die von Ihnen ausgewählten Repositorys zu beschränken, wählen Sie Nur ausgewählte Repositorys. Stellen Sie sicher, dass das Repo für die App, die Sie migrieren, in die von Ihnen ausgewählten Repos aufgenommen wird.
8. Wählen Sie Installieren und autorisieren.
9. Sie werden zur Zweigseite Repository hinzufügen für Ihre App in der Amplify-Konsole weitergeleitet.
10. Wählen Sie in der Liste der kürzlich aktualisierten Repositorys den Namen des Repositorys aus, zu dem Sie eine Verbindung herstellen möchten.
11. Wählen Sie in der Branch-Liste den Namen des Repository-Branches aus, zu dem Sie eine Verbindung herstellen möchten.
12. Wählen Sie Weiter aus.

13. Wählen Sie auf der Seite Build-Einstellungen konfigurieren die Option Weiter aus.
14. Wählen Sie auf der Seite Überprüfen die Option Speichern und bereitstellen aus.

Migration einer vorhandenen OAuth App zur GitHub Amplify App

Bestehende Amplify-Apps, die Sie zuvor über GitHub Repositorys verbunden haben, werden OAuth für den Repo-Zugriff verwendet. Wir empfehlen dringend, dass Sie diese Apps migrieren, um die Amplify GitHub App zu verwenden.

Verwenden Sie die folgenden Anweisungen, um eine App zu migrieren und den entsprechenden OAuth Webhook in Ihrem GitHub Konto zu löschen. Beachten Sie, dass das Verfahren für die Migration davon abhängt, ob die GitHub Amplify-App bereits installiert ist. Nachdem Sie Ihre erste App migriert und die App installiert und autorisiert haben, müssen Sie nur die Repository-Berechtigungen für nachfolgende GitHub App-Migrationen aktualisieren.

Um eine App von OAuth zur App zu migrieren GitHub

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amplify-Konsole.
2. Wählen Sie die App aus, die Sie migrieren möchten.
3. Suchen Sie auf der Informationsseite der App nach der blauen Meldung Zu unserer GitHub App migrieren und wählen Sie Migration starten aus.
4. Wählen Sie auf der Seite GitHub App installieren und autorisieren die Option GitHub App konfigurieren aus.
5. In Ihrem Browser wird auf GitHub .com eine neue Seite geöffnet, auf der Sie um Erlaubnis zur Autorisierung AWS Amplify in Ihrem GitHub Konto gebeten werden. Klicken Sie auf Authorize.
6. Wählen Sie das GitHub Konto aus, auf dem Sie die Amplify GitHub App installieren möchten.
7. Führen Sie eine der folgenden Aktionen aus:
   - Um die Installation auf alle Repositorys anzuwenden, wählen Sie Alle Repositorys.
   - Um die Installation auf die von Ihnen ausgewählten Repositorys zu beschränken, wählen Sie Nur ausgewählte Repositorys. Stellen Sie sicher, dass das Repo für die App, die Sie migrieren, in die von Ihnen ausgewählten Repositorys aufgenommen wird.
8. Wählen Sie Installieren und autorisieren.
9. Sie werden auf die Seite GitHub App installieren und autorisieren für Ihre App in der Amplify-Konsole weitergeleitet. Wenn die GitHub Autorisierung erfolgreich war, wird eine Erfolgsmeldung angezeigt. Wählen Sie „Weiter“.

10. Wählen Sie auf der Seite Vollständige Installation die Option Vollständige Installation aus. Mit diesem Schritt wird Ihr vorhandener Webhook gelöscht, ein neuer erstellt und die Migration abgeschlossen.

Einrichtung der Amplify GitHub App für CLI CloudFormation- und SDK-Bereitstellungen

Bestehende Amplify-Apps, die Sie zuvor über GitHub Repositorys verbunden haben, werden OAuth für den Repo-Zugriff verwendet. Dies kann Apps einschließen, die Sie über die Amplify Command Line Interface (CLI) bereitgestellt haben CloudFormation, oder die SDKs. Wir empfehlen Ihnen dringend, diese Apps zu migrieren, um die neue Amplify GitHub App zu verwenden. Die Migration muss in der Amplify-Konsole in der AWS-Managementkonsole durchgeführt werden. Detaillierte Anweisungen finden Sie unter Migration einer vorhandenen OAuth App zur GitHub Amplify App.

