Guía del usuario

AWS Amplify Hospedaje

AWS Amplify Hospedaje: Guía del usuario

Table of Contents

Bienvenido a AWS Amplify Hosting

Amplify Hosting proporciona un flujo de trabajo basado en Git para alojar aplicaciones web sin servidor full stack con implementación continua. Amplify implementa tu aplicación en la red AWS global de entrega de contenido (CDN). Esta guía del usuario proporciona toda la información necesaria para empezar a usar Amplify Hosting.

Marcos admitidos

Amplify Hosting es compatible con muchos marcos SSR comunes, marcos de aplicaciones de una sola página (SPA) y generadores de sitios estáticos, incluidos los siguientes.

marcos SSR

- Next.js
- Nuxt
- Astro con un adaptador comunitario
- SvelteKit con un adaptador comunitario
- Cualquier marco SSR con un adaptador personalizado

Marcos SPA

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

Generadores de sitios estáticos

- Eleventy
- Gatsby
- Hugo
- Jekyll

- VuePress

Características de Amplify Hosting

Ramificación de características

Administrar entornos de producción y ensayo para su frontend y backend conectando nuevas ramificaciones.

Dominios personalizados

Conecte su aplicación a un dominio personalizado.

Vista previa de una solicitud de extracción

Obtenga una vista previa de los cambios durante las revisiones del código.

End-to-end probando

Mejora la calidad de tu aplicación con end-to-end pruebas.

Ramificaciones protegidas con contraseña

Proteja su aplicación web mediante contraseña para poder trabajar en nuevas características sin hacer que estén accesibles públicamente.

Redireccionamientos y reescrituras

Configure reescrituras y redireccionamientos para mantener las clasificaciones de SEO y dirigir el tráfico en función de las necesidades de su aplicación cliente.

Implementaciones atómicas

Las implementaciones atómicas eliminan las ventanas de mantenimiento al asegurar que la aplicación web se actualice solo después de que se haya completado la implementación. Esto elimina las situaciones en las que los archivos no se cargan correctamente.

Introducción a Amplify Hosting

Para empezar a usar las características de Amplify Hosting, consulte el tutorial Introducción a la implementación de una aplicación en Amplify Hosting. Tras completar el tutorial, sabrás cómo conectar una aplicación web a un repositorio de Git (GitHub, BitBucket GitLab, o AWS CodeCommit) e implementarla en Amplify Hosting con despliegue continuo.

Creación de un backend

AWS Amplify Gen 2 presenta una experiencia de desarrollador TypeScript basada en el código para definir los backends. Para obtener información sobre cómo usar Amplify Gen 2 para crear y conectar un backend a su aplicación, consulte Build & connect backend en Amplify Docs.

Para entender mejor Amplify Gen 2Si su enfoque prioriza el código, consulte el taller Amplify Gen 2 en el AWS sitio web de Workshop Studio. En este tutorial completo, creará una aplicación sin servidor con React y Next.js y aprenderá a usar las bibliotecas de datos y autenticación de Amplify Gen 2, así como la biblioteca de Amplify UI para agregar funcionalidad a la aplicación.

Si busca la documentación para crear backends de una aplicación de primera generación mediante la CLI y Amplify Studio, consulte Build & connect backend en Amplify Docs para Gen 1.

Precios de Amplify Hosting

AWS Amplify es parte de. capa gratuita de AWS Puede empezar de manera gratuita y, posteriormente, seguir con el pago por uso cuando se superen los límites del nivel gratuito. Para obtener información sobre los gastos de Amplify Hosting, consulte Precios de AWS Amplify.

Introducción a la implementación de una aplicación en Amplify Hosting

Para comprender cómo funciona Amplify Hosting, consulte los siguientes tutoriales que lo guiarán a través de la creación e implementación de aplicaciones creadas con marcos SSR comunes compatibles con Amplify.

Tutoriales

- Implementación de una aplicación Next.js en Amplify Hosting
- Implementación de una aplicación Nuxt.js en Amplify Hosting
- Implementación de una aplicación Astro.js en Amplify Hosting
- Implemente una SvelteKit aplicación para Amplify Hosting

Implementación de una aplicación Next.js en Amplify Hosting

En este tutorial, se explica cómo crear e implementar una aplicación Next.js desde un repositorio de Git.

Antes de comenzar este tutorial, complete los siguientes requisitos previos.

Inscríbase en una Cuenta de AWS

Si aún no es AWS cliente, debe crear una Cuenta de AWS siguiendo las instrucciones en línea. Al registrarte, podrás acceder a Amplify y a otros AWS servicios que puedes usar con tu aplicación.

Creación de una aplicación de

Cree una aplicación básica de Next.js para utilizarla en este tutorial, siguiendo las create-next-appinstrucciones de la documentación de Next.js.

Creación de un repositorio de Git

Amplify admite GitHub Bitbucket y. GitLab AWS CodeCommit Envíe la aplicación `create-next-app` a su repositorio de Git.

Paso 1: Conectar un repositorio de Git

En este paso, conecta la aplicación Next.js en un repositorio de Git a Amplify Hosting.

Conexión de una aplicación a un repositorio de Git

1. Abra la consola de Amplify.
2. Si va a implementar su primera aplicación en la región actual, de forma predeterminada empezará desde la página de servicio de AWS Amplify.

   Elija Crear nueva aplicación en la parte superior de la página.
3. En la página Comenzar a crear con Amplify, seleccione el proveedor de repositorios de Git y, a continuación, elija Siguiente.

   En el GitHub caso de los repositorios, Amplify utiliza GitHub la función Aplicaciones para autorizar el acceso de Amplify. Para obtener más información sobre la instalación y la autorización de la GitHub aplicación, consulte. Configuración del acceso de Amplify a los repositorios GitHub

   > **Note**
   >
   > Tras autorizar la consola de Amplify con Bitbucket GitLab, o AWS CodeCommit Amplify obtiene un token de acceso del proveedor del repositorio, pero no lo almacena en los servidores. AWS Amplify obtiene acceso a su repositorio utilizando claves de implementación instaladas solo en un repositorio específico.
4. En la página Añadir ramificación de repositorio, siga estos pasos:
   a. Seleccione el nombre del repositorio que desea conectar.
   b. Seleccione el nombre de la ramificación del repositorio que desea conectar.
   c. Elija Next (Siguiente).

Paso 2: Confirmar la configuración de compilación

Amplify detecta automáticamente la secuencia de comandos de compilación que se va a ejecutar en la ramificación que se va a implementar. En este paso, se revisa y confirma la configuración de compilación.

Confirmación de la configuración de compilación de una aplicación

1. En la página de Configuración de la aplicación, busque la sección Configuración de compilación.

Compruebe que el comando de compilación de Frontend y el directorio de salida de compilación sean correctos. Para esta aplicación Next.js de ejemplo, el directorio de salida de compilación está establecido en `.next`.

2. El procedimiento para agregar un rol de servicio varía en función de si desea crear uno nuevo o usar uno existente.
   - Creación de un nuevo rol:
     - Elija Crear y utilizar un nuevo rol de servicio.
   - Uso de un rol existente:
     a. Elija Usar un rol existente.
     b. En la lista de roles de servicio, seleccione el que desee utilizar.
3. Elija Next (Siguiente).

Paso 3: implementar de la aplicación

En este paso, implementas tu aplicación en la red AWS global de entrega de contenido (CDN).

Guardar e implementar una aplicación

1. En la página de revisión, confirme que los detalles del repositorio y la configuración de la aplicación son correctos.
2. Elija Guardar e implementar. La compilación de frontend suele tardar de 1 a 2 minutos, pero puede variar en función del tamaño de la aplicación.
3. Una vez completada la implementación, podrá ver su aplicación con el enlace en el dominio predeterminado `amplifyapp.com`.

> **Note**
>
> Para aumentar la seguridad de las aplicaciones de Amplify, el dominio amplifyapp.com se ha registrado en la lista de sufijos públicos (PSL). Para una mayor seguridad, le recomendamos que utilice cookies con un prefijo `__Host-` si alguna vez necesita configurar cookies confidenciales en el nombre de dominio predeterminado de las aplicaciones de Amplify. Esta práctica le ayudará a proteger su dominio de los intentos de falsificación de solicitudes entre

sitios (CSRF). Para obtener más información, consulte la página de configuración de cookies en la red de desarrolladores de Mozilla.

Paso 4: Limpieza de recursos (opcional)

Si ya no necesita la aplicación que implementó para el tutorial, puede eliminarla. Este paso le permite asegurarse de que no se le cobre por los recursos que no vaya a utilizar.

Eliminación de una aplicación

1. En Configuración de la aplicación del menú del panel de navegación, elija Configuración general.
2. En la página Configuración general, elija Eliminar aplicación.
3. En la ventana de confirmación, introduzca **`delete`**. A continuación, elija Eliminar aplicación.

Agregar características a su aplicación

Ahora que tiene una aplicación implementada en Amplify, puede explorar algunas de las siguientes características disponibles para su aplicación alojada.

Variables de entorno

Las aplicaciones suelen necesitar información de configuración en el tiempo de ejecución. Estas configuraciones pueden ser los detalles de conexión de la base de datos, las claves de API o los parámetros. Las variables de entorno proporcionan una manera de exponer estas configuraciones en el momento de la compilación. Para obtener más información, consulte Environment variables.

Dominios personalizados

En este tutorial, Amplify aloja su aplicación en el dominio `amplifyapp.com` predeterminado con una URL como `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`. Si conecta su aplicación a un dominio personalizado, los usuarios verán que su aplicación está alojada en una URL personalizada, como `https://www.example.com`. Para obtener más información, consulte Setting up custom domains.

Vista previa de una solicitud de extracción

Las vistas previas de las solicitudes de extracción web ofrecen a los equipos una forma de previsualizar los cambios de las solicitudes de extracción (PRs) antes de fusionar el código con una rama de producción o integración. Para obtener más información, consulte Web previews for pull requests.

Administrar varios entornos

Para saber cómo Amplify trabaja con las ramas de funciones y los GitFlow flujos de trabajo para admitir múltiples implementaciones, consulte Implementaciones de ramas de funciones y flujos de trabajo de equipo.

Implementación de una aplicación Nuxt.js en Amplify Hosting

Utilice las siguientes instrucciones para implementar una aplicación Nuxt.js en Amplify Hosting. Nuxt implementó un adaptador preestablecido mediante el servidor de Nitro. Esto le permite implementar un proyecto de Nuxt sin ninguna configuración adicional.

Implementación de una aplicación Nuxt en Amplify Hosting

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, seleccione Crear nueva aplicación.
3. En la página Comenzar a crear con Amplify, seleccione el proveedor de repositorios de Git y, a continuación, elija Siguiente.
4. En la página Añadir ramificación de repositorio, haga lo siguiente:
   a. Seleccione el nombre del repositorio que desea conectar.
   b. Seleccione el nombre de la ramificación del repositorio que desea conectar.
   c. Elija Next (Siguiente).
5. Si quieres que Amplify pueda enviar registros de aplicaciones a Amazon CloudWatch Logs, debes habilitarlo explícitamente en la consola. Abra la sección Configuración avanzada y, a continuación, seleccione Habilitar registros de aplicaciones de SSR en la sección Implementación de la renderización del servidor (SSR).
6. Elija Next (Siguiente).
7. En la página Revisar, elija Guardar e implementar.

Implementación de una aplicación Astro.js en Amplify Hosting

Utilice las siguientes instrucciones para implementar una aplicación Astro.js en Amplify Hosting. Puede usar una aplicación existente o crear una aplicación inicial mediante uno de los ejemplos oficiales que proporciona Astro. Para crear una aplicación inicial, consulte Use a theme or starter template en la documentación de Astro.

Para implementar un sitio de Astro con SSR en Amplify Hosting, debe agregar un adaptador a su aplicación. No mantenemos un adaptador propiedad de Amplify para el marco de Astro. En este tutorial, se utiliza el adaptador `astro-aws-amplify` que creó un miembro de la comunidad. Este adaptador está disponible en github. com/alexnguyennz/astro-aws-amplify en el sitio web. GitHub AWS no mantiene este adaptador.

Implementación de una aplicación Astro en Amplify Hosting

1. En el equipo local, vaya a la aplicación Astro que desea implementar.
2. Para instalar el adaptador, abra una ventana de terminal y ejecute el siguiente comando. En este ejemplo, se usa el adaptador comunitario disponible en github. com/alexnguyennz/astro-aws-amplify. Puede sustituirlo por *`astro-aws-amplify`* el nombre del adaptador que esté utilizando.

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

3. Abra el archivo `astro.config.mjs` en la carpeta de proyectos de su aplicación Astro. Actualice el archivo para agregar el adaptador. El archivo debe tener un aspecto similar al siguiente.

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

   import sitemap from '@astrojs/sitemap';

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

4. Confirme el cambio y envíe el proyecto a su repositorio de Git.

   Ahora está listo o lista para implementar la aplicación Astro en Amplify.

5. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
6. En la página Todas las aplicaciones, seleccione Crear nueva aplicación.
7. En la página Comenzar a crear con Amplify, seleccione el proveedor de repositorios de Git y, a continuación, elija Siguiente.
8. En la página Añadir ramificación de repositorio, haga lo siguiente:

   a. Seleccione el nombre del repositorio que desea conectar.

   b. Seleccione el nombre de la ramificación del repositorio que desea conectar.

   c. Elija Next (Siguiente).

9. En la página de Configuración de la aplicación, busque la sección Configuración de compilación. En Directorio de salida de compilación, introduzca **`.amplify-hosting`**.
10. También debe actualizar los comandos de compilación del frontend de la aplicación en la especificación de compilación. Para abrir la especificación de compilación, seleccione Editar archivo YML.
11. En el archivo `amplify.yml`, localice la sección de comandos de compilación del frontend. Escriba **`mv node_modules ./.amplify-hosting/compute/default`**

    El archivo de configuración de compilación debe tener el aspecto siguiente.

```
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. Seleccione Guardar.
13. Si quieres que Amplify pueda enviar registros de aplicaciones a Amazon CloudWatch Logs, debes habilitarlo explícitamente en la consola. Abra la sección Configuración avanzada y, a continuación, seleccione Habilitar registros de aplicaciones de SSR en la sección Implementación de la renderización del servidor (SSR).
14. Elija Next (Siguiente).
15. En la página Revisar, elija Guardar e implementar.

Implemente una SvelteKit aplicación para Amplify Hosting

Utilice las siguientes instrucciones para implementar una SvelteKit aplicación en Amplify Hosting. Puede usar su propia aplicación o crear una aplicación de inicio. Para obtener más información, consulte Crear un proyecto en la SvelteKit documentación.

Para implementar una SvelteKit aplicación con SSR en Amplify Hosting, debes agregar un adaptador a tu proyecto. No mantenemos un adaptador propiedad de Amplify para el SvelteKit marco. En este ejemplo, utilizamos el `amplify-adapter` creado por un miembro de la comunidad. El adaptador está disponible en github. com/gzimbron/amplify-adapter en el sitio GitHub web. AWS no mantiene este adaptador.

Para implementar una SvelteKit aplicación en Amplify Hosting

1. En su computadora local, navegue hasta la SvelteKit aplicación que desee implementar.
2. Para instalar el adaptador, abra una ventana de terminal y ejecute el siguiente comando. En este ejemplo, se usa el adaptador comunitario disponible en github. com/gzimbron/amplify-adaptador. Si utiliza un adaptador comunitario diferente, *`amplify-adapter`* sustitúyalo por el nombre del adaptador.

   ```
   npm install amplify-adapter
   ```

3. En la carpeta del proyecto de tu SvelteKit aplicación, abre el `svelte.config.js` archivo. Edita el archivo para usarlo `amplify-adapter` o *`'amplify-adapter'`* sustitúyelo por el nombre de tu adaptador. El archivo debe tener un aspecto similar al siguiente.

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

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


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

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

export default config;
```

4. Confirme el cambio y envíe la aplicación a su repositorio de Git.
5. Ahora está listo para implementar su SvelteKit aplicación en Amplify.

   Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
6. En la página Todas las aplicaciones, seleccione Crear nueva aplicación.
7. En la página Comenzar a crear con Amplify, seleccione el proveedor de repositorios de Git y, a continuación, elija Siguiente.
8. En la página Añadir ramificación de repositorio, haga lo siguiente:

   a. Seleccione el nombre del repositorio que desea conectar.

   b. Seleccione el nombre de la ramificación del repositorio que desea conectar.

   c. Elija Next (Siguiente).
9. En la página de Configuración de la aplicación, busque la sección Configuración de compilación. En Directorio de salida de compilación, introduzca **`build`**.
10. También debe actualizar los comandos de compilación del frontend de la aplicación en la especificación de compilación. Para abrir la especificación de compilación, seleccione Editar archivo YML.

11. En el archivo `amplify.yml`, localice la sección de comandos de compilación del frontend. Introduzca **`- cd build/compute/default/`** y **`- npm i --production`**.

    El archivo de configuración de compilación debe tener el aspecto siguiente.

```
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. Seleccione Guardar.
13. Si quieres que Amplify pueda enviar registros de aplicaciones a Amazon CloudWatch Logs, debes habilitarlo explícitamente en la consola. Abra la sección Configuración avanzada y, a continuación, seleccione Habilitar registros de aplicaciones de SSR en la sección Implementación de la renderización del servidor (SSR).
14. Elija Next (Siguiente).
15. En la página Revisar, elija Guardar e implementar.

Implementación de aplicaciones renderizadas del servidor con Amplify Hosting

Puede utilizarla AWS Amplify para implementar y alojar aplicaciones web que utilizan la renderización del lado del servidor (SSR). Amplify Hosting detecta automáticamente las aplicaciones creadas con el marco Next.js y no es necesario realizar ninguna configuración manual en la AWS Management Console.

Amplify también es compatible con cualquier marco SSR basado en JavaScript con un adaptador de compilación de código abierto que transforme la salida de la compilación de una aplicación en la estructura de directorios que Amplify Hosting espera. Por ejemplo, puede implementar aplicaciones creadas con Nuxt, Astro y los SvelteKit marcos instalando los adaptadores disponibles.

Los usuarios avanzados pueden usar la especificación de implementación para crear un adaptador de compilación o configurar un script posterior a la compilación.

Puede implementar los siguientes marcos para Amplify Hosting con una configuración mínima.

Next.js

- Amplify es compatible con las aplicaciones Next.js 15 sin necesidad de un adaptador. Para empezar, consulte Compatibilidad de Amplify con Next.js.

Nuxt.js

- Amplify admite las implementaciones de aplicaciones de Nuxt.js con un adaptador preestablecido. Para empezar, consulte Compatibilidad de Amplify con Nuxt.js.

Astro.js

- Amplify admite las implementaciones de aplicaciones de Astro.js con un adaptador preestablecido. Para empezar, consulte Compatibilidad de Amplify con Astro.js.

SvelteKit

- Amplify admite la implementación de SvelteKit aplicaciones con un adaptador comunitario. Para empezar, consulte Amplify el soporte para SvelteKit.

Adaptadores de código abierto

- Uso de un adaptador de código abierto: para obtener instrucciones sobre el uso de cualquier adaptador que no esté en la lista anterior, consulte Uso de adaptadores de código abierto para cualquier marco SSR.

- Creación de un adaptador de marcos: los autores de marcos que deseen integrar las características que proporciona un marco pueden usar la especificación de implementación de Amplify Hosting para configurar la salida de la compilación para que se ajuste a la estructura que Amplify espera. Para obtener más información, consulte Using the Amplify Hosting deployment specification to configure build output.
- Configuración de un script a posterior a la compilación: puede usar la especificación de implementación de Amplify Hosting para manipular la salida de la compilación según sea necesario para situaciones específicas. Para obtener más información, consulte Using the Amplify Hosting deployment specification to configure build output. Para ver un ejemplo, consulta Implementación de un servidor Express mediante el manifiesto de implementación.

Temas

Compatibilidad de Amplify con Next.js

Amplify admite la implementación y el alojamiento de aplicaciones web renderizadas en el servidor (SSR) creadas con Next.js. Next.js es un marco de React para desarrollar SPAs con él JavaScript. Puede implementar aplicaciones creadas con versiones de Next.js hasta Next.js 15, con funciones como la optimización de imágenes y el middleware.

Los desarrolladores pueden utilizar Next.js para combinar la generación de sitios estáticos (SSG) y SSR en un solo proyecto. Las páginas SSG se renderizan previamente en el momento de la compilación y las páginas SSR se renderizan previamente en el momento de la solicitud.

La renderización previa puede mejorar el rendimiento y la optimización de los motores de búsqueda. Como Next.js renderiza previamente todas las páginas en el servidor, el contenido HTML de cada

página estará preparado cuando llegue al navegador del cliente. Este contenido también se puede cargar más rápido. Los tiempos de carga más rápidos mejoran la experiencia del usuario final con un sitio web y tienen un impacto positivo en la clasificación SEO del sitio. La renderización previa también mejora el SEO al permitir que los bots de los motores de búsqueda encuentren y rastreen fácilmente el contenido HTML de un sitio web.

Next.js ofrece un servicio de asistencia analítico que se integra para medir varias métricas de rendimiento, como el tiempo hasta el primer byte (TTFB) y el primer contenido de pintura (FCP). Para obtener más información acerca de Next.js, consulte Getting started en el sitio web de Next.js.

Compatibilidad de las características de Next.js

Amplify Hosting Compute administra completamente la renderización del lado del servidor (SSR) para las aplicaciones creadas con las versiones 12 a 15 de Next.js.

Si implementaste una aplicación de Next.js en Amplify antes del lanzamiento de Amplify Hosting Compute en noviembre de 2022, tu aplicación utiliza el proveedor de SSR anterior de Amplify, Classic (solo Next.js 11). El procesamiento de Amplify Hosting no admite aplicaciones creadas con la versión 11 o anteriores de Next.js. Le recomendamos encarecidamente que migre sus aplicaciones de Next.js 11 al proveedor de SSR gestionado por el procesamiento de Amplify Hosting.

La siguiente lista describe las características específicas que admite el proveedor SSR de procesamiento de Amplify Hosting.

Características admitidas

- Páginas renderizadas del servidor (SSR)
- Páginas estáticas
- Rutas de la API
- Rutas dinámicas
- Captura de todas las rutas
- SSG (generación estática)
- Regeneración estática incremental (ISR)
- Enrutamiento de subrutas internacionalizado (i18n)
- Enrutamiento de dominio internacionalizado (i18n)
- Detección automática de configuración regional internacionalizada (i18n)
- Middleware

- Variables de entorno
- Optimización de imágenes
- Directorio de aplicaciones Next.js 13

Características no admitidas

- Rutas de la API de Edge (no se admite el middleware de Edge)
- Regeneración estática incremental (ISR) bajo demanda
- Transmisión de Next.js
- Ejecución de middleware en activos estáticos e imágenes optimizadas
- Ejecutar código después de una respuesta con `unstable_after` (función experimental lanzada con Next.js 15)

Imágenes de Next.js

El tamaño máximo de salida de una imagen no puede superar los 4,3 MB. Puede almacenar un archivo de imagen más grande en algún lugar y utilizar el componente Image de Next.js para cambiarlo de tamaño y optimizarlo a un formato Webp o AVIF y, a continuación, distribuirlo como un tamaño más pequeño.

Tenga en cuenta que en la documentación de Next.js se recomienda instalar el módulo de procesamiento de imágenes de Sharp para permitir que la optimización de imágenes funcione correctamente en producción. Sin embargo, esto no es necesario para las implementaciones de Amplify. Amplify implementa Sharp automáticamente en su lugar.

Implementación de una aplicación SSR de Next.js en Amplify

De forma predeterminada, Amplify implementa nuevas aplicaciones SSR mediante el servicio de cómputo de Amplify Hosting compatible con las versiones 12 a 15 de Next.js. La computación de Amplify Hosting administra completamente los recursos necesarios para implementar una aplicación de SSR. Las aplicaciones SSR de su cuenta de Amplify que ha implementado antes del 17 de noviembre de 2022, utilizan el proveedor SSR clásico (solo Next.js 11).

Le recomendamos encarecidamente que migre las aplicaciones que utilizan SSR clásico (solo Next.js 11) al proveedor de SSR de procesamiento de Amplify Hosting. Amplify no realiza migraciones automáticas en su lugar. Debe migrar la aplicación manualmente y, a continuación,

iniciar una nueva compilación para completar la actualización. Para obtener instrucciones, consulte Migración de una aplicación SSR de Next.js 11 al procesamiento de Amplify Hosting.

Utilice las siguientes instrucciones para implementar una nueva aplicación SSR de Next.js.

Para implementar una aplicación SSR en Amplify mediante el proveedor de SSR de procesamiento de Amplify Hosting

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, seleccione Crear nueva aplicación.
3. En la página Comenzar a crear con Amplify, seleccione el proveedor de repositorios de Git y, a continuación, elija Siguiente.
4. En la página Añadir ramificación de repositorio, haga lo siguiente:
   a. En la lista de Repositorios actualizados recientemente, seleccione el nombre del repositorio que desea conectar.
   b. En la lista de Ramificaciones, seleccione el nombre de la ramificación del repositorio que desea conectar.
   c. Elija Siguiente.
5. La aplicación requiere un rol de servicio de IAM que Amplify asume cuando llama a otros servicios en su nombre. Puede permitir que el procesamiento de Amplify Hosting cree automáticamente un rol de servicio en su lugar, o puede especificar un rol que haya creado usted.
   - Para permitir que Amplify cree automáticamente un rol y lo asocie a su aplicación:
     - Elija Crear y utilizar un nuevo rol de servicio.
   - Para adjuntar un rol de servicio que haya creado anteriormente:
     a. Elija Utilizar un rol de servicio existente.
     b. Seleccione el rol que desea utilizar de la lista.
6. Elija Next (Siguiente).
7. En la página Revisar, elija Guardar e implementar.

Configuración del archivo Package.json

Al implementar una aplicación Next.js, Amplify inspecciona el script de compilación de la aplicación en el archivo `package.json` para determinar el tipo de aplicación.

A continuación, se muestra un ejemplo del script de compilación de una aplicación Next.js. El script de compilación `"next build"` indica que la aplicación es compatible con las páginas SSG y SSR. Este script de compilación también se usa para aplicaciones de Next.js 14 o versiones posteriores exclusivas para SSG.

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

A continuación, se muestra un ejemplo del script de compilación de una aplicación SSG de Next.js 13 o versiones anteriores. El script de compilación `"next build && next export"` indica que la aplicación solo admite páginas SSG.

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

Configuración de compilación de Amplify para una aplicación SSR de Next.js

Después de inspeccionar el archivo `package.json` de la aplicación, Amplify comprueba la configuración de compilación de la aplicación. Puede guardar la configuración de compilación en la consola de Amplify o en un archivo `amplify.yml` en la raíz de su repositorio. Para obtener más información, consulte Ajuste de la configuración de compilación de una aplicación.

Si Amplify detecta que está implementando una aplicación SSR de Next.js y no hay ningún archivo `amplify.yml`, genera una especificación de compilación para la aplicación y configura `baseDirectory` en `.next`. Si está implementando una aplicación en la que hay un archivo `amplify.yml`, la configuración de compilación del archivo anula cualquier configuración de compilación de la consola. Por lo tanto, debe configurar manualmente `baseDirectory` en `.next` en el archivo.

A continuación se muestra un ejemplo de la configuración de compilación de una aplicación donde `baseDirectory` se configura en `.next`. Esto indica que los artefactos de compilación son para una aplicación de Next.js que admite páginas SSG y SSR.

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

Configuración de compilación de Amplify para una aplicación SSG de Next.js 13 o anterior

Si Amplify detecta que está implementando una aplicación SSG de Next.js 13 o versiones anteriores, genera una especificación de compilación para la aplicación y configura `baseDirectory` en `out`. Si está implementando una aplicación en la que hay un archivo `amplify.yml`, debe configurar manualmente `baseDirectory` en `out` en el archivo. El directorio `out` es la carpeta predeterminada que crea Next.js para almacenar los activos estáticos exportados. Al configurar las especificaciones de compilación de la aplicación, cambie el nombre de la carpeta `baseDirectory` para que coincida con la configuración de la aplicación.

A continuación, se muestra un ejemplo de la configuración de compilación de una aplicación en la que `baseDirectory` se configura en `out` para indicar que los artefactos de compilación son para una aplicación Next.js 13 o anterior que solo admite páginas SSG.

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

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

Amplify la configuración de compilación de una aplicación SSG de Next.js 1.4 o posterior

En la versión 14 de Next.js, el comando `next export` quedó obsoleto y se sustituyó por `output: 'export'` en el archivo `next.config.js` para permitir las exportaciones estáticas. Si implementa una aplicación SSG de Next.js 14 en la consola, Amplify genera una especificación de compilación para la aplicación y configura `baseDirectory` en `.next`. Si está implementando una aplicación en la que hay un archivo `amplify.yml`, debe configurar manualmente `baseDirectory` en `.next` en el archivo. Esta es la misma configuración `baseDirectory` que Amplify usa para las aplicaciones `WEB_COMPUTE` de Next.js que admiten páginas SSG y SSR.

A continuación se muestra un ejemplo de la configuración de compilación de una aplicación SSG de Next.js 14 con el valor de `baseDirectory` en `.next`.

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

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

Migración de una aplicación SSR de Next.js 11 al procesamiento de Amplify Hosting

Al implementar una nueva aplicación de Next.js, Amplify utiliza de forma predeterminada la versión compatible más reciente de Next.js. Actualmente, el proveedor de SSR de cómputo Amplify Hosting es compatible con la versión 15 de Next.js.

La consola Amplify detecta las aplicaciones de tu cuenta que se implementaron antes de la versión de noviembre de 2022 del servicio de cómputo Amplify Hosting y es totalmente compatible con las versiones 12 a 15 de Next.js. La consola muestra un banner informativo que identifica las aplicaciones con ramificaciones que se han implementado con el anterior proveedor clásico de SSR de Amplify (solo Next.js 11). Se recomienda que migre sus aplicaciones al proveedor SSR de procesamiento de Amplify Hosting.

Si estás actualizando la aplicación Next.js 11 alojada a Next.js 12 o una versión posterior, es posible que se produzca un `"target" property is no longer supported` error cuando se active una implementación. En este caso, debe migrar a Amplify Hosting Compute.

Debe migrar manualmente la aplicación y todas sus ramificaciones de producción al mismo tiempo. Una aplicación no puede contener las ramificaciones clásicas (solo Next.js 11) y de Next.js 12 o posteriores.

Siga las siguientes instrucciones para migrar una aplicación al proveedor SSR de procesamiento de Amplify Hosting.

Para migrar una aplicación al proveedor de SSR de procesamiento de Amplify Hosting

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación Next.js que desea migrar.

   > **Note**
   >
   > Antes de migrar una aplicación a la consola de Amplify, debe primero actualizar el archivo package.json de la aplicación para utilizar la versión 12 o posterior de Next.js.

3. En el panel de navegación, elija Configuración de la aplicación y General.

4. En la página de inicio de la aplicación, la consola muestra un banner si la aplicación tiene ramificaciones implementadas con el proveedor SSR clásico (solo para Next.js 11). En el banner, elija Migrar.
5. En la ventana de confirmación de migración, elija las tres sentencias y elija Migrar.
6. Amplify compilará y volverá a implementar su aplicación para completar la migración.

Reversión de una migración de SSR

Al implementar una aplicación de Next.js, Amplify Hosting detecta la configuración de la aplicación y establece el valor de la plataforma interna de la aplicación. Existen tres valores de plataforma válidos. Una aplicación SSG se configura en el valor de la plataforma `WEB`. Una aplicación SSR que utilice la versión 11 de Next.js se configura en el valor de la plataforma `WEB_DYNAMIC`. Una aplicación SSR de Next.js 12 o posterior se configura en el valor de la plataforma `WEB_COMPUTE`.

Al migrar una aplicación siguiendo las instrucciones de la sección anterior, Amplify cambia el valor de la plataforma de la aplicación de `WEB_DYNAMIC` a `WEB_COMPUTE`. Una vez completada la migración al procesamiento de Amplify Hosting, no puede revertir la migración en la consola. Para revertir la migración, debe utilizar AWS Command Line Interface para cambiar la plataforma de la aplicación a `WEB_DYNAMIC`. Abra una ventana de terminal e introduzca el siguiente comando para actualizar el ID y la región de la aplicación con su información exclusiva.

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

Incorporación de la funcionalidad SSR a una aplicación Next.js estática

Puede añadir la funcionalidad SSR a una aplicación Next.js estática (SSG) existente implementada con Amplify. Antes de iniciar el proceso de conversión de la aplicación SSG a SSR, actualice la aplicación para que utilice la versión 12 de Next.js, o posterior, y añada la funcionalidad SSR. A continuación, tendrá que realizar los siguientes pasos.

1. Usa AWS Command Line Interface para cambiar el tipo de plataforma de la aplicación.
2. Añada un rol de servicio a la aplicación.
3. Actualice el directorio de salida en la configuración de compilación de la aplicación.
4. Actualice el archivo `package.json` de la aplicación para indicar que la aplicación utiliza SSR.

Actualización de la plataforma

Existen tres valores válidos para el tipo de plataforma. Una aplicación SSG se configura para el tipo de plataforma `WEB`. Una aplicación SSR que utilice la versión 11 de Next.js se configura en el tipo de plataforma `WEB_DYNAMIC`. En las aplicaciones implementadas en Next.js 12 mediante la SSR administrada por la computación de Amplify Hosting, el tipo de plataforma se establece en `WEB_COMPUTE`.

En el momento en que implementó su aplicación como una aplicación SSG, Amplify configuró el tipo de plataforma en `WEB`. Usa AWS CLI para cambiar la plataforma a la que va tu aplicación`WEB_COMPUTE`. Abra una ventana de terminal e introduzca el siguiente comando, actualizando el texto en rojo con su ID de aplicación y región únicos.

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

Adición de un rol de servicio

Una función de servicio es la función AWS Identity and Access Management (IAM) que Amplify asume cuando llama a otros servicios en su nombre. Siga estos pasos para añadir un rol de servicio a una aplicación SSG que ya se haya implementado con Amplify.

Para añadir un rol de servicio

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Si aún no ha creado un rol de servicio en su cuenta de Amplify, consulte Incorporación de un rol de servicio para completar este paso previo.
3. Elija la aplicación estática de Next.js a la que desea añadir un rol de servicio.
4. En el panel de navegación, elija Configuración de la aplicación y General.
5. En la página Detalles de la aplicación, elija Editar
6. En Rol de servicio, elija el nombre de un rol de servicio existente o el nombre del rol de servicio que ha creado en el paso 2.
7. Seleccione Guardar.

Actualización de la configuración de compilación

Antes de volver a implementar su aplicación con la funcionalidad SSR, debe actualizar la configuración de compilación de la aplicación para configurar el directorio de salida en `.next`. Puede

editar la configuración de compilación en la consola de Amplify o en un archivo `amplify.yml` almacenado en su repositorio. Para obtener más información, consulte Ajuste de la configuración de compilación de una aplicación.

A continuación se muestra un ejemplo de la configuración de compilación de una aplicación donde `baseDirectory` se configura en `.next`.

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

Actualización del archivo package.json

Después de añadir un rol de servicio y actualizar la configuración de compilación, actualice el archivo `package.json` de la aplicación. Como en el siguiente ejemplo, configure el script de compilación en `"next build"` para indicar que la aplicación Next.js es compatible con las páginas SSG y SSR.

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

Amplify detecta el cambio en el archivo `package.json` de su repositorio y vuelve a implementar la aplicación con la funcionalidad SSR.

Conversión de las variables de entorno en accesibles para los tiempos de ejecución del servidor

Amplify Hosting permite añadir variables de entorno a las compilaciones de su aplicación al configurarlas en la configuración del proyecto de la consola de Amplify. Sin embargo, un componente del servidor Next.js no tiene acceso a esas variables de entorno de forma predeterminada. Este comportamiento tiene como objetivo proteger cualquier secreto almacenado en las variables de entorno que utilice su aplicación durante la fase de compilación.

Para que Next.js pueda acceder a variables de entorno específicas, puede modificar el archivo de especificación de compilación de Amplify para configurarlas en los archivos de entorno que reconoce Next.js. Esto permite a Amplify cargar estas variables de entorno antes de compilar la aplicación. El siguiente ejemplo de especificación de compilación muestra cómo añadir variables de entorno en la sección de comandos de compilación.

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

En este ejemplo, la sección de comandos de compilación incluye dos comandos que escriben variables de entorno en el archivo `.env.production` antes de que se ejecute la compilación de la aplicación. Amplify Hosting permite que su aplicación acceda a estas variables cuando la aplicación recibe tráfico.

La siguiente línea de la sección de comandos de compilación del ejemplo anterior muestra cómo tomar una variable específica del entorno de compilación y añadirla al archivo `.env.production`.

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

Si las variables existen en su entorno de compilación, el archivo `.env.production` contendrá las siguientes variables de entorno.

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

La siguiente línea de la sección de comandos de compilación del ejemplo anterior muestra cómo añadir una variable de entorno con un prefijo específico al archivo `.env.production`. En este ejemplo, se añaden todas las variables con el prefijo `NEXT_PUBLIC_`.

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

Si existen varias variables con el prefijo `NEXT_PUBLIC_` en el entorno de compilación, el archivo `.env.production` tendrá un aspecto similar al siguiente.

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

Variables de entorno SSR para monorepos

Si implementa una aplicación SSR en un monorepo y quiere que Next.js pueda acceder a variables de entorno específicas, tiene que anteponer la raíz de la aplicación en el archivo `.env.production`. El siguiente ejemplo de especificación de compilación de una aplicación Next.js dentro de un monorepo Nx muestra cómo añadir variables de entorno en la sección de comandos de compilación.

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

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

Las siguientes líneas de la sección de comandos de compilación del ejemplo anterior muestra cómo tomar variables específicas del entorno de compilación y agregarlas al archivo `.env.production` de una aplicación en un monorepo con la raíz de aplicación `apps/app`.

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

Implementación de una aplicación de Next.js en un monorepo

Amplify admite aplicaciones en monorepos genéricos, así como aplicaciones en monorepos creadas con npm workspace, pnpm workspace, Yarn workspace, Nx y Turborepo. Al implementar su aplicación, Amplify detecta automáticamente el marco de compilación de monorepo que está utilizando. Amplify aplica automáticamente la configuración de compilación a las aplicaciones en un npm workspace, Yarn workspace o Nx. Las aplicaciones Turborepo y pnpm requieren una configuración adicional. Para obtener más información, consulte Modificar la configuración de compilación de monorepo.

Para ver un ejemplo detallado de Nx, consulte la publicación del blog Compartir código entre aplicaciones de Next.js con Nx en AWS Amplify Hosting.

Compatibilidad de Amplify con Nuxt.js

Nuxt es un marco para crear aplicaciones web full stack con Vue.js.

Adaptador

Puede implementar una aplicación Nuxt.js en Amplify mediante un adaptador preestablecido sin necesidad de configuración. Para obtener más información sobre este adaptador, consulte la documentación de Nuxt.

Tutorial

Para obtener información sobre cómo implementar una aplicación Nuxt.js en Amplify, consulte Implementación de una aplicación Nuxt.js en Amplify Hosting.

Demostración

Para ver una demostración en vídeo, consulte Nuxt Hosting With ZERO Configuration In Minutes (With AWS) on YouTube.

Compatibilidad de Amplify con Astro.js

Astro es un marco web para crear aplicaciones web basadas en contenido.

Adaptador

Puede implementar una aplicación Astro.js en Amplify mediante un adaptador comunitario. No mantenemos un adaptador propiedad de Amplify para el marco de Astro. Sin embargo, hay un adaptador disponible en github. com/alexnguyennz/astro-aws-amplify en el sitio web. GitHub Este adaptador lo creó un miembro de la comunidad y no lo mantiene AWS.

Tutorial

Para obtener información sobre cómo implementar una aplicación Astro en Amplify, consulte Implementación de una aplicación Astro.js en Amplify Hosting.

Demostración

Para ver una demostración en vídeo, consulte Cómo implementar un sitio web de Astro AWS en el YouTube canal Amazon Web Services.

Amplify el soporte para SvelteKit

SvelteKit es un marco para crear aplicaciones web completas con Svelte.

Adaptador

Puede implementar una SvelteKit aplicación en Amplify mediante un adaptador comunitario. No mantenemos un adaptador propiedad de Amplify para el SvelteKit marco. Sin embargo, hay un adaptador disponible en github. com/gzimbron/amplify-adapter en el sitio GitHub web. Este adaptador lo creó un miembro de la comunidad y no lo mantiene AWS.

Tutorial

Para obtener información sobre cómo implementar una SvelteKit aplicación en Amplify, consulte. Implemente una SvelteKit aplicación para Amplify Hosting

Demostración

Para ver una demostración en vídeo, consulte Cómo implementar un SvelteKit sitio web (con API) AWS en el YouTube canal Amazon Web Services.

Implementación de una aplicación SSR en Amplify

Puede usar estas instrucciones para implementar una aplicación creada con cualquier marco con una agrupación de implementación que se ajuste a la salida de compilación que Amplify espera. Si va a implementar una aplicación de Next.js, no se necesita ningún adaptador.

Si va a implementar una aplicación de SSR que usa un adaptador de marcos, primero debe instalar y configurar el adaptador. Para obtener instrucciones, consulte Uso de adaptadores de código abierto para cualquier marco SSR.

Implementación de una aplicación de SSR en Amplify Hosting

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, seleccione Crear nueva aplicación.
3. En la página Comenzar a crear con Amplify, seleccione el proveedor de repositorios de Git y, a continuación, elija Siguiente.
4. En la página Añadir ramificación de repositorio, siga estos pasos:
   a. Seleccione el nombre del repositorio que desea conectar.
   b. Seleccione el nombre de la ramificación del repositorio que desea conectar.
   c. Elija Next (Siguiente).

5. En la página Configuración de la aplicación, Amplify detecta automáticamente las aplicaciones SSR de Next.js.

   Si va a implementar una aplicación SSR que utiliza un adaptador para otro marco, debe habilitar Amazon CloudWatch Logs de forma explícita. Abra la sección Configuración avanzada y, a continuación, seleccione Habilitar registros de aplicaciones de SSR en la sección Implementación de la renderización del servidor (SSR).
6. La aplicación requiere un rol de servicio de IAM que Amplify asume para entregar los registros en su Cuenta de AWS.

   El procedimiento para agregar un rol de servicio varía en función de si desea crear uno nuevo o usar uno existente.

   - Creación de un nuevo rol:
     - Elija Crear y utilizar un nuevo rol de servicio.
   - Uso de un rol existente:
     a. Elija Usar un rol existente.
     b. En la lista de roles de servicio, seleccione el que desee utilizar.
7. Elija Next (Siguiente).
8. En la página Revisar, elija Guardar e implementar.

Características admitidas por SSR

En esta sección se proporciona información relativa a la compatibilidad de Amplify con las características de SSR.

Amplify ofrece compatibilidad con la versión de Node.js para que coincida con la versión que se utilizó para crear la aplicación.

Amplify proporciona una característica de optimización de imágenes integrada que es compatible con todas las aplicaciones SSR. Si no quiere usar la característica de optimización de imágenes predeterminada, puede implementar un cargador de optimización de imágenes personalizado.

Temas

- Amazon CloudWatch Logs para aplicaciones SSR
- Compatibilidad de Amplify con SSR de Next.js 11

Compatibilidad de las versiones de Node.js con las aplicaciones de Next.js

Cuando Amplify crea e implementa una aplicación de cómputo de Next.js, usa la Node.js versión en tiempo de ejecución que coincide con la versión principal de Node.js que se utilizó para crear la aplicación.

Puede especificar el Node.js versión para usar en la función de anulación de paquetes Live de la consola Amplify. Para obtener más información sobre cómo configurar las actualizaciones de paquetes en directo, consulte Uso de versiones específicas de paquetes y dependencias en la imagen de compilación. También puede especificar el Node.js versión mediante otros mecanismos, como nvm comandos. Si no especifica ninguna versión, Amplify usará de forma predeterminada la versión actual que utiliza el contenedor de compilación de Amplify.

Optimización de imágenes para aplicaciones de SSR

Amplify Hosting proporciona una característica de optimización de imágenes integrada que es compatible con todas las aplicaciones de SSR. Con la optimización de imágenes de Amplify, puede ofrecer imágenes de alta calidad en el formato, la dimensión y la resolución correctos para el dispositivo que accede a ellas y, al mismo tiempo, mantener el tamaño de archivo más pequeño posible.

Actualmente, puede utilizar el componente Image de Next.js para optimizar las imágenes bajo demanda o puede implementar un cargador de imágenes personalizado. Si utiliza Next.js 13 o una versión posterior, no necesita realizar ninguna otra acción para utilizar la característica de optimización de imágenes de Amplify. Si va a implementar un cargador personalizado, consulte el siguiente tema Uso de un cargador de imágenes personalizado.

Uso de un cargador de imágenes personalizado

Si utiliza un cargador de imágenes personalizado, Amplify detecta el cargador en el archivo `next.config.js` de la aplicación y no utiliza la característica de optimización de imágenes integrada. Para obtener más información sobre los cargadores personalizados compatibles con Next.js, consulte la documentación sobre las imágenes de Next.js.

Amazon CloudWatch Logs para aplicaciones SSR

Amplify envía información sobre tu tiempo de ejecución de SSR a Amazon CloudWatch Logs en tu. Cuenta de AWS Al implementar una aplicación SSR, la aplicación requiere un rol de servicio de IAM que Amplify asume cuando llama a otros servicios en su nombre. Puede permitir que el procesamiento de Amplify Hosting cree automáticamente un rol de servicio en su lugar, o puede especificar un rol que haya creado usted.

Si decides permitir que Amplify cree un rol de IAM para ti, el rol ya tendrá los permisos para crear registros. CloudWatch Si creas tu propia función de IAM, tendrás que añadir los siguientes permisos a tu política para permitir que Amplify acceda a Amazon CloudWatch Logs.

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

Para obtener más información acerca de los roles de servicio, consulte Añadir un rol de servicio con permisos para implementar recursos de backend.

Compatibilidad de Amplify con SSR de Next.js 11

Si ha implementado una aplicación de Next.js en Amplify antes del lanzamiento del procesamiento de Amplify Hosting el 17 de noviembre de 2022, su aplicación utiliza el proveedor clásico de SSR anterior de Amplify (solo Next.js 11). La documentación de esta sección se aplica únicamente a las aplicaciones implementadas con el proveedor clásico SSR (solo en Next.js 11).

> **Note**
>
> Le recomendamos encarecidamente que migre sus aplicaciones de Next.js 11 al proveedor de SSR gestionado por el procesamiento de Amplify Hosting. Para obtener más información, consulte Migración de una aplicación SSR de Next.js 11 al procesamiento de Amplify Hosting.

La siguiente lista describe las características específicas que admite el proveedor clásico de SSR de Amplify (solo Next.js 11).

Características admitidas

- Páginas renderizadas del servidor (SSR)
- Páginas estáticas
- Rutas de la API
- Rutas dinámicas
- Captura de todas las rutas
- SSG (generación estática)
- Regeneración estática incremental (ISR)
- Enrutamiento de subrutas internacionalizado (i18n)
- Variables de entorno

Características no admitidas

- Optimización de imágenes
- Regeneración estática incremental (ISR) bajo demanda
- Enrutamiento de dominio internacionalizado (i18n)
- Detección automática de configuración regional internacionalizada (i18n)
- Middleware
- Middleware de Edge
- Rutas de la API de Edge

Precios de las aplicaciones SSR de Next.js 11

Al implementar tu aplicación SSR de Next.js 11, Amplify crea recursos de backend adicionales en AWS tu cuenta, que incluyen:

- Un bucket de Amazon Simple Storage Service (Amazon S3) que almacena los recursos de los activos estáticos de la aplicación. Para obtener información acerca de los precios de Amazon S3, consulte los precios de Amazon S3.
- Una CloudFront distribución de Amazon para ofrecer la aplicación. Para obtener información sobre CloudFront los cargos, consulta los CloudFront precios de Amazon.
- Cuatro funciones de Lambda @Edge para personalizar el contenido que CloudFront se entrega.

AWS Identity and Access Management permisos para las aplicaciones SSR de Next.js 1.1

Amplify requiere permisos AWS Identity and Access Management (IAM) para implementar una aplicación SSR. Para las aplicaciones SSR, Amplify despliega recursos como un bucket de Amazon S3, una distribución, CloudFront Lambda@Edge funciones, una cola de Amazon SQS (si se utiliza ISR) y funciones de IAM. Si no cuenta con los permisos mínimos requeridos, recibirá un `Access Denied` mensaje de error cuando intente implementar la aplicación SSR. Para proporcionar a Amplify los permisos necesarios, debe especificar un rol de servicio.

Para crear un rol de servicio de IAM que asuma Amplify cuando llama a otros servicios en su nombre, consulte Añadir un rol de servicio con permisos para implementar recursos de backend. Estas instrucciones muestran cómo crear un rol que asocie la política `AdministratorAccess-Amplify` gestionada.

La política `AdministratorAccess-Amplify` gestionada proporciona acceso a varios AWS servicios, incluidas las acciones de IAM, y debe considerarse tan eficaz como la política. `AdministratorAccess` Esta política proporciona más permisos de los necesarios para implementar la aplicación SSR.

Se recomienda seguir la práctica recomendada de concesión de privilegios mínimos y reducir los permisos otorgados al rol de servicio. En lugar de conceder permisos de acceso de administrador para su rol de servicio, puede crear su propia política de IAM gestionada por el cliente que conceda únicamente los permisos necesarios para implementar su aplicación SSR. Para obtener instrucciones sobre cómo crear una política administrada por el cliente, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Si crea su propia política, consulte la siguiente lista de permisos mínimos necesarios para implementar una aplicación SSR.