Sie können die Amplify-CLI und die verwenden CloudFormation, SDKs um eine neue Amplify-App bereitzustellen, die die GitHub App für den Repo-Zugriff verwendet. Für diesen Vorgang müssen Sie zuerst die Amplify GitHub App in Ihrem GitHub Konto installieren. Als Nächstes müssen Sie ein persönliches Zugriffstoken in Ihrem GitHub Konto generieren. Stellen Sie abschließend die App bereit und geben Sie das persönliche Zugriffstoken an.

Installieren Sie die Amplify GitHub App in Ihrem Konto

1. Öffnen Sie einen Webbrowser und navigieren Sie zum Installationsort für die Amplify GitHub App in der AWS Region, in der Sie Ihre App bereitstellen werden.

   Verwenden Sie das Format `https://github.com/apps/aws-amplify-`*`REGION`*`/installations/new` und *`REGION`* ersetzen Sie es durch Ihre eigene Eingabe. Wenn Sie Ihre App beispielsweise in der Region USA West (Oregon) installieren, geben Sie an`https://github.com/apps/aws-amplify-us-west-2/installations/new`.
2. Wählen Sie das GitHub Konto aus, auf dem Sie die GitHub Amplify-App installieren möchten.
3. Führen Sie eine der folgenden Aktionen aus:
   - Um die Installation auf alle Repositorys anzuwenden, wählen Sie Alle Repositorys.
   - Um die Installation auf die von Ihnen ausgewählten Repositorys zu beschränken, wählen Sie Nur ausgewählte Repositorys. Stellen Sie sicher, dass das Repo für die App, die Sie migrieren, in die von Ihnen ausgewählten Repos aufgenommen wird.
4. Wählen Sie Installieren aus.

Generieren Sie ein persönliches Zugriffstoken in Ihrem Konto GitHub

1. Loggen Sie sich in Ihr GitHub Konto ein.
2. Suchen Sie in der oberen rechten Ecke Ihr Profilfoto und wählen Sie im Menü Einstellungen aus.
3. Wählen Sie im linken Navigationsmenü die Option Entwicklereinstellungen aus.
4. Wählen Sie auf der GitHub Apps-Seite im linken Navigationsmenü die Option Persönliche Zugriffstoken aus.
5. Wählen Sie auf der Seite Persönliche Zugriffstoken die Option Neues Token generieren aus.
6. Geben Sie auf der Seite Neues persönliches Zugriffstoken für Notiz einen aussagekräftigen Namen für das Token ein.
7. Wählen Sie im Abschnitt Bereiche auswählen die Option admin:repo_hook aus.
8. Wählen Sie Generate token (Token erstellen) aus.
9. Kopieren und speichern Sie das persönliche Zugriffstoken. Sie müssen es angeben, wenn Sie eine Amplify-App mit der CLI, CloudFormation, oder der SDKs bereitstellen.

Nachdem die GitHub Amplify-App in Ihrem GitHub Konto installiert ist und Sie ein persönliches Zugriffstoken generiert haben, können Sie eine neue App mit der Amplify-CLI oder dem bereitstellen. CloudFormation SDKs Verwenden Sie das `accessToken` Feld, um das persönliche Zugriffstoken anzugeben, das Sie im vorherigen Verfahren erstellt haben. Weitere Informationen finden Sie CreateAppin der Amplify API-Referenz und AWS::Amplify::Appim AWS CloudFormation Benutzerhandbuch.

Der folgende CLI-Befehl stellt eine neue Amplify-App bereit, die die GitHub App für den Repository-Zugriff verwendet. Ersetzen Sie *myapp-using-githubapphttps://github.com/Myaccount/react-app*, und *MY_TOKEN* durch Ihre eigenen Informationen.

```
aws amplify create-app --name myapp-using-githubapp --repository https://github.com/
Myaccount/react-app --access-token MY_TOKEN
```

Webvorschauen mit der GitHub Amplify App einrichten

Eine Webvorschau stellt jede Pull-Anfrage (PR), die an Ihr GitHub Repository gesendet wird, an eine eindeutige Vorschau-URL bereit. Vorschauen verwenden jetzt die Amplify GitHub App für den

Zugriff auf Ihr GitHub Repo. Anweisungen zur Installation und Autorisierung der GitHub App für Webvorschauen finden Sie unter. Aktivieren Sie Webvorschauen für Pull-Requests

AWS Amplify Hosting-Referenz

Verwenden Sie die Themen in diesem Abschnitt, um detailliertes Referenzmaterial für zu finden. AWS Amplify

Topics

AWS CloudFormation Unterstützung

Verwenden Sie AWS CloudFormation Vorlagen, um Amplify-Ressourcen bereitzustellen und so wiederholbare und zuverlässige Web-App-Bereitstellungen zu ermöglichen. AWS CloudFormation bietet eine gemeinsame Sprache, in der Sie alle Infrastrukturressourcen in Ihrer Cloud-Umgebung beschreiben und bereitstellen können, und vereinfacht den Rollout in mehreren AWS and/or Kontoregionen mit nur wenigen Klicks.