```
acm:DescribeCertificate
acm:DescribeCertificate
acm:ListCertificates
acm:RequestCertificate
cloudfront:CreateCloudFrontOriginAccessIdentity
cloudfront:CreateDistribution
cloudfront:CreateInvalidation
cloudfront:GetDistribution
cloudfront:GetDistributionConfig
cloudfront:ListCloudFrontOriginAccessIdentities
```

```
cloudfront:ListDistributions
cloudfront:ListDistributionsByLambdaFunction
cloudfront:ListDistributionsByWebACLId
cloudfront:ListFieldLevelEncryptionConfigs
cloudfront:ListFieldLevelEncryptionProfiles
cloudfront:ListInvalidations
cloudfront:ListPublicKeys
cloudfront:ListStreamingDistributions
cloudfront:UpdateDistribution
cloudfront:TagResource
cloudfront:UntagResource
cloudfront:ListTagsForResource
iam:AttachRolePolicy
iam:CreateRole
iam:CreateServiceLinkedRole
iam:GetRole
iam:PutRolePolicy
iam:PassRole
lambda:CreateFunction
lambda:EnableReplication
lambda:DeleteFunction
lambda:GetFunction
lambda:GetFunctionConfiguration
lambda:PublishVersion
lambda:UpdateFunctionCode
lambda:UpdateFunctionConfiguration
lambda:ListTags
lambda:TagResource
lambda:UntagResource
route53:ChangeResourceRecordSets
route53:ListHostedZonesByName
route53:ListResourceRecordSets
s3:CreateBucket
s3:GetAccelerateConfiguration
s3:GetObject
s3:ListBucket
s3:PutAccelerateConfiguration
s3:PutBucketPolicy
s3:PutObject
s3:PutBucketTagging
s3:GetBucketTagging
lambda:ListEventSourceMappings
lambda:CreateEventSourceMapping
iam:UpdateAssumeRolePolicy
```

```
iam:DeleteRolePolicy
sqs:CreateQueue           // SQS only needed if using ISR feature
sqs:DeleteQueue
sqs:GetQueueAttributes
sqs:SetQueueAttributes
amplify:GetApp
amplify:GetBranch
amplify:UpdateApp
amplify:UpdateBranch
```

Resolución de problemas de implementaciones de SSR en Next.js 11

Si tiene problemas imprevistos al implementar una aplicación SSR clásica (solo para Next.js 11) con Amplify, consulte los siguientes temas de resolución de problemas.

Temas

El directorio de salida de mi aplicación se anuló

El directorio de salida de una aplicación de Next.js implementada con Amplify debe configurarse en `.next`. Si se está anulando el directorio de salida de la aplicación, compruebe el archivo `next.config.js`. Para configurar el directorio de salida de la compilación como predeterminado en `.next`, elimine la siguiente línea del archivo:

```
distDir: 'build'
```

Compruebe que el directorio de salida se haya configurado `.next` en su configuración de compilación. Para obtener información sobre cómo ver la configuración de compilación de su aplicación, consulte Ajuste de la configuración de compilación de una aplicación.

A continuación se muestra un ejemplo de la configuración de compilación de una aplicación donde `baseDirectory` se configura en `.next`.

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

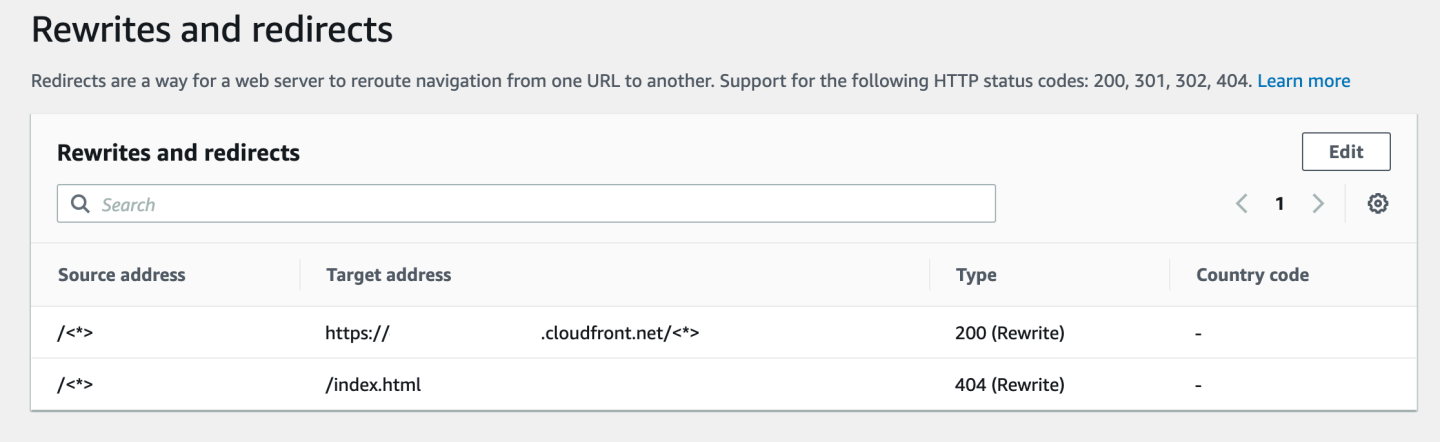
Recibo un error 404 después de implementar mi sitio SSR

Si recibe un error 404 después de implementar su sitio, el problema podría deberse a que se ha anulado el directorio de salida. Para comprobar el archivo `next.config.js` y comprobar el directorio de salida de la compilación correcto en las especificaciones de compilación de la aplicación, siga los pasos del tema anterior, El directorio de salida de mi aplicación se anuló.

A mi aplicación le falta la regla de reescritura para las distribuciones SSR CloudFront

Cuando implementas una aplicación SSR, Amplify crea una regla de reescritura para CloudFront tus distribuciones SSR. Si no puedes acceder a tu aplicación en un navegador web, comprueba que la regla de CloudFront reescritura exista para tu aplicación en la consola de Amplify. Si falta, puede añadirla manualmente o volver a implementar la aplicación.

Para ver o editar las reglas de reescritura y redireccionamiento de una aplicación en la consola de Amplify, en el panel de navegación, elija Configuración de la aplicación y, a continuación, Reescrituras y redireccionamientos. La siguiente captura de pantalla muestra un ejemplo de las reglas de reescritura que Amplify crea en su lugar al implementar una aplicación SSR. Observa que, en este ejemplo, existe una regla de CloudFront reescritura.



Mi aplicación es demasiado grande para implementarla

Amplify limita el tamaño de una implementación de SSR a 50 MB. Si intenta implementar una aplicación SSR de Next.js en Amplify y aparece un error `RequestEntityTooLargeException`, su aplicación es demasiado grande para implementarla. Puede intentar solucionar este problema añadiendo un código de limpieza de la memoria caché a su archivo `next.config.js`.

A continuación se muestra un ejemplo del código del archivo `next.config.js` que limpia la memoria caché.

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

Mi compilación falla debido a un error de memoria insuficiente

Next.js le permite almacenar en la memoria caché los artefactos de compilación para mejorar el rendimiento en las compilaciones posteriores. Además, el AWS CodeBuild contenedor de Amplify comprime y carga esta caché en Amazon S3, en su nombre, para mejorar el rendimiento de la

compilación posterior. Esto podría provocar un error de compilación debido a un error de memoria insuficiente.

Realice las siguientes acciones para evitar que su aplicación supere el límite de memoria durante la fase de compilación. En primer lugar, elimine `.next/cache/**/*` de la sección cache.paths de su configuración de compilación. A continuación, elimine la variable de entorno `NODE_OPTIONS` de su archivo de configuración de compilación. En su lugar, configure la variable de entorno `NODE_OPTIONS` en la consola de Amplify para definir el límite máximo de memoria del nodo. Para obtener más información sobre cómo configurar variables de entorno utilizando la consola de Amplify, consulte Configuración de variables de entorno.

Después de realizar estos cambios, intente realizar la compilación de nuevo. Si tiene éxito, añada de nuevo `.next/cache/**/*` a la sección cache.paths del archivo de configuración de compilación.

Para obtener más información sobre la configuración de la caché de Next.js para mejorar el rendimiento de la compilación, consulte AWS CodeBuild en el sitio web Next.js.

Mi aplicación tiene las ramificaciones SSR y SSG

No puede implementar una aplicación que tenga ramificaciones SSR y SSG a la vez. Si necesita implementar las ramificaciones SSR y SSG, debe implementar una aplicación que solo utilice las ramificaciones SSR y otra aplicación que solo utilice las ramificaciones SSG.

Mi aplicación almacena los archivos estáticos en una carpeta con una ruta reservada

Next.js puede almacenar archivos estáticos de una carpeta denominada `public` que esté almacenada en el directorio raíz del proyecto. Al implementar y alojar una aplicación de Next.js con Amplify, su proyecto no puede incluir carpetas con la ruta `public/static`. Amplify reserva la ruta `public/static` para utilizarla al distribuir la aplicación. Si su aplicación incluye esta ruta, debe cambiar el nombre de la carpeta `static` antes de implementarla con Amplify.

Mi aplicación ha alcanzado un CloudFront límite

CloudFront las cuotas de servicio limitan su AWS cuenta a 25 distribuciones con funciones Lambda @Edge asociadas. Si supera esta cuota, puede eliminar de su cuenta CloudFront las distribuciones no utilizadas o solicitar un aumento de la cuota. Para obtener más información, consulte Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas.

Las variables de entorno no se transfieren a las funciones de Lambda

Las variables de entorno que especifiques en la consola de Amplify para una aplicación SSR no se transfieren a las funciones de la aplicación. AWS Lambda Consulte Conversión de las variables

de entorno en accesibles para los tiempos de ejecución del servidor para obtener instrucciones detalladas sobre cómo añadir variables de entorno a las que puede hacer referencia desde las funciones de Lambda.

Las funciones de Lambda@Edge se crean en la región este de EE. UU. (Norte de Virginia)

Al implementar una aplicación de Next.js, Amplify crea funciones de Lambda @Edge para personalizar el contenido que se entrega. CloudFront Las funciones de Lambda@Edge se crean en la región este de EE. UU. (Norte de Virginia), en lugar de en la región en la que se implementa la aplicación. Se trata de una restricción de Lambda@Edge. Para obtener más información sobre las funciones de Lambda @Edge, consulte Restricciones de las funciones periféricas en la Guía para CloudFront desarrolladores de Amazon.

Mi aplicación Next.js utiliza características no compatibles

Las aplicaciones implementadas con Amplify son compatibles con las versiones principales de Next.js hasta la versión 11. Para obtener una lista detallada de las características de Next.js compatibles y no compatibles con Amplify, consulte supported features.

Al implementar una nueva aplicación de Next.js, Amplify utiliza la versión compatible más reciente de Next.js de forma predeterminada. Si ya tiene una aplicación de Next.js que ha implementado en Amplify con una versión anterior de Next.js, puede migrar la aplicación al proveedor SSR de procesamiento de Amplify Hosting. Para obtener instrucciones, consulte Migración de una aplicación SSR de Next.js 11 al procesamiento de Amplify Hosting.

Las imágenes de mi aplicación Next.js no se cargan

Al añadir imágenes a la aplicación Next.js mediante el componente `next/image`, el tamaño de la imagen no puede superar 1 MB. Al implementar la aplicación en Amplify, las imágenes de más de 1 MB devolverán un error 503. Esto se debe a un límite de Lambda@Edge que restringe el tamaño de una respuesta generada por una función de Lambda, incluidos los encabezados y el cuerpo, a 1 MB.

El límite de 1 MB se aplica a otros artefactos de la aplicación, como los archivos PDF y de documentos.

Regiones no admitidas

Amplify no admite la implementación de aplicaciones SSR clásicas (solo Next.js 11) en todas las regiones de AWS en las que Amplify esté disponible. El SSR clásico (solo Next.js 11) no se admite

en las siguientes regiones: Europa (Milán) eu-south-1, Medio Oriente (Baréin) me-south-1 y Asia Pacífico (Hong Kong) ap-east-1.

Precios de las aplicaciones SSR

Al implementar una aplicación SSR, el procesamiento de Amplify Hosting administra los recursos necesarios para implementar la aplicación SSR en su lugar. Para obtener información sobre los gastos de procesamiento de Amplify Hosting, consulte los precios de AWS Amplify.

Resolución de problemas de las implementaciones de SSR

Si tiene problemas imprevistos al implementar una aplicación SSR con la computación de Amplify Hosting, consulte Solución de problemas de renderización del servidor en el capítulo de solución de problemas.

Avanzado: Adaptadores de código abierto

Los autores de los marcos pueden usar la especificación de implementación basada en el sistema de archivos para desarrollar adaptadores de compilación de código abierto personalizados para sus marcos específicos. Estos adaptadores transformarán la salida de la compilación de una aplicación en un paquete de implementación que se ajuste a la estructura de directorios prevista de Amplify Hosting. Este paquete de implementación incluirá todos los archivos y activos necesarios para alojar una aplicación, incluida la configuración del tiempo de ejecución, como las reglas de enrutamiento.

Si no utiliza un marco, puede desarrollar su propia solución para generar la salida de compilación que Amplify espera.

Temas

- Uso de la especificación de implementación de Amplify Hosting para configurar la salida de la compilación
- Implementación de un servidor Express mediante el manifiesto de implementación
- Integración de la optimización de imágenes para autores de marcos
- Uso de adaptadores de código abierto para cualquier marco SSR

Uso de la especificación de implementación de Amplify Hosting para configurar la salida de la compilación

La especificación de implementación de Amplify Hosting es una especificación basada en un sistema de archivos que define la estructura de directorios que facilita las implementaciones en Amplify Hosting. Un marco puede generar esta estructura de directorios prevista como resultado de su comando de compilación, lo que permite que el marco utilice los elementos primitivos de servicio de Amplify Hosting. Amplify Hosting entiende la estructura del paquete de implementación y lo implementa como corresponde.

Para ver una demostración en vídeo en la que se explica cómo utilizar la especificación de despliegue, consulte Cómo alojar cualquier sitio web mediante AWS Amplify el YouTube canal Amazon Web Services.

A continuación se incluye un ejemplo de la estructura de carpetas que Amplify espera para el paquete de implementación. En un nivel superior, tiene una carpeta denominada `static`, una carpeta denominada `compute` y un archivo de manifiesto de implementación denominado `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
```

Compatibilidad de los elementos primitivos de SSR con Amplify

La especificación de implementación de Amplify Hosting define un contrato que se corresponde estrechamente con los siguientes elementos primitivos.