Informationen zu Amplify Hosting finden Sie in der Amplify-Dokumentation CloudFormation. Informationen zu Amplify Studio finden Sie in der Amplify UI CloudFormation Builder-Dokumentation.

AWS Command Line Interface Unterstützung

Verwenden Sie die AWS Command Line Interface , um Amplify-Apps programmgesteuert über die Befehlszeile zu erstellen. Informationen finden Sie in der Dokumentation.AWS CLI

Unterstützung für Ressourcen-Tagging

Sie können das verwenden AWS Command Line Interface , um Amplify-Ressourcen zu taggen. Weitere Informationen finden Sie in der AWS CLI Tag-Resource-Dokumentation.

Hosting-API Amplify

Diese Referenz enthält Beschreibungen der Aktionen und Datentypen für die Amplify Hosting API. Weitere Informationen finden Sie in der Amplify API-Referenzdokumentation.

Dokumenthistorie für AWS Amplify

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der letzten Version von beschrieben. AWS Amplify

- Letzte Aktualisierung der Dokumentation: 8. September 2025

Änderungen	Beschreibung	Date
Aktualisierungen der von Node.js unterstützten Versionen	Die Informationen zu den von Node.js unterstützten Versionen wurden unter Deploying serverseitig gerenderter Anwendungen mit Amplify Hosting aktualisiert.	8. September 2025
Das Kapitel mit den Build-Einstellungen und der Konfiguration wurde aktualisiert	Das Verwaltung der Build-Konfiguration für eine Amplify-Anwendung Kapitel wurde aktualisiert, um die neue Funktion für konfigurierbare Build-Instance-Typen zu beschreiben, mit der Sie einen Instance-Typ auswählen können, der Ihrer Anwendung die benötigten CPU-, Arbeitsspeicher- und Festplattenspeicherressourcen zur Verfügung stellt.	28. Mai 2025
Firewall-Kapitel aktualisiert	Das Firewall-Unterstützung für von Amplify gehostete Websites Kapitel wurde aktualisiert, um die allgemeine Verfügbarkeit (GA) der Integration mit Amplify zu	26. März 2025

Änderungen	Beschreibung	Date
	beschreiben AWS WAF, einschließlich der GA-Funktionalität und der Preisstruktur.	
Neues Kapitel zum Schutz vor Schräglagen	Das Skew-Schutz für Amplify-Bereitstellungen Kapitel zur Beschreibung der Skew-Schutzfunktion wurde hinzugefügt, mit der Probleme mit Versionsverzerrungen zwischen Client und Servern in Amplify-Webanwendungen vermieden werden.	10. März 2025
Das Webhooks-Kapitel wurde aktualisiert	Das Vereinheitlichte Webhooks für Git-Repositorys Thema zur Beschreibung der vereinheitlichten Webhooks-Funktion wurde hinzugefügt, die einen umfassenden Webhook für alle Amplify-Anwendungen verwendet, die mit einem einzigen Git-Repository verknüpft sind.	10. März 2025

Änderungen	Beschreibung	Date
Neues Thema: Hinzufügen einer SSR-Computing-Rolle, um den Zugriff auf AWS Ressourcen zu ermöglichen	Das Hinzufügen einer SSR-Compute-Rolle, um den Zugriff auf Ressourcen zu ermöglichen AWS Thema wurde hinzugefügt, um zu beschreiben, wie eine Amplify SSR Compute-Rolle erstellt und mit einer App verknüpft wird, um dem Amplify Compute-Dienst Zugriff auf Ressourcen zu gewähren. AWS	17. Februar 2025
Neues Kapitel Verwenden AWS WAF zum Schutz Ihrer Amplify-Apps	Es wurde das Firewall-Unterstützung für von Amplify gehostete Websites Kapitel hinzugefügt, in dem die Integration von Amplify mit AWS WAF (in der Vorschau) beschrieben wird, mit der Sie Ihre Webanwendungen mit einer Web-Zugriffskontrollliste (Web-ACL) schützen können.	18. Dezember 2024
Das Thema „Verwaltete Richtlinien“ wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	14. November 2024
Das Thema Amplify-Unterstützung für Next.js wurde aktualisiert	Das Amplify Sie die Unterstützung für Next.js Thema wurde aktualisiert, um die Unterstützung von Amplify für Next.js Version 15 zu beschreiben.	6. November 2024