Activos estáticos

Proporciona a los marcos la capacidad de alojar archivos estáticos.

Computación

Proporciona a los marcos la capacidad de ejecutar un servidor HTTP de Node.js en el puerto 3000.

Optimización de imágenes

Proporciona a los marcos un servicio para optimizar las imágenes en tiempo de ejecución.

Reglas de enrutamiento

Proporciona a los marcos un mecanismo para asignar las rutas de las solicitudes entrantes a destinos específicos.

La .amplify-hosting/static directory

Debe colocar en el directorio `.amplify-hosting/static` todos los archivos estáticos de acceso público que estén destinados a distribuirse desde la URL de la aplicación. Los archivos de este directorio se distribuyen a través del elemento primitivo de activos estáticos.

Se puede acceder a los archivos estáticos en la raíz (/) de la URL de la aplicación sin hacer ningún cambio en su contenido, nombre de archivo o extensión. Además, los subdirectorios se conservan en la estructura de URL y aparecen antes del nombre del archivo. Por ejemplo, `.amplify-hosting/static/favicon.ico` se distribuirá desde `https://myAppId.amplify-hostingapp.com/favicon.ico` y `.amplify-hosting/static/_nuxt/main.js` se distribuirá desde `https://myAppId.amplify-hostingapp.com/_nuxt/main.js`.

Si un marco admite la posibilidad de modificar la ruta base de la aplicación, debe anteponer la ruta base a los activos estáticos del directorio `.amplify-hosting/static`. Por ejemplo, si la ruta base es `/folder1/folder2`, la salida de la compilación de un activo estático llamado `main.css` será `.amplify-hosting/static/folder1/folder2/main.css`.

La .amplify-hosting/compute directory

Un único recurso de computación se representa mediante un único subdirectorio denominado `default` que se incluye en el directorio `.amplify-hosting/compute`. La ruta es `.amplify-hosting/compute/default`. Este recurso de computación se asigna al elemento primitivo de computación de Amplify Hosting.

El contenido del subdirectorio `default` debe cumplir con las siguientes reglas.

- Debe existir un archivo en la raíz del subdirectorio `default` para que sirva como punto de entrada al recurso de computación.
- El archivo de punto de entrada debe ser un módulo de Node.js y debe iniciar un servidor HTTP que escuche en el puerto 3000.
- Puede colocar otros archivos en el subdirectorio `default` y hacer referencia a ellos desde el código en el archivo de punto de entrada.
- El contenido del subdirectorio debe ser independiente. El código del módulo de punto de entrada no puede hacer referencia a ningún módulo de fuera del subdirectorio. Tenga en cuenta que los marcos pueden agrupar su servidor HTTP de la forma que deseen. Si el proceso de computación se puede iniciar con el comando `node server.js`, donde `server.js is` es el nombre del archivo de entrada, desde el subdirectorio, Amplify considera que la estructura del directorio se ajusta a la especificación de implementación.

Amplify Hosting agrupa e implementa todos los archivos del subdirectorio `default` en un recurso de computación aprovisionado. Se asignan 512 MB de almacenamiento efímero a cada recurso de computación. Este almacenamiento no se comparte entre las instancias de ejecución, sino que se comparte entre las invocaciones posteriores de la misma instancia de ejecución. Las instancias de ejecución están limitadas a un tiempo máximo de ejecución de 15 minutos y la única ruta en la que se puede escribir dentro de la instancia de ejecución es el directorio `/tmp`. El tamaño comprimido de cada paquete de recursos de computación no puede superar los 220 MB. Por ejemplo, el subdirectorio `.amplify/compute/default` no puede superar los 220 MB cuando está comprimido.

La .amplify-hosting/deploy-manifest.json archivo

Utilice el archivo `deploy-manifest.json` para almacenar los detalles de configuración y los metadatos de una implementación. Como mínimo, un archivo `deploy-manifest.json` debe incluir

un atributo `version`, el atributo `routes` con una ruta de método catch-all especificada y el atributo `framework` con los metadatos del marco especificados.

En la siguiente definición de objeto se muestra la configuración de un manifiesto de implementación.

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

En los temas siguientes se describen los detalles y el uso de cada atributo del manifiesto de implementación.

Uso del atributo version

El atributo `version` define la versión de la especificación de implementación que se está implementando. Actualmente, la única versión para la especificación de implementación de Amplify Hosting es la versión 1. En el siguiente JSON de ejemplo se muestra el uso del atributo `version`.

```
"version": 1
```

Uso del atributo routes

El atributo `routes` permite a los marcos utilizar el elemento primitivo de reglas de enrutamiento de Amplify Hosting. Las reglas de enrutamiento proporcionan un mecanismo para enrutar las rutas de solicitudes entrantes a un destino específico del paquete de implementación. Las reglas de enrutamiento solo dictan el destino de una solicitud entrante y se aplican después de que las reglas de reescritura y redireccionamiento hayan transformado la solicitud. Para obtener más información sobre cómo Amplify Hosting gestiona las reescrituras y los redireccionamientos, consulte Configuración de redirecciones y reescrituras de una aplicación de Amplify.

Las reglas de enrutamiento no reescriben ni transforman la solicitud. Si una solicitud entrante coincide con el patrón de ruta de una ruta, la solicitud se enruta tal cual al destino de la ruta.

Las reglas de enrutamiento especificadas en la matriz `routes` deben cumplir con las siguientes reglas.

- Se debe especificar una ruta de método catch-all. Una ruta de método catch-all tiene el patrón `/*` que coincide con todas las solicitudes entrantes.
- La matriz `routes` puede contener un máximo de 25 elementos.
- Debe especificar una ruta `Static` o una ruta `Compute`.
- Si especifica una ruta `Static`, el directorio `.amplify-hosting/static` debe existir.
- Si especifica una ruta `Compute`, el directorio `.amplify-hosting/compute` debe existir.
- Si especifica una ruta `ImageOptimization`, también debe especificar una ruta `Compute`. Es necesario hacerlo porque la optimización de imágenes aún no es compatible con aplicaciones puramente estáticas.

En la siguiente definición de objeto se muestra la configuración del objeto `Route`.

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

En la siguiente tabla se describen las propiedades del objeto `Route`.

Clave	Tipo	Obligatorio	Descripción
path	Cadena	Sí	Define un patrón que coincide con las rutas de las solicitudes entrantes (excepto la cadena de consulta). La longitud máxima de la ruta es de 255 caracteres. Una ruta debe empezar por la barra diagonal `/`.

Clave	Tipo	Obligatorio	Descripción
			Una ruta puede contener cualquier a de los siguientes caracteres: [A-Z], [a-z], [0-9], [_-.*$/~"'@:+]. En el caso de la coincidencia de patrones, solo se admiten los siguientes caracteres comodín: • * (coincide con 0 o más caracteres). • El patrón /* se denomina patrón de método catch-all y coincidirá con todas las solicitudes entrantes.

Clave	Tipo	Obligatorio	Descripción
destino	Destino	Sí	Objeto que define el destino al que se debe enrutar la solicitud coincidente. Si se especifica una ruta `Compute`, debe existir un objeto `ComputeResource` correspondiente. Si se especifica una ruta `ImageOptimization` , también se debe especificar `imageSettings` .

Clave	Tipo	Obligatorio	Descripción
fallback	Destino	No	Objeto que define el destino de reserva si el destino original devuelve un error 404. El tipo `target` y el tipo `fallback` no pueden ser iguales para una ruta específica. Por ejemplo, no se permite una acción de reserva de `Static` a `Static`. Las acciones de reserva solo se admiten en el caso de las solicitud es GET que no tienen cuerpo. Si hay un cuerpo en la solicitud , se eliminará durante la acción de reserva.

En la siguiente definición de objeto se muestra la configuración del objeto `Target`.

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

En la siguiente tabla se describen las propiedades del objeto `Target`.

Clave	Tipo	Obligatorio	Descripción
kind	Targetkind	Sí	`enum` que define el tipo de destino. Los valores válidos son `Static`, `Compute` y `ImageOpti mization` .
src	Cadena	Sí para `Compute` No para otros elementos primitivos	Cadena que especific a el nombre del subdirectorio en el paquete de implement ación que contiene el código ejecutable del elemento primitivo . Válido y obligatorio solo para el elemento primitivo Compute. El valor debe apuntar a uno de los recursos de computación presentes en el paquete de implement ación. En la actualida d, el único valor que se admite para este campo es `default`.
cacheControl	Cadena	No	Cadena que especific a el valor del encabezado Cache-Control que se va a aplicar a la respuesta . Válido solo para los estáticos y los

Clave	Tipo	Obligatorio	Descripción
			ImageOptimization primitivos. Los encabezados personalizados anulan el valor especificado. Para obtener más información sobre los encabezados de cliente de Amplify Hosting, consulte Configuración de encabezados HTTP personalizados para una aplicación de Amplify. **Note** Este encabezado Cache-Control solo se aplica a las respuestas correctas con un código de estado establecido en 200 (OK).

En la siguiente definición de objeto se muestra el uso de la enumeración `TargetKind`.

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

En la siguiente lista se especifican los valores válidos de la enumeración `TargetKind`.

Estático

Enruta las solicitudes al elemento primitivo de activos estáticos.

Computación

Enruta las solicitudes al elemento primitivo de computación.

ImageOptimization

Enruta las solicitudes al elemento primitivo de optimización de imágenes.

En el siguiente JSON de ejemplo se muestra el uso del atributo `routes` con varias rutas de enrutamiento especificadas.

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

```
      }
    },
    {
      "path": "/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ]
```

Para obtener más información sobre cómo especificar reglas de enrutamiento en el manifiesto de implementación, consulte Prácticas recomendadas para configurar reglas de enrutamiento.

Uso del atributo computeResources

El atributo `computeResources` permite a los marcos proporcionar metadatos sobre los recursos de computación aprovisionados. Cada recurso de computación debe tener una ruta correspondiente asociada.

En la siguiente definición de objeto se muestra el uso del objeto `ComputeResource`.

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

```
};

type ComputeRuntime = 'nodejs16.x' | 'nodejs18.x' | 'nodejs20.x';
```

En la siguiente tabla se describen las propiedades del objeto `ComputeResource`.

Clave	Tipo	Obligatorio	Descripción
name	Cadena	Sí	Especifica el nombre del recurso de computación. El nombre debe coincidir con el nombre del subdirectorio que está dentro de `.amplify-hosting/compute directory` . Para la versión 1 de la especificación de implementación, el único valor válido es `default`.
tiempo de ejecución	ComputeRuntime	Sí	Define el tiempo de ejecución del recurso de computación aprovisionado. Los valores válidos son `nodejs16.x` , `nodejs18.x` y `nodejs20.x` .
entrypoint	Cadena	Sí	Especifica el nombre del archivo de inicio desde el que se

Clave	Tipo	Obligatorio	Descripción
			ejecutará el código para el recurso de computación especific ado. El archivo debe estar dentro del subdirectorio que representa un recurso de computación.

Si tiene una estructura de directorios con un aspecto similar al siguiente.

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

El JSON del atributo `computeResource` tendrá el siguiente aspecto.

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

Uso del atributo imageSettings

El atributo `imageSettings` permite a los marcos personalizar el comportamiento del elemento primitivo de optimización de imágenes, que proporciona una optimización de imágenes bajo demanda en tiempo de ejecución.

En la siguiente definición de objeto se muestra el uso del objeto `ImageSettings`.

```
type ImageSettings = {
  sizes: number[];
  domains: string[];
```

```
  remotePatterns: RemotePattern[];
  formats: ImageFormat[];
  minumumCacheTTL: number;
  dangerouslyAllowSVG: boolean;
};

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

En la siguiente tabla se describen las propiedades del objeto `ImageSettings`.

Clave	Tipo	Obligatorio	Descripción
sizes	Number[]	Sí	Matriz de anchos de imagen admitidos.
domains	String[]	Sí	Matriz de dominios externos permitidos que pueden utilizar la optimización de imágenes. Deje la matriz vacía para permitir que solo el dominio de implementación utilice la optimización de imágenes.
remotePatterns	RemotePattern[]	Sí	Matriz de patrones externos permitidos que pueden utilizar la optimización de imágenes. Similar a domains, pero proporciona más control con expresion es regulares (regex).

Clave	Tipo	Obligatorio	Descripción
formats	ImageFormat[]	Sí	Matriz de formatos de imagen de salida permitidos.
minimumCacheTTL	Número	Sí	Duración del almacenamiento en caché en segundos para las imágenes optimizadas.
dangerouslyAllowSVG	Booleano	Sí	Permite introducir imágenes en formato SVG. URLs De forma predeterminada, está deshabilitada por motivos de seguridad.

En la siguiente definición de objeto se muestra el uso del objeto `RemotePattern`.

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

En la siguiente tabla se describen las propiedades del objeto `RemotePattern`.

Clave	Tipo	Obligatorio	Descripción
protocolo	Cadena	No	Protocolo del patrón remoto permitido. Los valores válidos son `http` o `https`.

Clave	Tipo	Obligatorio	Descripción
hostname	Cadena	Sí	Nombre de host del patrón remoto permitido. Puede especificar un literal o un comodín. Un carácter único “*” coincide con un único subdominio. Un carácter doble “**” coincide con cualquier cantidad de subdominios. Amplify no permite caracteres comodín generales cuando solo se especifica “**”.
puerto	Cadena	No	Puerto del patrón remoto permitido.
pathname	Cadena	No	Nombre de ruta del patrón remoto permitido.

En el siguiente ejemplo se muestra el atributo `imageSettings`.

```
"imageSettings": {
    "sizes": [
      100,
      200
    ],
    "domains": [
      "example.com"
    ],
    "remotePatterns": [
```

```
      {
        "protocol": "https",
        "hostname": "example.com",
        "port": "",
        "pathname": "/**",
      }
    ],
    "formats": [
      "image/webp"
    ],
    "minumumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  }
```

Uso del atributo framework

Utilice el atributo `framework` para especificar los metadatos del marco.

En la siguiente definición de objeto se muestra la configuración del objeto `FrameworkMetadata`.

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

En la siguiente tabla se describen las propiedades del objeto `FrameworkMetadata`.

Clave	Tipo	Obligatorio	Descripción
name	Cadena	Sí	Nombre del marco.
versión	Cadena	Sí	Versión del marco. Debe ser una cadena válida de control de versiones semántico (semver).

Prácticas recomendadas para configurar reglas de enrutamiento

Las reglas de enrutamiento proporcionan un mecanismo para enrutar las rutas de solicitudes entrantes a destinos específicos del paquete de implementación. En un paquete de implementación, los autores de marcos pueden enviar archivos a la salida de la compilación que se implementan en cualquiera de los siguientes destinos:

- Elemento primitivo de activos estáticos: los archivos se encuentran en el directorio `.amplify-hosting/static`.
- Elemento primitivo de computación: los archivos se encuentran en el directorio `.amplify-hosting/compute/default`.

Los autores de marcos también proporcionan una matriz de reglas de enrutamiento en el archivo de manifiesto de implementación. Cada regla de la matriz se compara con la solicitud entrante en orden de recorrido secuencial hasta que haya una coincidencia. Cuando hay una regla coincidente, la solicitud se enruta al destino especificado en la regla coincidente. De forma opcional, se puede especificar un destino de reserva para cada regla. Si el destino original devuelve un error 404, la solicitud se enruta al destino de reserva.

La especificación de implementación requiere que la última regla del orden de recorrido sea una regla de método catch-all. Se especifica una regla de método catch-all con la ruta `/*`. Si la solicitud entrante no coincide con ninguna de las rutas anteriores de la matriz de reglas de enrutamiento, la solicitud se enruta al destino de la regla de método catch-all.

Para marcos SSR como Nuxt.js, el objetivo de la regla general tiene que ser la primitiva de cálculo. Esto se debe a que las aplicaciones de SSR tienen páginas representadas del servidor con rutas que no son predecibles en el momento de la compilación. Por ejemplo, si un Nuxt.js la aplicación tiene una página en la `/blog/[slug]` que `[slug]` hay un parámetro de ruta dinámica. El destino de la regla de método catch-all es la única forma de enrutar las solicitudes a estas páginas.

Por el contrario, se pueden usar patrones de ruta específicos para dirigirse a rutas conocidas en el momento de la compilación. Por ejemplo: Nuxt.js sirve los activos estáticos de la `/_nuxt` ruta. Esto significa que es posible dirigirse a la ruta `/_nuxt/*` mediante una regla de enrutamiento específica que enrute las solicitudes al elemento primitivo de activos estáticos.

Enrutamiento de carpetas públicas

La mayoría de los marcos de SSR ofrecen la posibilidad de distribuir activos estáticos mutables desde una carpeta `public`. Por lo general, los archivos como `favicon.ico` y `robots.txt` se

guardan en la carpeta `public` y se distribuyen desde la URL raíz de la aplicación. Por ejemplo, el archivo `favicon.ico` se distribuye desde `https://example.com/favicon.ico`. Tenga en cuenta que no existe un patrón de ruta predecible para estos archivos. El nombre del archivo los dicta casi en su totalidad. La única forma de dirigirse a los archivos de la carpeta `public` consiste en utilizar la ruta de método catch-all. Sin embargo, el destino de la ruta de método catch-all debe ser el elemento primitivo de computación.

Recomendamos uno de los siguientes enfoques para administrar la carpeta `public`.

1. Use un patrón de rutas para dirigirse a las rutas de solicitud que contienen extensiones de archivo. Por ejemplo, se puede utilizar `/*.*` para dirigirse a todas las rutas de solicitud que contienen una extensión de archivo.

   Tenga en cuenta que este enfoque puede ser poco fiable. Por ejemplo, si hay archivos sin extensiones de archivo en la carpeta `public`, esta regla no se dirige a ellos. Otro problema que hay que tener en cuenta con este enfoque es que la aplicación podría tener páginas con puntos en los nombres. Por ejemplo, la regla `/*.*` se dirigirá a una página en `/blog/2021/01/01/hello.world`. Esto no es lo ideal, ya que la página no es un activo estático. Sin embargo, puede agregar un destino de reserva a esta regla para garantizar que, cuando se produzca un error 404 del elemento primitivo estático, la solicitud utilice el elemento primitivo de computación como reserva.

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

2. Identifique los archivos de la carpeta `public` en el momento de la compilación y emita una regla de enrutamiento para cada archivo. Este enfoque no es escalable, ya que la especificación de implementación impone un límite de 25 reglas.

   ```
   {
       "path": "/favicon.ico",
       "target": {
   ```

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

3. Recomiende a los usuarios del marco almacenar todos los activos estáticos mutables en una subcarpeta dentro de la carpeta `public`.

   En el siguiente ejemplo, el usuario puede almacenar todos los activos estáticos mutables dentro de la carpeta `public/assets`. A continuación, se puede utilizar una regla de enrutamiento con el patrón de ruta `/assets/*` para dirigirse a todos los activos estáticos mutables de la carpeta `public/assets`.

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

4. Especifique una reserva estática para la ruta de método catch-all. Este enfoque presenta algunos inconvenientes que se describen en más detalle en la siguiente sección Enrutamiento de reserva de método catch-all.

Enrutamiento de reserva de método catch-all

Para marcos SSR como Nuxt.js, donde se especifica una ruta general para el objetivo primitivo de cálculo, los autores del framework podrían considerar la posibilidad de especificar una alternativa estática para la ruta general a fin de resolver el problema de enrutamiento de carpetas. `public` Sin embargo, este tipo de regla de enrutamiento interrumpe las páginas 404 representadas del servidor. Por ejemplo, si el usuario final visita una página que no existe, la aplicación devuelve una página 404 con el código de estado 404. Sin embargo, si la ruta de método catch-all tiene una reserva estática, no se devuelve la página 404. En su lugar, la solicitud utiliza el elemento primitivo estático como reserva y, aun así, termina con un código de estado 404, pero no se devuelve la página 404.

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

Enrutamiento de rutas base

Está previsto que los marcos que ofrecen la posibilidad de modificar la ruta base de la aplicación antepongan la ruta base a los activos estáticos del directorio `.amplify-hosting/static`. Por ejemplo, si la ruta base es `/folder1/folder2`, la salida de la compilación de un activo estático llamado main.css será `.amplify-hosting/static/folder1/folder2/main.css`.

Esto significa que las reglas de enrutamiento también deben actualizarse para reflejar la ruta base. Por ejemplo, si la ruta base es `/folder1/folder2`, la regla de enrutamiento de los activos estáticos de la carpeta `public` tendrá el siguiente aspecto.

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

Del mismo modo, las rutas del servidor también deben tener la ruta base antepuesta. Por ejemplo, si la ruta base es `/folder1/folder2`, la regla de enrutamiento de la ruta `/api` tendrá el siguiente aspecto.

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

Sin embargo, la ruta base no debe anteponerse a la ruta de método catch-all. Por ejemplo, si la ruta base es `/folder1/folder2`, la ruta de método catch-all seguirá siendo como la siguiente.

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

Ejemplos de rutas de Nuxt.js

A continuación se incluye un archivo `deploy-manifest.json` de ejemplo para una aplicación de Nuxt en el que se muestra cómo especificar las reglas de enrutamiento.

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

```
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
      "runtime": "nodejs18.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

A continuación se incluye un archivo `deploy-manifest.json` de ejemplo para Nuxt en el que se muestra cómo especificar las reglas de enrutamiento, incluidas rutas base.

```
{
  "version": 1,
  "routes": [
    {
      "path": "/base-path/_nuxt/image",
```

```
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/base-path/_nuxt/builds/meta/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
```

```
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
      "runtime": "nodejs18.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

Para obtener más información sobre el uso del atributo `routes`, consulte Uso del atributo routes.

Implementación de un servidor Express mediante el manifiesto de implementación

En este ejemplo, se explica cómo implementar un servidor Express básico mediante la especificación de implementación de Amplify Hosting. Puede utilizar el manifiesto de implementación proporcionado para especificar el enrutamiento, los recursos de computación y otras configuraciones.

Configure un servidor Express localmente antes de implementarlo en Amplify Hosting

1. Cree un nuevo directorio para su proyecto e instale Express y 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. Agregue un archivo `tsconfig.json` a la raíz de su proyecto con el siguiente contenido.

   ```
   {
     "compilerOptions": {
       "target": "es6",
   ```

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

3. Cree un directorio denominado `src` en la raíz del proyecto.
4. Cree un archivo `index.ts` en el directorio `src`. Será el punto de entrada a la aplicación que inicia un servidor Express. El servidor debe configurarse para escuchar en el puerto 3000.