Änderungen	Beschreibung	Date
Neues Kapitel Bereitstellen einer statischen Website für Amplify von Amazon S3	Das Bereitstellung einer statischen Website für Amplify aus einem Amazon S3 S3-Bucket Kapitel zur Beschreibung der neuen Integration von Amplify mit Amazon S3 wurde hinzugefügt, mit der Sie S3 mit nur wenigen Klicks statische Website-Inhalte hosten können, auf denen gespeichert ist.	16. Oktober 2024
Neues Kapitel „Cache-Konfiguration verwalten“	Es wurde ein Verwaltung der Cache-Konfiguration für eine App Kapitel hinzugefügt, in dem das Standard-Caching-Verhalten von Amplify und die Anwendung verwalteter Cache-Richtlinien auf Inhalte beschrieben werden.	13. August 2024
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	18. Juli 2024
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	31. Mai 2024

Änderungen	Beschreibung	Date
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	17. April 2024
Das Kapitel „Erste Schritte“ wurde aktualisiert	Das Erste Schritte mit der Bereitstellung einer App auf Amplify Hosting Kapitel wurde aktualisiert und verwendet nun eine Beispielanwendung von Next.js im Tutorial.	12. April 2024
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	5. April 2024
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	4. April 2024

Änderungen	Beschreibung	Date
Neues Kapitel zur Fehlerbehebung	Das Fehlerbehebung bei Amplify Hosting Kapitel wurde hinzugefügt, in dem beschrieben wird, wie Probleme behoben werden können, die bei Anwendungen auftreten, die auf Amplify Hosting bereitgestellt werden.	2. April 2024
Neue Unterstützung für benutzerdefinierte SSL/TLS Zertifikate	Dem Eine benutzerdefinierte Domain verbinden Kapitel wurde das Verwenden von Zertifikaten SSL/TLS Thema hinzugefügt, das die Amplify-Unterstützung für benutzerdefinierte SSL/TLS Zertifikate beschreibt, wenn eine App mit einer benutzerdefinierten Domain verbunden wird.	20. Februar 2024
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	2. Januar 2024

Änderungen	Beschreibung	Date
Neue Unterstützung für SSR-Frameworks	Das Bereitstellung serversei tig gerenderter Anwendung en mit Amplify Hosting Thema wurde aktualisiert, um die Amplify-Unterstützung für jedes JavaScript-basierte SSR-Framework mit einem Open-Source-Adapter zu beschreiben.	19. November 2023
Neue Unterstützung für den Start der Bildoptimierungsfu nktion	Das Bildoptimierung für SSR-Apps Thema zur Beschreib ung der integrierten Unterstüt zung für die Bildoptimierung für serverseitig gerenderte Apps wurde hinzugefügt.	19. November 2023
Das Thema „Verwaltete Richtlinien“ wurde aktualisiert	Das AWS verwaltete Richtlini en für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	17. November 2023
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlini en für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	6. November 2023

Änderungen	Beschreibung	Date
Neues Thema für Wildcard-Subdomains	Das Platzhalter-Subdomänen einrichten Thema zur Beschreibung der Unterstützung von Wildcard-Subdomänen in benutzerdefinierten Domänen wurde hinzugefügt.	6. November 2023
Neue von verwaltete Richtlinie	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die neue AmplifyBackendDeployFullAccess AWS verwaltete Richtlinie für Amplify zu beschreiben.	8. Oktober 2023
Neue Unterstützung für Monorepo-Frameworks-Funktionen	Das Konfiguration der Monorepo-Build-Einstellungen Thema wurde aktualisiert, um die Unterstützung für die Bereitstellung von Apps in Monorepos zu beschreiben, die mit npm workspace, pnpm workspace, Yarn workspace, Nx und Turborepo erstellt wurden.	19. Juni 2023
Das Thema „Verwaltete Richtlinien“ wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	01. Juni 2023