```
// src/index.ts
import express from 'express';

const app: express.Application = express();
const port = 3000;

app.use(express.text());

app.listen(port, () => {
  console.log(`server is listening on ${port}`);
});

// Homepage
app.get('/', (req: express.Request, res: express.Response) => {
  res.status(200).send("Hello World!");
});

// GET
app.get('/get', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-get-header", "get-header-value").send("get-response-
from-compute");
});

//POST
app.post('/post', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-post-header", "post-header-
value").send(req.body.toString());
});
```

```
//PUT
app.put('/put', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-put-header", "put-header-
value").send(req.body.toString());
});

//PATCH
app.patch('/patch', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-patch-header", "patch-header-
value").send(req.body.toString());
});

// Delete
app.delete('/delete', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-delete-header", "delete-header-value").send();
});
```

5. Agregue los siguientes scripts al archivo `package.json`.

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

6. Cree un directorio denominado `public` en la raíz del proyecto. A continuación, cree un archivo denominado `hello-world.txt` con el siguiente contenido.

```
Hello world!
```

7. Agregue un archivo `.gitignore` a la raíz de su proyecto con el siguiente contenido.

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

Configuración del manifiesto de implementación de Amplify

1. Cree un archivo denominado `deploy-manifest.json` en el directorio raíz del proyecto.
2. Copie y pegue el siguiente manifiesto en el archivo `deploy-manifest.json`.

```
{
  "version": 1,
  "framework": { "name": "express", "version": "4.18.2" },
  "imageSettings": {
    "sizes": [
      100,
      200,
      1920
    ],
    "domains": [],
    "remotePatterns": [],
    "formats": [],
    "minimumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  },
  "routes": [
    {
      "path": "/_amplify/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static",
        "cacheControl": "public, max-age=2"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
```

```
    {
      "name": "default",
      "runtime": "nodejs18.x",
      "entrypoint": "index.js"
    }
  ]
}
```

En el manifiesto se describe cómo Amplify Hosting debe gestionar la implementación de su aplicación. La configuración principal es la siguiente.

- version: indica la versión de la especificación de implementación que está utilizando.
- marco: ajústelo para especificar su Express configuración del servidor.
- ImageSettings: esta sección es opcional para una Express servidor, a menos que esté gestionando la optimización de imágenes.
- routes: son fundamentales para dirigir el tráfico a las partes correctas de la aplicación. La ruta `"kind": "Compute"` dirige el tráfico a la lógica del servidor.
- ComputerResources: utilice esta sección para especificar su Express tiempo de ejecución y punto de entrada del servidor.

A continuación, configure un script posterior a la compilación que transfiera los artefactos de la aplicación creada al paquete de implementación `.amplify-hosting`. La estructura de directorios se alinea con la especificación de implementación de Amplify Hosting.

Configuración del script posterior a la compilación

1. Cree un directorio denominado `bin` en la raíz del proyecto.
2. Cree un archivo denominado `postbuild.sh` en el directorio `bin`. Añada el siguiente contenido al archivo `postbuild.sh`.

```
#!/bin/bash

rm -rf ./.amplify-hosting

mkdir -p ./.amplify-hosting/compute

cp -r ./dist ./.amplify-hosting/compute/default
cp -r ./node_modules ./.amplify-hosting/compute/default/node_modules
```

```
cp -r public ./.amplify-hosting/static

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

3. Agregue un script `postbuild` al archivo `package.json`. El archivo debe tener un aspecto similar al siguiente.

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

4. Ejecute el siguiente comando para compilar la aplicación.

```
npm run build
```

5. (Opcional) Ajuste las rutas para Express. Puede modificar las rutas del manifiesto de implementación para adaptarlas al servidor Express. Por ejemplo, si no tiene ningún activo estático en el directorio `public`, es posible que solo necesite que la ruta de método catch-all `"path": "/*"` se dirija a Compute. Esto dependerá de la configuración del servidor.

La estructura de directorios final debería ser similar a la siguiente.

```
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
```

Implementación del servidor

1. Inserte el código en el repositorio de Git y, a continuación, implemente la aplicación en Amplify Hosting.
2. Actualice la configuración de compilación para apuntar `baseDirectory` a `.amplify-hosting` de la siguiente forma. Durante la compilación, Amplify detectará el archivo de manifiesto en el directorio `.amplify-hosting` e implementará el servidor Express según la configuración.

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

3. Para comprobar que la implementación se ha realizado correctamente y que el servidor funciona de forma adecuada, visite la aplicación en la URL predeterminada que proporciona Amplify Hosting.

Integración de la optimización de imágenes para autores de marcos

Los autores de marcos pueden integrar la característica de optimización de imágenes de Amplify mediante la especificación de implementación de Amplify Hosting. Para habilitar la optimización de imágenes, el manifiesto de implementación debe contener una regla de enrutamiento dirigida al servicio de optimización de imágenes. En el siguiente ejemplo se muestra cómo configurar la regla de enrutamiento.

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

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

Para obtener más información sobre cómo definir la configuración de la optimización de imágenes mediante la especificación de implementación, consulte Uso de la especificación de implementación de Amplify Hosting para configurar la salida de la compilación.

Descripción de la API de optimización de imágenes

La optimización de imágenes se puede invocar en tiempo de ejecución a través de la URL de dominio de una aplicación de Amplify, en la ruta definida por la regla de enrutamiento.

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

La optimización de imágenes impone las siguientes reglas a las imágenes.

- Amplify no puede optimizar los formatos GIF, APNG ni SVG, ni convertirlos a otro formato.
- Las imágenes SVG no se distribuyen a menos que la configuración `dangerouslyAllowSVG` esté habilitada.
- El ancho o el alto de las imágenes de origen no pueden superar los 11 MB o los 9000 píxeles.
- El límite de tamaño de una imagen optimizada es de 4 MB.

- HTTP o HTTPS es el único protocolo compatible para obtener imágenes de forma remota URLs.

Encabezados HTTP

El encabezado HTTP de la solicitud Accept se utiliza para especificar los formatos de imagen, expresados como tipos MIME, que permite el cliente (normalmente un navegador web). El servicio de optimización de imágenes intentará convertir la imagen al formato especificado. El valor especificado para este encabezado tendrá una prioridad mayor que el parámetro de consulta de formato. Por ejemplo, un valor válido para el encabezado Accept es `image/png, image/webp, */*` . La configuración de formatos especificada en el manifiesto de implementación de Amplify restringirá los formatos a los de la lista. Incluso si el encabezado Accept solicita un formato específico, se ignorará si el formato no está en la lista de permitidos.

Parámetros de solicitud del URI

En la tabla siguiente se describen los parámetros de solicitud de URI para la optimización de imágenes.

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
url	Cadena	Sí	Ruta relativa o URL absoluta a la imagen de origen. En el caso de una URL remota, se admiten los protocolos http y https. El valor debe tener codificación de URL.	`?url=https%3A%2F%2Fwww.example.com%2Fbuffalo.png`
width	Número	Sí	Ancho en píxeles de la imagen optimizada.	`?width=800`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
height	Número	No	Alto en píxeles de la imagen optimizada. Si no se especifica, la imagen se escalará automáticamente para que coincida con el ancho.	`?height=600`
fit	Valores de enumeración: `cover`, `contain`, `fill`, `inside`, `outside`	No	Forma en que se redimensiona la imagen para que se ajuste al ancho y alto especificados.	`?width=800&height=600&fit=cover`
position	Valores de enumeración: `center`, `top`, `right`, `bottom`, `left`	No	Posición que se utilizará cuando fit sea `cover` o `contain`.	`?fit=contain&position=centre`
trim	Número	No	Recorta los píxeles de todos los bordes que contienen valores similares al color de fondo especificado del píxel superior izquierdo.	`?trim=50`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
ampliar	Objeto	No	Agrega píxeles a los bordes de la imagen con el color derivado de los píxeles del borde más cercano. El formato es `{top}_{right}_{bottom}_{left}` , donde cada valor es el número de píxeles que se van a agregar.	`?extend=10_0_5_0`
extract	Objeto	No	Recorta la imagen al rectángulo especificado, delimitado por la parte superior, la izquierda, el ancho y el alto. El formato es {left}_{top}_{width}_{right}, donde cada valor es el número de píxeles que se van a recortar.	`?extract=10_0_5_0`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
formato	Cadena	No	Formato de salida deseado de la imagen optimizada.	`?format=webp`
quality	Número	No	Calidad de la imagen, de 1 a 100. Solo se utiliza al convertir el formato de la imagen.	`?quality=50`
rotate	Número	No	Rota la imagen en el ángulo especificado en número de grados.	`?rotate=45`
flip	Booleano	No	Refleja la imagen verticalmente (de arriba a abajo) en el eje X. Siempre ocurre antes de la rotación, si se produce.	`?flip`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
flop	Booleano	No	Refleja la imagen horizontalmente (de izquierda a derecha) en el eje Y. Siempre ocurre antes de la rotación, si se produce.	`?flop`
sharpen	Número	No	La nitidez mejora la definición de los bordes de la imagen. Los valores válidos se encuentran entre 0,000001 y 10.	`?sharpen=1`
median	Número	No	Aplica un filtro mediano. Esto elimina el ruido o suaviza los bordes de la imagen.	`?sharpen=3`
blur	Número	No	Aplica un desenfoque gaussiano del sigma especificado. Los valores válidos se encuentran entre 0,3 y 1000.	`?blur=20`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
gamma	Número	No	Aplica una corrección gamma para mejorar el brillo percibido de una imagen redimensionada. El valor debe estar entre 1,0 y 3,0.	`?gamma=1`
negate	Booleano	No	Invierte los colores de la imagen.	`?negate`
normalize	Booleano	No	Mejora el contraste de la imagen al ampliar su luminancia para cubrir un rango dinámico completo.	`?normalize`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
threshold	Número	No	Sustituye los píxeles de la imagen por píxeles negros si su intensidad es inferior al umbral especificado. Si la intensidad es superior al umbral, los sustituye por píxeles blancos. Los valores válidos se encuentran entre 0 y 255.	`?threshold=155`
tint	Cadena	No	Colorea la imagen con el RGB proporcionado y, al mismo tiempo, conserva la luminancia de la imagen.	`?tint=#7743CE`
grayscale	Booleano	No	Convierte la imagen a escala de grises (blanco y negro).	`?grayscale`

Códigos de estado de respuesta

En la lista siguiente se describen los códigos de estado de respuesta de la optimización de imágenes.

Success: código de estado HTTP 200

La solicitud se ha completado correctamente.

BadRequest - Código de estado HTTP 400

- Se especificó incorrectamente un parámetro de consulta de entrada.
- La URL remota no aparece como permitida en la configuración `remotePatterns`.
- La URL remota no se resuelve en una imagen.
- El ancho o alto solicitados no aparecen como permitidos en la configuración `sizes`.
- La imagen solicitada es SVG, pero la configuración `dangerouslyAllowSvg` está deshabilitada.

Not Found: código de estado HTTP 404

No se ha encontrado la imagen de origen.

Content too large: código de estado HTTP 413

La imagen de origen o la imagen optimizada superan el tamaño máximo permitido en bytes.

Descripción del almacenamiento en caché de imágenes optimizado

Amplify Hosting almacena en caché las imágenes optimizadas en nuestra CDN para que las solicitudes posteriores a la misma imagen, con los mismos parámetros de consulta, se atiendan desde la caché. El tiempo de vida (TTL) de la caché se controla mediante el encabezado `Cache-Control`. En la siguiente lista se describen las opciones para especificar el encabezado `Cache-Control`.

- Uso de la clave `Cache-Control` en la regla de enrutamiento que se dirige a la optimización de imágenes.
- Uso de encabezados personalizados definidos en la aplicación de Amplify.
- En el caso de las imágenes remotas, se respeta el encabezado `Cache-Control` devuelto por la imagen remota.

Lo `minimumCacheTTL` especificado en la configuración de optimización de la imagen define el límite inferior del Cache-Control max-age directiva. Por ejemplo, si la URL de una imagen remota responde con `Cache-Control s-max-age=10`, pero el valor de `minimumCacheTTL` es 60, se utiliza 60.

Uso de adaptadores de código abierto para cualquier marco SSR

Puede usar cualquier adaptador de compilación de marcos SSR que se haya creado para la integración con Amplify Hosting. Cada marco que ofrece un adaptador determina cómo se configura el adaptador y se conecta a su proceso de compilación. Normalmente, instalará el adaptador como una dependencia de desarrollo de npm.

Después de crear una aplicación con un marco, utilice la documentación del marco para obtener información sobre cómo instalar el adaptador de Amplify Hosting y configurarlo en el archivo de configuración de la aplicación.

A continuación, cree un archivo `amplify.yml` en el directorio raíz del proyecto. En el archivo `amplify.yml`, establezca el valor de `baseDirectory` en el directorio de salida de la compilación de la aplicación. El marco ejecuta el adaptador durante el proceso de compilación para transformar la salida en el paquete de implementación de Amplify Hosting.

El nombre del directorio de salida de la compilación puede ser cualquiera, pero el nombre de archivo `.amplify-hosting` tiene importancia. Amplify busca primero un directorio definido como `baseDirectory`. Si existe, Amplify busca la salida de la compilación en dicho directorio. Si el directorio no existe, Amplify busca la salida de la compilación en `.amplify-hosting`, incluso si el cliente no lo ha definido.

A continuación se muestra un ejemplo de la configuración de compilación de una aplicación. El valor de `baseDirectory` se establece en `.amplify-hosting` para indicar que la salida de la compilación está en la carpeta `.amplify-hosting`. Siempre que el contenido de la carpeta `.amplify-hosting` coincida con la especificación de implementación de Amplify Hosting, la aplicación se implementará correctamente.

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

Después de configurar la aplicación para usar un adaptador de marcos, puede implementarla en Amplify Hosting. Para obtener instrucciones detalladas, consulte Implementación de una aplicación SSR en Amplify.

Implementación de un sitio web estático en Amplify desde un bucket de Amazon S3

Puede utilizar la integración entre Amplify Hosting y Amazon S3 para alojar contenido estático de sitios web almacenado en S3 con solo unos pocos clics. La implementación de Amplify Hosting le brinda las siguientes ventajas y características.

- Despliegue automático en la red de entrega de AWS contenido (CDN) disponible a nivel mundial con la tecnología de CloudFront
- Compatibilidad con HTTPS
- Conectar fácilmente su sitio web a un dominio personalizado mediante la consola de Amplify
- Uso de sus propios certificados SSL personalizados
- Supervise su sitio web con registros y CloudWatch métricas de acceso integrados
- Configuración de la protección de contraseñas de su sitio web
- Creación de reglas de redireccionamiento y reescritura en la consola de Amplify

Puede iniciar el proceso de implementación desde la consola Amplify AWS CLI, la o la. AWS SDKs Solo se pueden hacer implementaciones en Amplify desde un bucket de uso general de Amazon S3 ubicado en su propia cuenta. Amplify no admite cuentas cruzadas S3 acceso a cubos.

Cuando despliega su aplicación desde un paquete de uso general de Amazon S3 a Amplify Hosting, los AWS cargos se basan en el modelo de precios de Amplify. Para obtener más información, consulte AWS Amplify Precios.

> **Important**
>
> Amplify Hosting no está disponible en todos los Regiones de AWS lugares donde Amazon S3 está disponible. Para implementar un sitio web estático en Amplify Hosting, el bucket de uso general de Amazon S3 que contiene su sitio web debe ubicarse en una región en la que esté disponible Amplify. Para ver una lista de las regiones en las que está disponible Amplify, consulte Amplify endpoints en la Referencia general de Amazon Web Services.

Consulte los siguientes temas para obtener información sobre cómo implementar y actualizar un sitio web estático de Amazon S3 a Amplify Hosting.

Temas

Implementación de un sitio web estático desde S3 uso de la consola Amplify

Utilice las siguientes instrucciones para implementar un sitio web estático nuevo desde un bucket de uso general de Amazon S3 mediante la consola de Amplify.

Implementación de un sitio web estático desde un bucket de uso general de Amazon S3 mediante la consola de Amplify

1. Inicie sesión en la consola Amplify AWS Management Console y ábrala en. https://console.aws.amazon.com/amplify/
2. En la página Todas las aplicaciones, seleccione Crear nueva aplicación.
3. En la página Comenzar a crear con Amplify, seleccione Implementar sin Git.
4. Elija Next (Siguiente).
5. En la página Iniciar una implementación manual, haga lo siguiente.
    a. En Nombre de la aplicación, escriba el nombre de la aplicación.
    b. En Nombre de la ramificación, escriba el nombre de la ramificación que desea implementar.
6. En Método, elija Amazon S3.
7. Para el S3 ubicación de los objetos que desee alojar, seleccione Examinar. Seleccione el bucket de uso general de Amazon S3 que desee utilizar y, a continuación, seleccione Elegir prefijo.
8. Elija Guardar e implementar.

Crear una política de bucket para implementar un sitio web estático desde S3 utilizando el AWS SDKs

Puede usarlo AWS SDKs para implementar un sitio web estático desde Amazon S3 en Amplify Hosting. Si despliegas tu sitio web mediante un SDK, debes crear tu propia política de bucket que conceda permiso a Amplify Hosting para recuperar los objetos de tu S3 cubo.

Para obtener más información sobre la creación de políticas de buckets, consulte Políticas de buckets para Amazon S3 en la Guía del usuario de Amazon Simple Storage Service.

El siguiente ejemplo de política de bucket otorga a Amplify Hosting permisos para enumerar buckets y recuperar objetos de bucket para la rama y el identificador de aplicación de Cuenta de AWS Amplify especificados.

Para usar este ejemplo:

- *amzn-s3-demo-website-bucket/prefix*Sustitúyalo por el nombre del bucket y el prefijo de tu sitio web.
- *111122223333*Sustitúyelo por tu Cuenta de AWS identificador.
- *region-id*Sustitúyalo por el lugar en el Región de AWS que se encuentra la aplicación Amplify, como. **us-east-1**
- *app_id*Sustitúyala por tu ID de aplicación Amplify. Esta información está disponible en la consola de Amplify.
- Reemplácelo *branch_name* por el nombre de su sucursal.