Änderungen	Beschreibung	Date
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	24. Februar 2023
Das Kapitel zum serversei tigen Rendern wurde aktualisi ert	Das Bereitstellung serversei tig gerenderter Anwendung en mit Amplify Hosting Kapitel wurde aktualisiert, um die jüngsten Änderungen an der Unterstützung von Amplify für die Versionen 12 und 13 von Next.js zu beschreiben.	17. November 2022
Das Thema „Verwaltete Richtlinien“ wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	30. August 2022
Das Thema verwaltete Richtlinien wurde aktualisiert	Das Ein Backend für eine Anwendung erstellen Thema wurde aktualisiert, um zu beschreiben, wie ein Backend mit Amplify Studio bereitges tellt wird.	23. August 2022

Änderungen	Beschreibung	Date
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	27. April 2022
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	17. April 2022
Einführung einer neuen GitHub App-Funktion	Das Amplify-Zugriff auf Repositorys einrichten GitHub Thema zur Beschreibung der neuen GitHub App zur Autorisierung des Amplify-Zugriffs auf Ihr GitHub Repository wurde hinzugefügt.	5. April 2022
Einführung der neuen Amplify Studio-Funktion	Das Willkommen bei AWS Amplify Hosting Thema wurde aktualisiert, um die Updates für Amplify Studio zu beschreiben, die einen visuellen Designer zum Erstellen von UI-Kompon enten bieten, die Sie mit Ihren Backend-Daten verbinden können.	2. Dezember 2021

Änderungen	Beschreibung	Date
Das Thema „Verwaltete Richtlinien“ wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zur Unterstützung von Amplify Studio zu beschreiben.	2. Dezember 2021
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	8. November 2021
Das Thema verwaltete Richtlinien wurde aktualisiert	Das AWS verwaltete Richtlinien für AWS Amplify Thema wurde aktualisiert, um die jüngsten Änderungen an den AWS verwalteten Richtlinien für Amplify zu beschreiben.	27. September 2021
Neues Thema „Verwaltete Richtlinien“	Das AWS verwaltete Richtlinien für AWS Amplify Thema zur Beschreibung der AWS verwalteten Richtlinien für Amplify und der jüngsten Änderungen an diesen Richtlinien wurde hinzugefügt.	28. Juli 2021

Änderungen	Beschreibung	Date
Das Kapitel zum serversei tigen Rendern wurde aktualisi ert	Das Bereitstellung serversei tig gerenderter Anwendung en mit Amplify Hosting Kapitel wurde aktualisiert, um die neue Unterstützung für Next.js Version 10 zu beschreiben. x. x und Next.js Version 11.	22. Juli 2021
Das Kapitel „Konfiguration der Build-Einstellungen“ wurde aktualisiert	Das Konfiguration der Monorepo-Build-Einstellungen Thema wurde hinzugefügt, um zu beschreiben, wie die Build-Einstellungen und die neue `AMPLIFY_MONOREPO_A PP_ROOT` Umgebungs variable konfiguriert werden, wenn eine Monorepo-App mit Amplify bereitgestellt wird.	20. Juli 2021

Änderungen	Beschreibung	Date
Das Kapitel über Feature-Branch-Bereitstellungen wurde aktualisiert	Es wurde ein Automatische Generierung der Amplify-Konfiguration während der Erstellung (nur Gen 1-Apps) Thema hinzugefügt, das beschreibt, wie die `aws-exports.js` Datei während der Erstellung automatisch generiert wird. Das Bedingte Backend-Builds (nur Apps der Generation 1) Thema zur Beschreibung der Aktivierung bedingter Backend-Builds wurde hinzugefügt. Es wurde ein Verwenden Sie Amplify-Backends für mehrere Apps (nur Gen-1-Apps) Thema hinzugefügt, das beschreibt, wie vorhandene Backends wiederverwendet werden können, wenn Sie eine neue App erstellen, einen neuen Branch mit einer vorhandenen App verbinden oder ein vorhandenes Frontend aktualisieren, sodass es auf eine andere Backend-Umgebung verweist.	30. Juni 2021