> **Note**
>
> En la política de bucket, `aws:SourceArn` debe ser un ARN de ramificación codificado en una URL (codificación porcentual).

```
{
    "Version": "2008-10-17",
    "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"
                }
            }
        }
    ]
}
```

Actualización de un sitio web estático implementado en Amplify desde un S3 bucket

Si actualiza alguno de los objetos para un sitio web estático de uso general S3 si está alojado en Amplify, debes volver a implementar la aplicación en Amplify Hosting para que los cambios surtan efecto. Amplify Hosting no detecta automáticamente los cambios en el S3 balde. Le recomendamos que utilice la AWS Command Line Interface (CLI) para actualizar el sitio web.

Sincroniza las actualizaciones con S3

Tras realizar cambios en los archivos de proyecto de su sitio web, utilice el siguiente comando s3 sync para sincronizar los cambios hechos en el directorio de origen local con el bucket de uso general de Amazon S3 de destino. Para usar este ejemplo, *<source>* sustitúyalo por el nombre de tu directorio local y *<target>* por el nombre de tu bucket de Amazon S3.

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

Reimplementación del sitio web en Amplify Hosting

Utilice el siguiente comando amplify start-deployment para volver a implementar la aplicación actualizada en un bucket de Amazon S3 en Amplify Hosting. Para usar este ejemplo, *<app_id>* sustitúyalo por el identificador de tu aplicación Amplify, *<branch_name>* por el nombre de tu sucursal y *s3://amzn-s3-demo-website-bucket/prefix* por tu S3 cubo y prefijo.

```
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
```

Actualización de un S3 implementación para usar un bucket y un prefijo en lugar de un archivo.zip

Si ya tiene un sitio web estático implementado en Amplify Hosting desde un archivo .zip en un bucket de uso general de Amazon S3, puede actualizar la implementación de la aplicación para que utilice el nombre y el prefijo del bucket que contienen los objetos que desea alojar. Este tipo de implementación elimina la necesidad de cargar un archivo independiente en el bucket que incluya el contenido comprimido del resultado de la compilación.

Migración de un sitio web estático de un archivo .zip al contenido del bucket

1. Inicie sesión en la consola Amplify AWS Management Console y ábrala en. https://console.aws.amazon.com/amplify/
2. En la página Todas las aplicaciones, elija el nombre de la aplicación implementada manualmente que quiera migrar del uso de un archivo .zip al uso directo de los archivos de la aplicación.
3. En la página Información general de la aplicación, seleccione Implementar actualizaciones.
4. En la página Implementar actualizaciones, en Método, elija Amazon S3.
5. Para el S3 ubicación de los objetos que desee alojar, seleccione Examinar. Seleccione el bucket que desee utilizar y, a continuación, Elegir prefijo.
6. Elija Guardar e implementar.

Implementación de una aplicación en Amplify sin un repositorio de Git

Las implementaciones manuales le permiten publicar su aplicación web con Amplify Hosting sin necesidad de conectarse a un proveedor de Git. Puedes arrastrar y soltar una carpeta comprimida desde tu escritorio y alojar tu sitio en cuestión de segundos. De forma alternativa, puede hacer referencia a los activos de un bucket de Amazon S3 o especificar una dirección URL pública de la ubicación en la que se almacenan los archivos.

> **Note**
>
> Las implementaciones manuales tienen un límite máximo de tamaño de archivos.zip de 5 GB debido a las limitaciones de la operación de copia de Amazon S3. Si alguno de sus artefactos de construcción supera este tamaño, considere la posibilidad de dividirlo en archivos más pequeños o utilizar un método de implementación alternativo.

En Amazon S3, también puede configurar AWS Lambda activadores para actualizar su sitio cada vez que se carguen nuevos activos. Consulte la publicación del blog sobre cómo implementar archivos almacenados en Amazon S3, Dropbox o su escritorio en la consola de AWS Amplify para obtener más información sobre la configuración de este escenario.

Amplify Hosting no admite la implementación manual de aplicaciones renderizadas del servidor (SSR). Para obtener más información, consulte Implementación de aplicaciones renderizadas del servidor con Amplify Hosting.

Implementaciones manuales de arrastrar y soltar

Para implementar manualmente una aplicación mediante la función de arrastrar y soltar

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija Crear nueva aplicación en la esquina superior derecha.
3. En la página Comenzar a crear con Amplify, seleccione Implementar sin Git. A continuación, elija Siguiente.
4. En la página Iniciar una implementación manual, especifique el nombre de la aplicación en Nombre de la aplicación.

5. En Nombre de la ramificación, introduzca un nombre significativo, como **`development`** o **`production`**.
6. En Método, elija Arrastrar y soltar.
7. Arrastre y suelte la carpeta del escritorio en la zona de colocación o utilice Eligir carpeta .zip para seleccionar el archivo del equipo. El archivo que arrastre y suelte o seleccione debe ser una carpeta comprimida con el contenido de su archivo de compilación.
8. Elija Guardar e implementar.

Implementación manual de Amazon S3 o URL

> **Note**
>
> Si está implementando un sitio web estático desde S3, el siguiente procedimiento requiere que cargue una carpeta comprimida con el contenido del resultado de la compilación en su S3 balde. Le recomendamos que implemente un sitio web estático directamente desde S3 utilizando el nombre y el prefijo del bucket. Para obtener más información acerca de este proceso simplificado, consulte Implementación de un sitio web estático en Amplify desde un bucket de Amazon S3.

Para implementar manualmente una aplicación desde Amazon S3 o una dirección URL pública

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija Crear nueva aplicación en la esquina superior derecha.
3. En la página Comenzar a crear con Amplify, seleccione Implementar sin Git. A continuación, elija Siguiente.
4. En la página Iniciar una implementación manual, especifique el nombre de la aplicación en Nombre de la aplicación.
5. En Nombre de la ramificación, introduzca un nombre significativo, como **`development`** o **`production`**.
6. En Método, elija Amazon S3 o Cualquier dirección URL.
7. El procedimiento para cargar los archivos depende del método de carga.
   - Amazon S3

a. En S3 location of objects to host, selecciona Explorar S3. A continuación, seleccione el nombre del bucket de Amazon S3 de la lista. Las listas de control de acceso (ACLs) deben estar habilitadas para el bucket que seleccione. Para obtener más información, consulte Solución de problemas de acceso al bucket de Amazon S3 para implementaciones manuales.

b. Seleccione el nombre del archivo .zip que desee implementar.

c. Seleccione Elegir prefijo.

- Cualquier dirección URL

  - En Dirección URL del recurso, introduzca la dirección URL del archivo .zip que desee implementar.

8. Elija Guardar e implementar.

> **Note**
>
> Al crear la carpeta zip, asegúrese de comprimir el contenido del resultado de la compilación y no el de la carpeta de nivel superior. Por ejemplo, si el resultado de la compilación genera una carpeta denominada “build” o “public”, navegue primero hasta esa carpeta, seleccione todo el contenido y comprímalo desde allí. Si no lo hace, aparecerá un error de “acceso denegado” porque el directorio raíz del sitio no se inicializará correctamente.

Solución de problemas de acceso al bucket de Amazon S3 para implementaciones manuales

Al crear un bucket de Amazon S3, utiliza su configuración de propiedad de objetos de Amazon S3 para controlar si las listas de control de acceso (ACLs) están habilitadas o deshabilitadas para el bucket. Para implementar manualmente una aplicación en Amplify desde un bucket de Amazon S3, ACLs debe estar habilitada en el bucket.

Si aparece un `AccessControlList` error al realizar la implementación desde un bucket de Amazon S3, significa que el bucket se creó con ACLs disabled y debe habilitarlo en la consola de Amazon S3. Para obtener instrucciones, consulte la configuración de propiedad de objetos en un bucket existente en la Guía del usuario de Amazon Simple Storage Service.

Uso de funciones de IAM con aplicaciones Amplify

Un rol de IAM es una identidad de IAM con permisos específicos. Los permisos del rol determinan lo que la identidad puede y no puede hacer en ese ámbito. AWS Puedes crear funciones de IAM en tu cuenta Cuenta de AWS y utilizarlas para delegar permisos a Amplify Hosting. Para obtener más información sobre las funciones, consulte las funciones de IAM en la Guía del usuario de IAM.

Puedes usar los siguientes tipos de funciones de IAM para conceder a Amplify Hosting los permisos que necesita para realizar acciones en tu nombre o ejecutar código de cómputo que acceda AWS a otros recursos.

Rol de servicio de IAM

Amplify asume esta función para realizar acciones en su nombre. Esta función es necesaria para las aplicaciones con recursos de back-end.

Función informática de IAM SSR

Permite que una aplicación renderizada en el lado del servidor (SSR) acceda de forma segura a recursos específicos. AWS

Función de registro de IAM SSR CloudWatch

Al implementar una aplicación SSR, la aplicación requiere una función de servicio de IAM que Amplify asume para permitir que Amplify acceda a Amazon Logs. CloudWatch

Temas

- Añadir un rol de servicio con permisos para implementar recursos de backend
- Añadir un rol de SSR Compute para permitir el acceso a los recursos AWS
- Añadir un rol de servicio con permisos para acceder a CloudWatch los registros

Añadir un rol de servicio con permisos para implementar recursos de backend

Amplify requiere permisos para implementar recursos de backend con el frontend. Se utiliza un rol de servicio para llevar esto a cabo. Un rol de servicio es el rol AWS Identity and Access Management (IAM) que proporciona a Amplify Hosting permisos para implementar, crear y administrar backends en tu nombre.

Al crear una nueva aplicación que requiera un rol de servicio de IAM, puede permitir que Amplify Hosting cree automáticamente un rol de servicio para usted o puede seleccionar un rol de IAM que ya haya creado. En esta sección, aprenderá a crear un rol de servicio de Amplify que tenga permisos administrativos de cuenta y que permita explícitamente el acceso directo a los recursos que las aplicaciones de Amplify requieren para implementar, crear y administrar los backends.

Creación de un rol de servicio Amplify en la consola de IAM

Para crear un rol de servicio

1. Abra la consola de IAM y elija Roles en la barra de navegación izquierda y, a continuación, Crear rol.
2. En la página Seleccionar tipo de entidad de confianza, elija Servicio de AWS . En Caso de uso, selecciona Amplify - Backend Deployment y, a continuación, selecciona Siguiente.
3. En la página Agregar permisos, elija Siguiente.
4. En la página Asignar nombre, revisar y crear, en Nombre de rol, ingrese un nombre relevante, como **`AmplifyConsoleServiceRole-AmplifyRole`**.
5. Acepte todos los valores predeterminados y elija Crear rol.
6. Vuelva a la consola de Amplify para adjuntar el rol a su aplicación.
   - Si está en proceso de implementar una nueva aplicación, haga lo siguiente:
     a. Actualice la lista de roles de servicio.
     b. Seleccione el rol que acaba de crear. En este ejemplo, debería tener el siguiente aspecto AmplifyConsoleServiceRole: - AmplifyRole.
     c. Seleccione Siguiente y siga los pasos para completar la implementación de la aplicación.
   - Si tiene una aplicación existente, haga lo siguiente:
     a. En el panel de navegación, selecciona Configuración de la aplicación y, a continuación, selecciona Funciones de IAM.
     b. En la página de funciones de IAM, en la sección Función de servicio, selecciona Editar.
     c. En la página Función de servicio, seleccione la función que acaba de crear de la lista de funciones de servicio.
     d. Seleccione Guardar.
7. Amplify ahora tiene permisos para implementar recursos de back-end para tu aplicación.

Editar la política de confianza de un puesto de servicio para evitar que un diputado se confunda

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. Para obtener más información, consulte Prevención de la sustitución confusa entre servicios.

En la actualidad, la política de confianza predeterminada para el rol de servicio `Amplify-Backend Deployment` impone las claves de condición del contexto global `aws:SourceArn` y `aws:SourceAccount` para evitar que se confunda el representante. Sin embargo, si ya ha creado un rol de `Amplify-Backend Deployment` en su cuenta, puede actualizar la política de confianza del rol para añadir estas condiciones y evitar que el representante se confunda.

Utilice el siguiente ejemplo para restringir el acceso a las aplicaciones de su cuenta. En el ejemplo, reemplace la región y el ID de la aplicación con su información.

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
```

Para obtener instrucciones sobre cómo editar la política de confianza de un rol mediante el AWS Management Console, consulte Modificación de un rol (consola) en la Guía del usuario de IAM.

Añadir un rol de SSR Compute para permitir el acceso a los recursos AWS

Esta integración le permite asignar una función de IAM al servicio Amplify SSR Compute para permitir que su aplicación renderizada en el lado del servidor (SSR) acceda de forma segura a recursos AWS específicos en función de los permisos de la función. Por ejemplo, puedes permitir que las funciones informáticas SSR de tu aplicación accedan de forma segura a otros AWS servicios o recursos, como Amazon Bedrock un bucket de Amazon S3, en función de los permisos definidos en la función de IAM asignada.

La función SSR Compute de IAM proporciona credenciales temporales, lo que elimina la necesidad de codificar credenciales de seguridad de larga duración en las variables de entorno. El uso de la función SSR Compute de IAM se ajusta a las mejores prácticas de AWS seguridad, que consisten en conceder permisos con privilegios mínimos y utilizar credenciales de corta duración siempre que sea posible.

Las instrucciones que aparecen más adelante en esta sección describen cómo crear una política con permisos personalizados y cómo adjuntarla a un rol. Al crear el rol, debes adjuntar una política de confianza personalizada que dé permiso a Amplify para asumir el rol. Si la relación de confianza no está definida correctamente, aparecerá un error al intentar añadir el rol. La siguiente política de confianza personalizada otorga a Amplify el permiso para asumir el rol.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Puede asociar una función de IAM suya Cuenta de AWS a una aplicación SSR existente mediante la consola Amplify o la. AWS SDKs AWS CLI El rol que adjuntas se asocia automáticamente al servicio de cómputo Amplify SSR, lo que le otorga los permisos que especifiques para acceder a otros recursos. AWS A medida que las necesidades de la aplicación cambien con el tiempo, puede modificar la función de IAM asociada sin tener que volver a implementar la aplicación. Esto proporciona flexibilidad y reduce el tiempo de inactividad de las aplicaciones.

> **Important**
>
> Usted es responsable de configurar la aplicación para que cumpla sus objetivos de seguridad y conformidad. Esto incluye administrar su función de SSR Compute, que debe configurarse

para tener el conjunto mínimo de permisos necesarios para su caso de uso. Para obtener más información, consulte Gestión de la seguridad del rol de cómputo de IAM SSR.

Crear un rol de SSR Compute en la consola de IAM

Antes de poder asociar una función de procesamiento SSR de IAM a una aplicación de Amplify, la función debe existir ya en su. Cuenta de AWS En esta sección, aprenderá a crear una política de IAM y a asociarla a una función que Amplify pueda asumir para acceder AWS a recursos específicos.

Le recomendamos que siga la práctica AWS recomendada de conceder permisos con privilegios mínimos al crear un rol de IAM. El rol de cómputo SSR de IAM solo se invoca desde funciones de cómputo de SSR y, por lo tanto, solo debe conceder los permisos necesarios para ejecutar el código.

Puede usar AWS Management Console AWS CLI, o SDKs para crear políticas en IAM. Para obtener más información, consulte Definir permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Las siguientes instrucciones muestran cómo utilizar la consola de IAM para crear una política de IAM que defina los permisos que se van a conceder al servicio Amplify Compute.

Para usar el editor de políticas JSON de la consola de IAM para crear una política

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en. https://console.aws.amazon.com/iam/
2. En el panel de navegación de la izquierda, elija Políticas.
3. Elija Crear política.
4. En la sección Editor de políticas, seleccione la opción JSON.
5. Escriba o pegue un documento de política de JSON.
6. Cuando haya terminado de agregar permisos a la política, seleccione Siguiente.
7. En la página Revisar y crear, escriba el Nombre de la política y la Descripción (opcional) para la política que está creando. Revise los Permisos definidos en esta política para ver los permisos que concede la política.
8. Elija Crear política para guardar la nueva política.

Tras crear una política, siga las instrucciones siguientes para asociarla a un rol de IAM.

Para crear un rol que conceda permisos de Amplify a recursos específicos AWS

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en. https://console.aws.amazon.com/iam/
2. En el panel de navegación de la consola, elija Roles y, a continuación, seleccione Crear rol.
3. Elija el tipo de rol Custom trust policy (Política de confianza personalizada).
4. En la sección Política de confianza personalizada, introduzca la política de confianza personalizada para el rol. Se requiere una política de confianza de roles que defina los directores en los que se puede confiar para que asuman el rol.

   Copia y pega la siguiente política de confianza para conceder al servicio Amplify permiso para asumir esta función.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

5. Resuelva las advertencias de seguridad, errores o advertencias generales surgidos durante la validación de política y luego seleccione Siguiente.
6. En la página Agregar permisos, busque el nombre de la política que creó en el procedimiento anterior y selecciónela. A continuación, elija Siguiente.
7. En Nombre de rol, ingrese un nombre de rol. Los nombres de los roles deben ser únicos dentro de su Cuenta de AWS. No distinguen entre mayúsculas y minúsculas. Por ejemplo, no puede crear funciones denominado tanto **`PRODROLE`** como **`prodrole`**. Como otros AWS recursos pueden hacer referencia al rol, no puede editar el nombre del rol una vez creado.
8. (Opcional) En Descripción, ingrese una descripción para el nuevo rol.

9. (Opcional) Seleccione Editar en las secciones Paso 1: seleccionar entidades de confianza o Paso 2: agregar permisos para modificar la política personalizada y los permisos del rol.
10. Revise el rol y, a continuación, seleccione Crear rol.

Añadir una función de procesamiento SSR de IAM a una aplicación Amplify

Una vez que hayas creado un rol de IAM en tu Cuenta de AWS, puedes asociarlo a una aplicación en la consola de Amplify.

Para añadir un rol de SSR Compute a una aplicación en la consola Amplify

1. Inicie sesión en la consola Amplify AWS Management Console y ábrala en. https://console.aws.amazon.com/amplify/
2. En la página Todas las aplicaciones, elige el nombre de la aplicación a la que quieres añadir un rol de cómputo.
3. En el panel de navegación, selecciona Configuración de la aplicación y, a continuación, selecciona Funciones de IAM.
4. En la sección Función de cálculo, selecciona Editar.
5. En la lista de roles predeterminados, busca el nombre del rol que deseas adjuntar y selecciónalo. Para este ejemplo, puede elegir el nombre del rol que creó en el procedimiento anterior. De forma predeterminada, el rol que selecciones se asociará a todas las ramas de la aplicación.

   Si la relación de confianza del rol no está definida correctamente, aparecerá un error y no podrás añadir el rol.
6. (opcional) Si tu aplicación se encuentra en un repositorio público y utiliza la creación automática de sucursales o tiene habilitadas las vistas previas web para las solicitudes de incorporación de cambios, no te recomendamos usar un rol a nivel de aplicación. En su lugar, asocie la función de cómputo únicamente a las sucursales que requieran acceso a recursos específicos. Para anular el comportamiento predeterminado a nivel de aplicación y asignar un rol a una rama específica, haz lo siguiente:

   a. En Branch, selecciona el nombre de la rama que quieres usar.
   b. En la función de cómputo, seleccione el nombre de la función que desee asociar a la rama.
7. Elija Guardar.

Gestión de la seguridad del rol de cómputo de IAM SSR

La seguridad es una responsabilidad compartida entre usted AWS y usted. Usted es responsable de configurar la aplicación para que cumpla sus objetivos de seguridad y conformidad. Esto incluye administrar su función de SSR Compute, que debe configurarse para tener el conjunto mínimo de permisos necesarios para su caso de uso. Las credenciales para el rol de SSR Compute que especifiques están disponibles de forma inmediata en el tiempo de ejecución de tu función SSR. Si el código SSR expone estas credenciales, ya sea de forma intencionada, debido a un error o al permitir la ejecución remota de código (RCE), un usuario no autorizado puede acceder a la función SSR y a sus permisos.

Cuando una aplicación de un repositorio público utiliza un rol de SSR Compute y la creación automática de sucursales o vistas previas web para las solicitudes de incorporación de cambios, debes gestionar cuidadosamente qué sucursales pueden acceder a esa función. Te recomendamos que no utilices un rol a nivel de aplicación. En su lugar, debes asignar un rol de cómputo a nivel de sucursal. Esto te permite conceder permisos solo a las sucursales que requieren acceso a recursos específicos.

Si las credenciales de su función están expuestas, lleve a cabo las siguientes acciones para eliminar todo acceso a las credenciales de la función.

1. Revoca todas las sesiones

   Para obtener instrucciones sobre cómo revocar inmediatamente todos los permisos de las credenciales del rol, consulte Revocar las credenciales de seguridad temporales del rol de IAM.
2. Eliminar el rol de la consola de Amplify

   Esta acción surte efecto de forma inmediata. No es necesario volver a implementar la aplicación.

Para eliminar un rol de cómputo en la consola de Amplify

1. Inicie sesión en la consola Amplify AWS Management Console y ábrala en. https://console.aws.amazon.com/amplify/
2. En la página Todas las aplicaciones, elija el nombre de la aplicación de la que desee eliminar la función de cómputo.
3. En el panel de navegación, selecciona Configuración de la aplicación y, a continuación, selecciona Funciones de IAM.
4. En la sección Función de cálculo, selecciona Editar.

5. Para eliminar el rol predeterminado, selecciona la X situada a la derecha del nombre del rol.
6. Seleccione Guardar.

Añadir un rol de servicio con permisos para acceder a CloudWatch los registros

Amplify envía información sobre tu tiempo de ejecución de SSR a Amazon CloudWatch Logs en tu. Cuenta de AWS Al implementar una aplicación SSR, la aplicación requiere un rol de servicio de IAM que Amplify asume cuando llama a otros servicios en su nombre. Puede permitir que el procesamiento de Amplify Hosting cree automáticamente un rol de servicio en su lugar, o puede especificar un rol que haya creado usted.

Si decides permitir que Amplify cree un rol de IAM para ti, el rol ya tendrá los permisos para crear registros. CloudWatch Si creas tu propia función de IAM, tendrás que añadir los siguientes permisos a tu política para permitir que Amplify acceda a Amazon CloudWatch Logs.

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

Configuración de dominios personalizados

Puede conectar una aplicación que haya implementado con Amplify Hosting a un dominio personalizado. Cuando usa Amplify para implementar la aplicación web, Amplify la aloja en su nombre en el dominio predeterminado `amplifyapp.com` con una URL del tipo `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`. Si conecta su aplicación a un dominio personalizado, los usuarios verán que su aplicación está alojada en una URL personalizada, como `https://www.example.com`.

Puede comprar un dominio personalizado a través de un registrador de dominios acreditado, como Amazon Route 53 o GoDaddy. Route 53 es el servicio web del sistema de nombres de dominio (DNS) de Amazon. Para obtener más información sobre el funcionamiento de Route 53, consulte Qué es Amazon Route 53. Para obtener una lista de registradores de dominio externos acreditados, consulte Accredited Registrar Directory en el sitio web de ICANN.

Cuando configura su dominio personalizado, puede utilizar el certificado administrado predeterminado que Amplify le proporciona o puede utilizar su propio certificado personalizado. Puede cambiar en cualquier momento el certificado que se usa en el dominio. Para obtener información detallada acerca de la administración de certificados, consulte Uso de certificados de SSL/TLS.

Antes de continuar con la configuración de un dominio personalizado, verifique que se cumplan los requisitos previos que se describen a continuación.

- Es propietario de nombre de dominio registrado.
- Tiene un certificado emitido o importado a AWS Certificate Manager.
- Implementó su aplicación en Amplify Hosting.

  Para obtener más información sobre cómo completar este paso, consulte Introducción a la implementación de una aplicación en Amplify Hosting.
- Tiene conocimientos básicos de la terminología de dominios y DNS.

  Para obtener más información sobre dominios y DNS, consulte Descripción de la terminología y conceptos de DNS.

Temas

- Descripción de la terminología y conceptos de DNS

- Uso de certificados de SSL/TLS
- Agregar un dominio personalizado administrado en Amazon Route 53
- Adición de un dominio personalizado administrado por un proveedor de DNS de terceros
- Actualizar los registros DNS de un dominio administrado por GoDaddy
- Actualización del certificado SSL/TLS de un dominio
- Administración de subdominios
- Configuración de subdominios comodín
- Configuración de subdominios automáticos para un dominio personalizado de Amazon Route 53
- Solución de problemas de dominios personalizados

Descripción de la terminología y conceptos de DNS

Si no está familiarizado con los términos y conceptos del sistema de nombres de dominio (DNS), los siguientes temas pueden ayudarle a entender el procedimiento para añadir dominios personalizados.

Terminología de DNS

En la siguiente lista se enumeran términos comunes de DNS. Pueden ayudarle a entender el procedimiento para añadir dominios personalizados.

CNAME

> Un nombre de registro canónico (CCNAME) es un tipo de registro de DNS que le permite enmascarar el dominio para un conjunto de páginas web y hacer que aparezcan como si se encontrasen en otros lugares. Un CNAME apunta un subdominio a un nombre de dominio completo (FQDN). Por ejemplo, puede crear un nuevo registro CNAME para mapear el subdominio www.ejemplo.com. En este caso, www es el subdominio, con el dominio FQDN branch-name.d1m7bkiki6tdw1.cloudfront.net asignado a su aplicación en la consola de Amplify.

ANAME

> Un ANAME: un registro ANAME es como un registro CNAME, pero en el nivel raíz. Un ANAME apunta la raíz de su dominio a un FQDN. Este FQDN apuntará a una dirección IP.

Servidor de nombres

Un servidor de nombres es un servidor de Internet especializado en gestionar consultas acerca de la ubicación de los diversos servicios de un nombre de dominio. Si ha configurado su dominio en Amazon Route 53, ya existe una lista de servidores de nombres asignada a su dominio.

Registro de NS

El registro NS apunta a los servidores de nombres que buscan los detalles de su dominio.

Verificación de DNS

Un sistema de nombres de dominio (DNS) es como una guía telefónica que traduce los nombres de dominio legibles por humanos en direcciones IP legibles por equipos. Al escribir **`https://google.com`** en un navegador, se realiza una operación de búsqueda en el proveedor de DNS para encontrar la dirección IP del servidor que aloja el sitio web.

Los proveedores de DNS contienen registros de dominios y sus direcciones IP correspondientes. Los registros de DNS más utilizados son CNAME, ANAME y NS.

Amplify usa un registro CNAME para verificar que posee su dominio personalizado. Si aloja su dominio con Route 53, la verificación se lleva a cabo automáticamente en su nombre. Sin embargo, si alojas tu dominio con un proveedor externo, por ejemplo GoDaddy, debes actualizar manualmente la configuración de DNS de tu dominio y añadir un nuevo registro CNAME proporcionado por Amplify.

Proceso de activación de dominios personalizados

Cuando se conecta la aplicación de Amplify a un dominio personalizado en la consola de Amplify, Amplify debe llevar a cabo varios pasos antes de que se pueda ver la aplicación mediante el dominio personalizado. La siguiente lista describe en detalle cada paso del proceso de configuración y activación del dominio.

Creación de SSL/TLS

Si utilizas un certificado gestionado, AWS Amplify emite un certificado SSL/TLS para configurar un dominio personalizado y seguro.

Configuración y verificación de SSL/TLS

Antes de emitir un certificado administrado, Amplify verifica que es el propietario del dominio. Para los dominios administrados por Amazon Route 53, Amplify actualiza automáticamente el registro de verificación de DNS. Para los dominios administrados fuera de Route 53, deberá

agregar manualmente el registro de verificación de DNS proporcionado en la consola de Amplify a su dominio con un proveedor de DNS externo.

Si utiliza un certificado personalizado, es responsable de validar la propiedad del dominio.

Activación del dominio

El dominio se ha verificado correctamente. Para los dominios administrados fuera de Route 53, tendrá que añadir manualmente los registros CNAME proporcionado en la consola de Amplify a su dominio con un proveedor de DNS externo.