Änderungen	Beschreibung	Date
Das Kapitel Sicherheit wurde aktualisiert	Das Datenschutz in Amplify Thema wurde hinzugefügt, in dem beschrieben wird, wie das Modell der gemeinsamen Verantwortung angewendet wird und wie Amplify Verschlüsselung einsetzt, um Ihre Daten im Ruhezustand und bei der Übertragung zu schützen.	3. Juni 2021
Neue Unterstützung für den Start der SSR-Funktion	Das Bereitstellung serversei tig gerenderter Anwendungen mit Amplify Hosting Kapitel zur Beschreibung der Amplify-Unterstützung für Web-Apps, die serverseitiges Rendering (SSR) verwenden und mit Next.js erstellt wurden, wurde hinzugefügt.	18. Mai 2021
Neues Kapitel bezüglich der Sicherheit	Es wurde ein Sicherheit in Amplify Kapitel hinzugefügt, in dem beschrieben wird, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amplify anwenden und wie Sie Amplify konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen.	26. März 2021

Änderungen	Beschreibung	Date
Das Thema benutzerdefinierte Builds wurde aktualisiert	Das Thema Custom Build-Images und Live-Paket-Updates wurde aktualisiert und beschreibt nun, wie ein Custom Build-Image konfiguriert wird, das in Amazon Elastic Container Registry Public gehostet wird.	12. März 2021
Das Thema Überwachung wurde aktualisiert	Das Thema Überwachung wurde aktualisiert und beschreibt nun, wie Sie auf Amazon CloudWatch Metrics-Daten zugreifen und Alarme einrichten können.	2. Februar 2021
Neues Thema zur CloudTrail Protokollierung	Es wurde das AWS CloudTrail Thema Logging Amplify API calls using hinzugefügt, um zu beschreiben, wie alle API-Aktionen für die AWS Amplify Konsolen-API-Referenz und die AWS Amplify Admin-UI-API-Referenz AWS CloudTrail erfasst und protokolliert werden.	2. Februar 2021

Änderungen	Beschreibung	Date
Einführung einer neuen Admin-UI-Funktion	Das Willkommen bei AWS Amplify Hosting Thema wurde aktualisiert, um die neue Admin-Benutzeroberfläche zu beschreiben, die eine visuelle Oberfläche für Frontend-Web- und Mobilentwickler bietet, um App-Backends außerhalb von zu erstellen und zu verwalten. AWS-Managementkonsole	1. Dezember 2020
Start der neuen Funktion im Leistungsmodus	Das Thema App-Leistung verwalten wurde aktualisi ert und beschreibt nun, wie der Leistungsmodus aktiviert werden kann, um eine schnellere Hosting-Leistung zu erzielen.	4. November 2020
Das Thema „Benutzerdefiniert e Header“ wurde aktualisiert	Das Thema Benutzerdefinierte Header wurde aktualisiert, um zu beschreiben, wie benutzerdefinierte Header für eine Amplify-App mithilfe der Konsole oder durch Bearbeite n einer YML-Datei definiert werden.	28. Oktober 2020

Änderungen	Beschreibung	Date
Einführung der neuen Auto-Subdomains-Funktion	Das Thema Automatische Subdomänen für eine benutzerdefinierte Route 53-Domain einrichten wurde hinzugefügt, um zu beschreiben, wie musterbasierte Feature-Branch-Bereitstellungen für eine App verwendet werden, die mit einer benutzerdefinierten Amazon Route 53-Domain verbunden ist. Das Thema Webvorschauzugriff mit Subdomänen wurde hinzugefügt, um zu beschreiben, wie Webvorschauen anhand von Pull-Requests eingerichtet werden, sodass sie mit Subdomänen zugänglich sind.	20. Juni 2020
Neues Thema für Benachrichtigungen	Das Thema Benachrichtigungen wurde hinzugefügt, um zu beschreiben, wie E-Mail-Benachrichtigungen für eine Amplify-App eingerichtet werden, um Stakeholder oder Teammitglieder zu benachrichtigen, wenn ein Build erfolgreich ist oder fehlschlägt.	20. Juni 2020

Änderungen	Beschreibung	Date
Das Thema „Benutzerdefiniert e Domains“ wurde aktualisiert	Das Eine benutzerdefinierte Domain verbinden Thema wurde aktualisiert, um die Verfahren zum Hinzufügen benutzerdefinierter Domains in Amazon Route 53 und Google Domains zu verbesser n. GoDaddy Dieses Update enthält auch neue Informati onen zur Fehlerbehebung bei der Einrichtung benutzerd efinierter Domains.	12. Mai 2020
AWS Amplify Veröffentlichung	In dieser Version wird Amplify eingeführt.	26. November 2018

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.