Uso de certificados de SSL/TLS

Un protocolo. SSL/TLS certificate is a digital document that allows web browsers to identify and establish encrypted network connections to web sites using the secure SSL/TLS Cuando configura su dominio personalizado, puede utilizar el certificado administrado predeterminado que Amplify le proporciona o puede utilizar su propio certificado personalizado.

Con un certificado administrado, Amplify emite un certificado SSL/TLS para todos los dominios conectados a su aplicación, de modo que todo el tráfico se protege a través de HTTPS/2. El certificado predeterminado generado por AWS Certificate Manager (ACM) es válido durante 13 meses y se renueva automáticamente siempre que la aplicación esté alojada en Amplify.

> **Warning**
>
> Amplify no puede renovar el certificado si el registro de verificación de CNAME se ha modificado o eliminado en la configuración de DNS con su proveedor de dominio. Deberá eliminar y volver a añadir el dominio en la consola de Amplify.

Para usar un certificado personalizado, primero debe obtener uno de la entidad emisora de certificados externa que elija. Amplify Hosting admite dos tipos de certificados: RSA (Rivest-Shamir-Adleman) y ECDSA (Elliptic Curve Digital Signature Algorithm). Cada tipo de certificado debe cumplir los siguientes requisitos.

Certificados RSA

- Amplify Hosting admite claves RSA de 1024, 2048, 3072 y 4096 bits.
- AWS Certificate Manager (ACM) emite certificados RSA con claves de hasta 2048 bits.

- Para utilizar un certificado RSA de 3072 o 4096 bits, obtenga el certificado externamente e impórtelo a ACM. A continuación, estará disponible para su uso con Amplify Hosting.

Certificados ECDSA

- Amplify Hosting admite claves de 256 bits.
- Use la curva elíptica prime256v1 para obtener un certificado ECDSA para Amplify Hosting.

Tras obtener un certificado, impórtelo a. AWS Certificate Manager ACM es un servicio que le permite aprovisionar, administrar e implementar fácilmente certificados SSL/TLS públicos y privados para usarlos con los recursos internos conectados Servicios de AWS y con ellos. Asegúrese de solicitar o importar el certificado en la región del este de EE. UU. (norte de Virginia) (us-east-1).

Asegúrese de que su certificado personalizado cubra todos los subdominios que planea agregar. Puede usar un comodín al principio del nombre de dominio para incluir varios subdominios. Por ejemplo, si su dominio es `example.com`, puede incluir el dominio comodín `*.example.com`. Esto cubrirá subdominios como `product.example.com` y `api.example.com`.

Una vez que su certificado personalizado esté disponible en ACM, podrá seleccionarlo durante el proceso de configuración del dominio. Para obtener instrucciones sobre la importación de certificados en AWS Certificate Manager, consulte Importación de certificados a AWS Certificate Manager en la Guía del usuario de AWS Certificate Manager .

Si renueva o vuelve a importar su certificado personalizado en ACM, Amplify actualiza los datos del certificado asociados a su dominio personalizado. En el caso de los certificados importados, ACM no administra las renovaciones automáticamente. Es responsable de renovar sus certificados personalizados y de volver a importarlos.

Puede cambiar en cualquier momento el certificado que se usa en el dominio. Por ejemplo, puede cambiar del certificado administrado predeterminado a uno personalizado o cambiar de uno personalizado a uno administrado. Además, puede cambiar el certificado personalizado en uso por un certificado personalizado diferente. Para obtener instrucciones sobre cómo actualizar los certificados, consulte Update the SSL/TLS certificate for a domain.

Agregar un dominio personalizado administrado en Amazon Route 53

Amazon Route 53 es un servicio de DNS escalable y de alta disponibilidad. Para obtener más información, consulte Amazon Route 53 en la Guía para desarrolladores de Amazon Route 53. Si ya tiene un dominio de Route 53, siga las siguientes instrucciones para conectar el dominio personalizado a su aplicación de Amplify.

Para añadir un dominio personalizado gestionado por Route 53

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación que desea conectar a un dominio personalizado.
3. En el panel de navegación, elija Alojamiento y Dominios personalizados.
4. En la página Dominios personalizados, seleccione Agregar un dominio.
5. Ingrese el nombre de su dominio raíz. Por ejemplo, si el nombre de su dominio es https://example.com, introduzca**`example.com`**.

   A medida que comience a escribir, aparecen en la lista los dominios raíz que ya administra en Route 53. Puede elegir de la lista el dominio que quiera utilizar. Si aún no ha adquirido el dominio y este se encuentra disponible, puede comprarlo en Amazon Route 53.
6. Después de introducir el nombre de dominio, seleccione Configurar dominio.
7. Amplify crea automáticamente dos entradas de subdominio para su dominio de forma predeterminada. Por ejemplo, si tu nombre de dominio es example.com, verás los subdominios https://www.example.comy https://example.comuna redirección configurada del dominio raíz al subdominio www.

   (Opcional) Puede modificar la configuración predeterminada si desea añadir solo subdominios. Para cambiar la configuración predeterminada, elija Reescrituras y redireccionamientos en el panel de navegación y luego configure su dominio.
8. Seleccione el certificado SSL/TLS que desee utilizar. Puedes usar el certificado gestionado predeterminado que Amplify te proporciona o un certificado personalizado de terceros al que hayas importado. AWS Certificate Manager
   - Utilice el certificado predeterminado administrado por Amplify.
     - Elija Certificado administrado por Amplify.
   - Utilice un certificado personalizado de terceros.

a. Elija Certificado SSL personalizado).

b. Seleccione de la lista el certificado que desee utilizar.

9. Elija Añadir dominio.

> **Note**
>
> El DNS puede tardar hasta 24 horas en propagarse y emitir el certificado SSL. Si necesita ayuda para resolver posibles errores durante el proceso, consulte Solución de problemas de dominios personalizados.

Adición de un dominio personalizado administrado por un proveedor de DNS de terceros

Si no usa Amazon Route 53 para administrar su dominio, puede añadir un dominio personalizado gestionado por un proveedor de DNS externo a su aplicación implementada con Amplify.

Si lo está utilizando GoDaddy, consulte las instrucciones específicas the section called “Actualizar los registros DNS de un dominio administrado por GoDaddy” para este proveedor.

Para añadir un dominio personalizado administrado por un proveedor de DNS externo

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación a la que desea añadir un dominio personalizado.
3. En el panel de navegación, elija Alojamiento y Dominios personalizados.
4. En la página Dominios personalizados, seleccione Agregar un dominio.
5. Ingrese el nombre de su dominio raíz. Por ejemplo, si el nombre de su dominio es https://example.com, introduzca**`example.com`**.
6. Amplify detecta que no usa un dominio de Route 53 y le da la opción de crear una zona alojada en Route 53.
   - Creación de una zona alojada en Route 53

     a. Elija Crear zona alojada en Route 53.

     b. Elija Configurar dominio.

c. Los servidores de nombres de zonas alojadas se muestran en la consola. Vaya a la página web de su proveedor de DNS y agregue los servidores de nombres a la configuración de DNS.

d. Seleccione He agregado los servidores de nombres anteriores a mi registro de dominios.

e. Continúe con el paso siete.

- Para continuar con la configuración manual

  a. Elija Configuración manual

  b. Elija Configurar dominio.

  c. Continúe con el paso siete.

7. Amplify crea automáticamente dos entradas de subdominio para su dominio de forma predeterminada. Por ejemplo, si tu nombre de dominio es example.com, verás los subdominios https://www.example.comy https://example.comuna redirección configurada del dominio raíz al subdominio www.

   (Opcional) Puede modificar la configuración predeterminada si desea añadir solo subdominios. Para cambiar la configuración predeterminada, elija Reescrituras y redireccionamientos en el panel de navegación y configure su dominio.

8. Seleccione el certificado SSL/TLS que desee utilizar. Puedes usar el certificado gestionado predeterminado que Amplify te proporciona o un certificado personalizado de terceros al que hayas importado. AWS Certificate Manager

   - Utilice el certificado predeterminado administrado por Amplify.

     - Elija Certificado administrado por Amplify.

   - Utilice un certificado personalizado de terceros.

     a. Elija Certificado SSL personalizado).

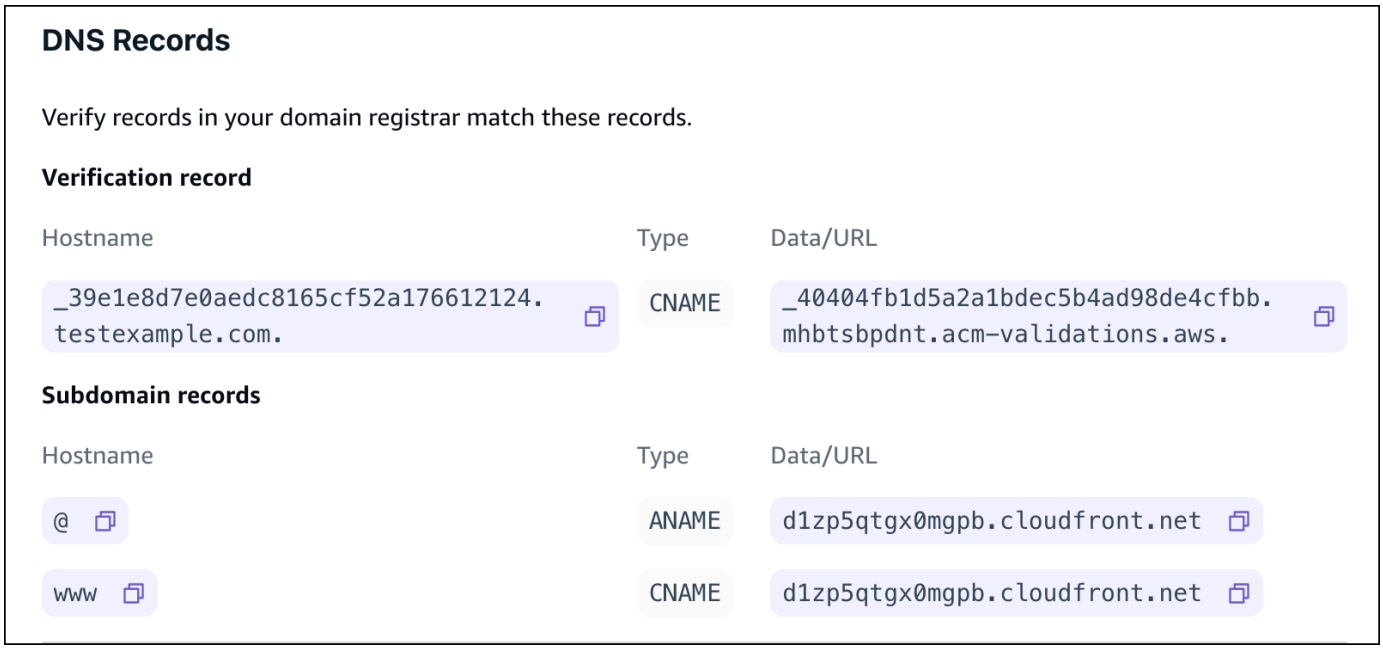
     b. Seleccione de la lista el certificado que desee utilizar.

9. Elija Añadir dominio.

10. Si eligió Crear zona alojada en Route 53 en el paso seis, continúe con el paso 15.

    Si ha elegido Configuración manual, en el paso seis debe actualizar los registros de DNS con su proveedor de dominios externo.

En el menú Acciones, elija Ver registros de DNS. La siguiente captura de pantalla muestra los registros de DNS mostrados en la consola.



11. Realice una de las siguientes acciones:

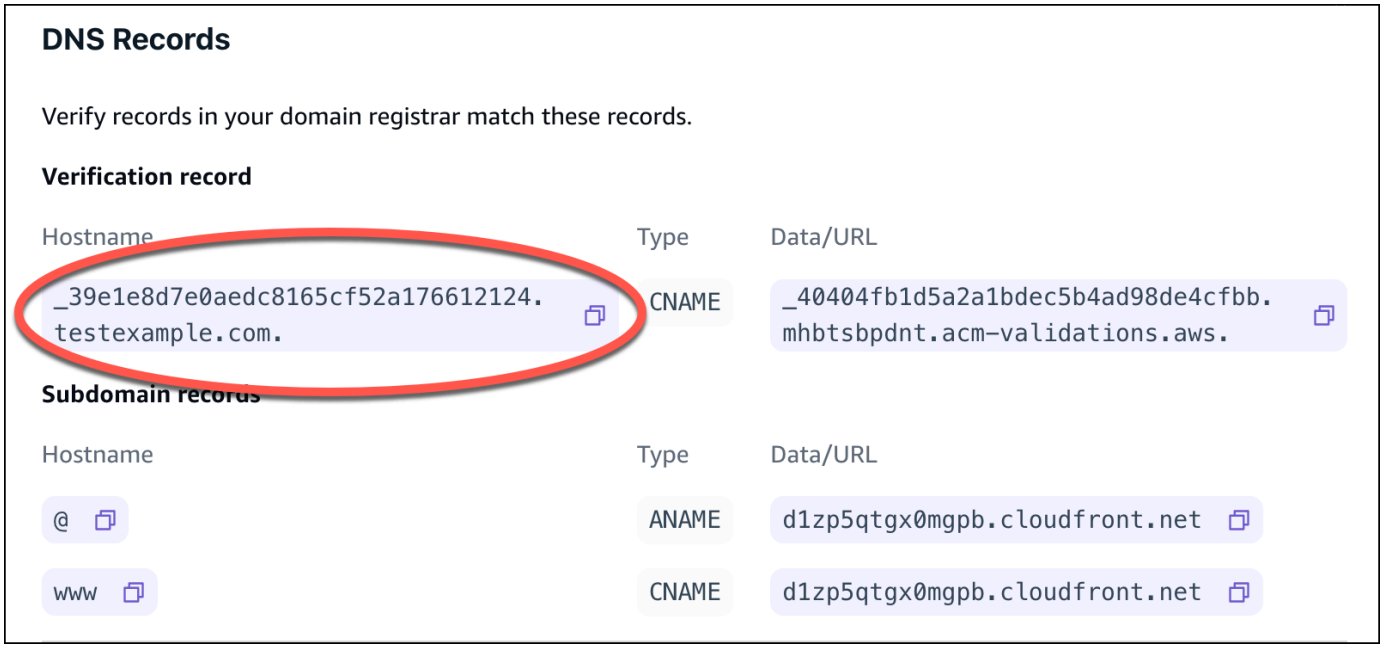
    - Si lo está utilizando GoDaddy, vaya aActualizar los registros DNS de un dominio administrado por GoDaddy.
    - Si usa otro proveedor de DNS externo, continúe con el siguiente paso de este procedimiento.

12. Acceda al sitio web de su proveedor de DNS, inicie sesión en su cuenta y busque la configuración de gestión de DNS de su dominio. Configurará dos registros CNAME.

13. Configura el primer registro CNAME para que dirija tu subdominio al servidor de AWS validación.

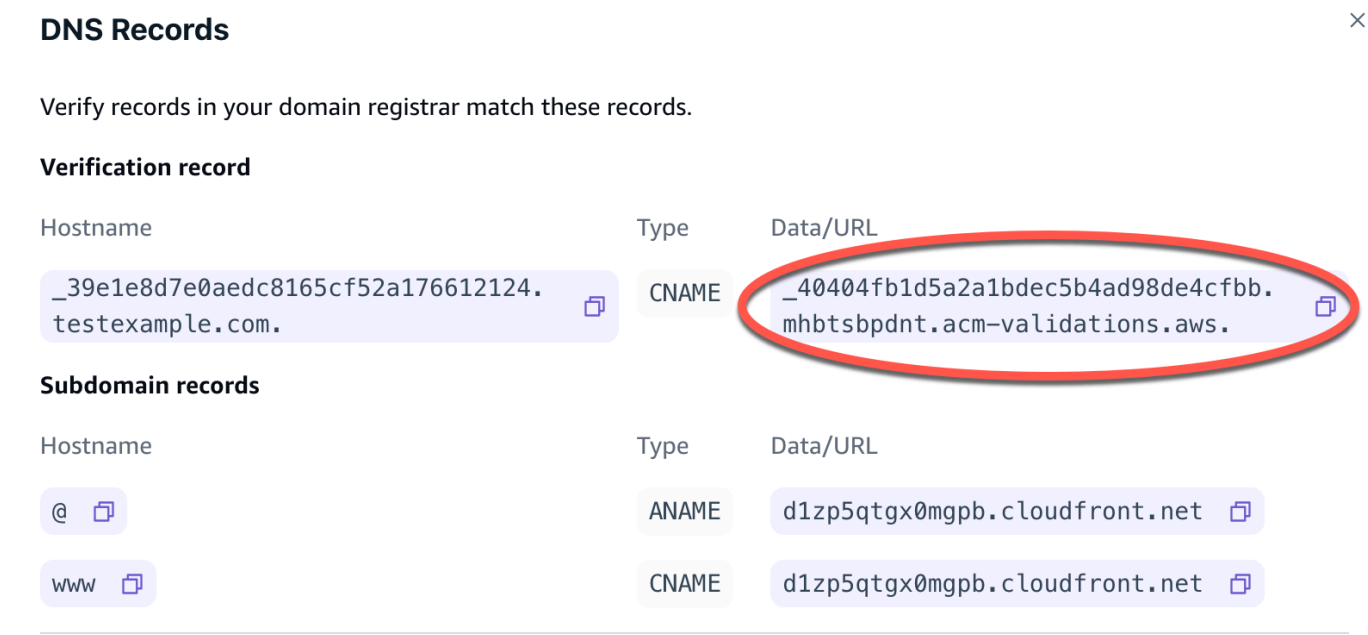
    Si la consola de Amplify muestra un registro de DNS para comprobar la propiedad de su subdominio, como _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, introduzca solo **`_c3e2d7eaf1e656b73f46cd6980fdc0e`** en el nombre de subdominio del registro CNAME.

    La siguiente captura de pantalla muestra la ubicación del registro de verificación que se debe utilizar.



Si la consola de Amplify muestra un registro del servidor de validación de ACM, como _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, introduzca **`_cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`** en el valor del registro CNAME.

La siguiente captura de pantalla muestra la ubicación del registro de verificación de ACM que se debe utilizar.



Amplify usa esta información para verificar la propiedad de su dominio y generar un certificado SSL/TLS para su dominio. Una vez que Amplify valide la propiedad de su dominio, todo el tráfico se servirá mediante HTTPS/2.

> **Note**
>
> El certificado Amplify predeterminado generado por AWS Certificate Manager (ACM) es válido durante 13 meses y se renueva automáticamente siempre que la aplicación esté alojada en Amplify. Si el registro de verificación del CNAME se ha modificado o eliminado, Amplify no podrá renovar el certificado. Deberá eliminar y volver a añadir el dominio en la consola de Amplify.

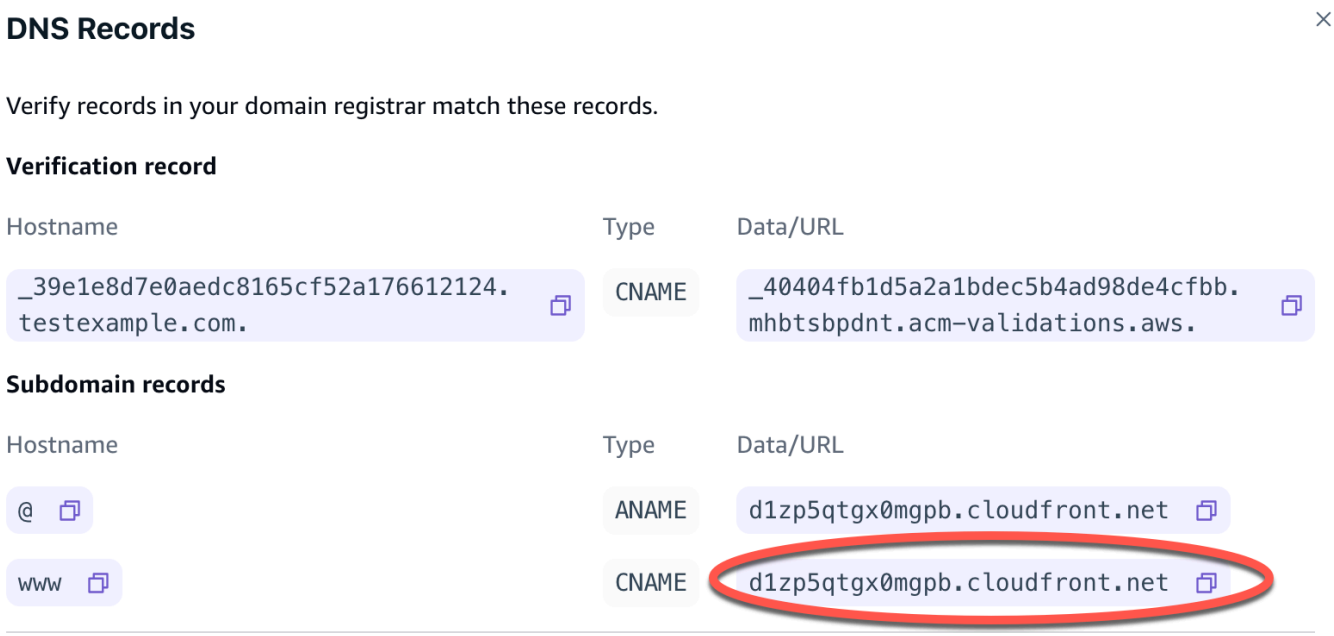
> **Important**
>
> Es importante que lleve a cabo este paso justo después de añadir su dominio personalizado en la consola de Amplify. El AWS Certificate Manager (ACM) comienza inmediatamente a intentar verificar la propiedad. Con el paso del tiempo, los controles serán menos frecuentes. Si agrega o actualiza sus registros CNAME unas horas después de haber creado la aplicación, es posible que la aplicación se quede atascada en el estado pendiente de verificación.

14. Configure un segundo registro CNAME para apuntar sus subdominios al dominio de Amplify. Por ejemplo, si su subdominio es www.ejemplo.com, introduzca www como nombre del subdominio.

    Si la consola de Amplify muestra el dominio de su aplicación como d111111abcdef8.cloudfront.net, introduzca **`d111111abcdef8.cloudfront.net`** en el dominio de Amplify.

    Si tiene tráfico de producción, se recomienda que actualice este registro CNAME una vez que el estado de su dominio sea DISPONIBLE en la consola de Amplify.

    La siguiente captura de pantalla muestra la ubicación del registro del nombre del dominio que se debe utilizar.



15. Configura el registro ANAME/ALIAS para que apunte al dominio raíz de tu aplicación (por ejemplo). https://example.com Un registro ANAME apunta la raíz de su dominio a un nombre de host. Si tiene tráfico de producción, se recomienda que actualice su registro ANAME una vez que el estado de su dominio sea DISPONIBLE en la consola. Para proveedores de DNS que no admiten ANAME/ALIAS, le recomendamos encarecidamente migrar su DNS a Route 53. Para obtener más información, consulte Configuración de Amazon Route 53 como servicio DNS.

> **Note**
>
> La verificación de la propiedad del dominio y la propagación de DNS para dominios de terceros puede tardar hasta 48 horas. Para resolver los posibles errores que puedan surgir, consulte Solución de problemas de dominios personalizados.

Actualizar los registros DNS de un dominio administrado por GoDaddy

Si GoDaddy es tu proveedor de DNS, sigue las siguientes instrucciones para actualizar tus registros de DNS en la GoDaddy interfaz de usuario y terminar de conectar la aplicación Amplify a tu GoDaddy dominio.

Para añadir un dominio personalizado gestionado por GoDaddy

1. Antes de poder actualizar los registros de DNS con ellos GoDaddy, complete los pasos uno a nueve del procedimientothe section called “Adición de un dominio personalizado administrado por un proveedor de DNS de terceros”.
2. Inicia sesión en tu GoDaddy cuenta.
3. En la lista de dominios, busque el dominio que desea agregar y elija Administrar DNS.
4. En la página DNS, GoDaddy muestra una lista de registros de tu dominio en la sección Registros DNS. Deberá añadir dos nuevos registros CNAME.
5. Cree el primer registro CNAME para apuntar sus subdominios al dominio de Amplify.

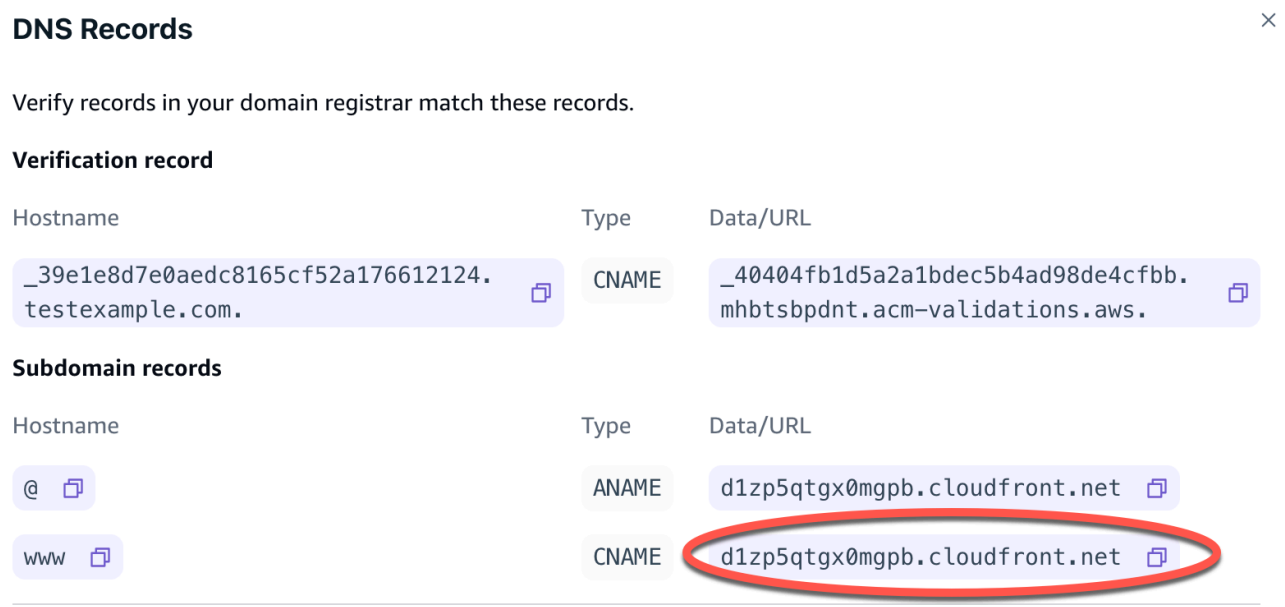
   a. En la sección Registro de DNS, elija Añadir un registro nuevo.

   b. En Tipo, elija CNAME.

   c. En Nombre, introduzca solo el subdominio. Por ejemplo, si el subdominio es www.ejemplo.com, introduzca www en Nombre.

   d. En Valor, consulte los registros de DNS en la consola de Amplify y, a continuación, introduzca el valor. Si la consola de Amplify muestra el dominio de su aplicación como d111111abcdef8.cloudfront.net, introduzca **`d111111abcdef8.cloudfront.net`** en Valor.

      La siguiente captura de pantalla muestra la ubicación del registro del nombre del dominio que se debe utilizar.



   e. Seleccione Guardar.

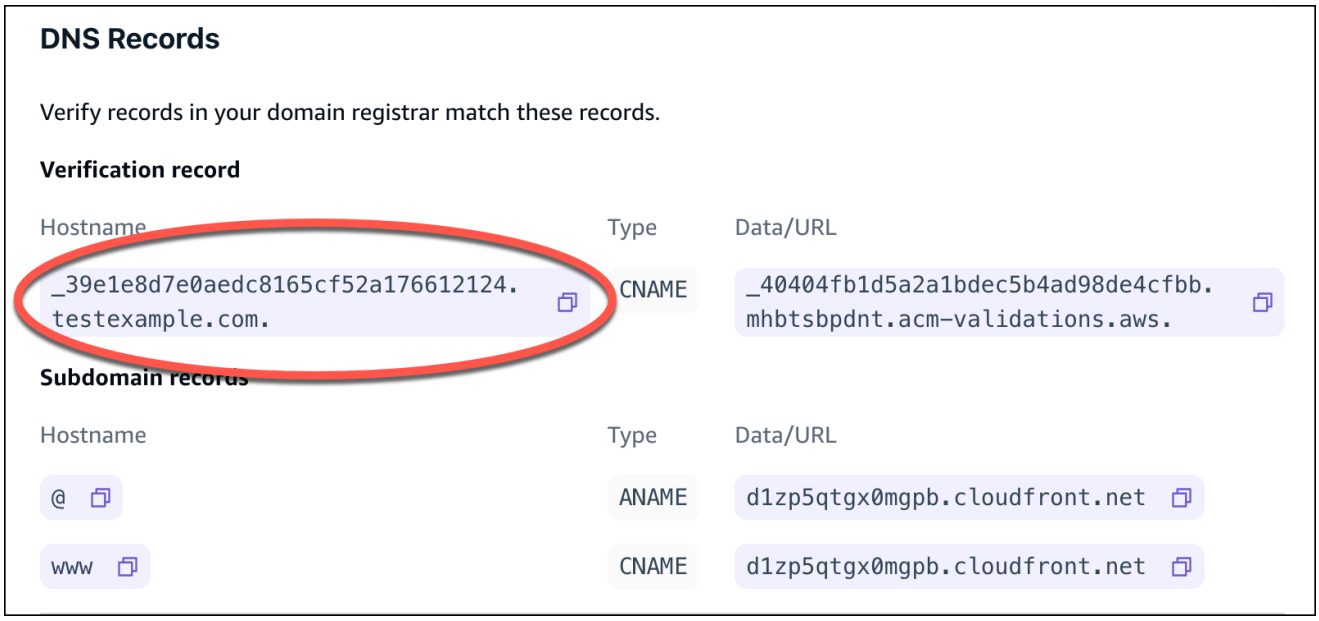
6. Crea el segundo registro CNAME para que apunte al servidor de validación AWS Certificate Manager (ACM). Un único ACM validado genera un certificado SSL/TLS para el dominio.

   a. En Tipo, elija CNAME.

   b. En Nombre, introduzca el subdominio.

      Por ejemplo, si el registro de DNS en la consola de Amplify para verificar la propiedad de su subdominio es _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, ingrese **`_c3e2d7eaf1e656b73f46cd6980fdc0e`** en Nombre.

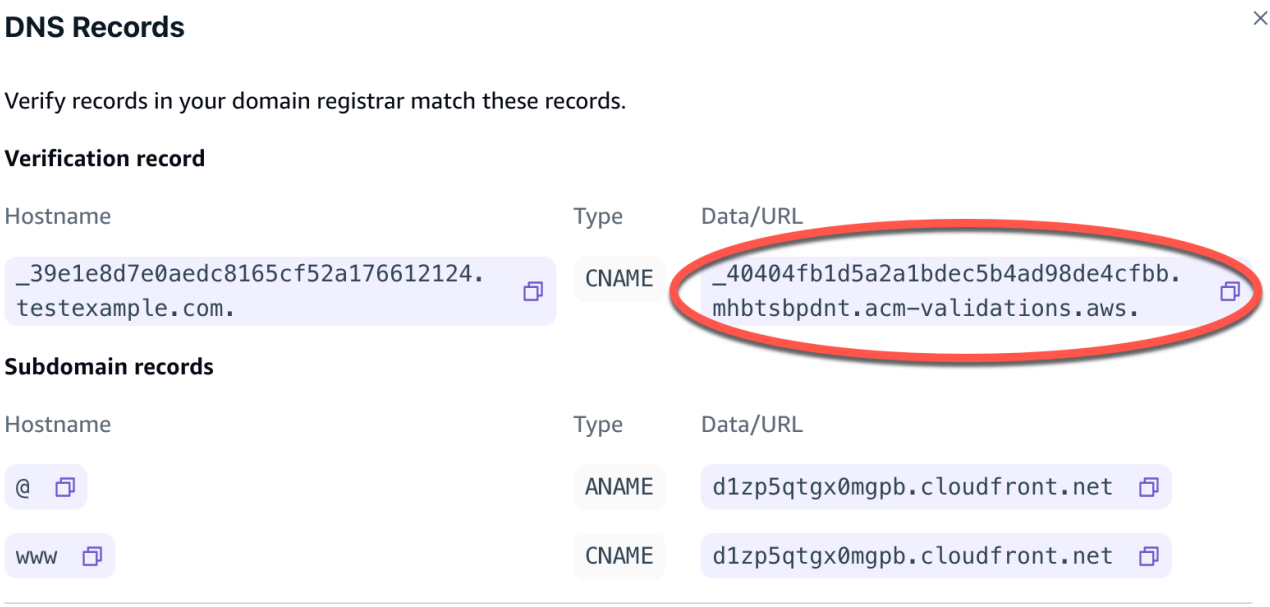
      La siguiente captura de pantalla muestra la ubicación del registro de verificación que se debe utilizar.



   c. En Valor, introduzca el certificado de validación ACM.

      Por ejemplo, si el servidor de validación es _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, introduzca _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws en Valor.

      La siguiente captura de pantalla muestra la ubicación del registro de verificación de ACM que se debe utilizar.



d. Seleccione Guardar.

> **Note**
>
> El certificado Amplify predeterminado generado por AWS Certificate Manager (ACM) es válido durante 13 meses y se renueva automáticamente siempre que la aplicación esté alojada en Amplify. Si el registro de verificación del CNAME se ha modificado o eliminado, Amplify no podrá renovar el certificado. Deberá eliminar y volver a añadir el dominio en la consola de Amplify.

7. Este paso no es obligatorio para los subdominios. GoDaddy no admite ANAME/ALIAS records. For DNS providers that do not have ANAME/ALIAS soporte, le recomendamos encarecidamente que migre su DNS a Amazon Route 53. Para obtener más información, consulte Configuración de Amazon Route 53 como servicio DNS.

   Si quieres seguir GoDaddy siendo tu proveedor y actualizar el dominio raíz, añade Forwarding y configura un reenvío de dominios:

   a. En la página DNS, ubique el menú que está en la parte superior y seleccione Reenvío.

   b. En la sección Dominio, seleccione Agregar reenvío.

   c. Elija http:// y, a continuación, introduzca el nombre del subdominio al que desea redireccionar (por ejemplo, www.ejemplo.com) en URL de destino.

   d. En Tipo de redirección, elija Temporal (302).

e. Elija Guardar.

Actualización del certificado SSL/TLS de un dominio

Puede cambiar en cualquier momento el certificado SSL/TLS que se usa en el dominio. Por ejemplo, puede pasar de usar un certificado administrado a uno personalizado. Esto resulta útil si quieres gestionar el certificado y sus notificaciones de caducidad. También puede cambiar el certificado personalizado que se utiliza en el dominio. Si realizas cambios en el certificado SSL, tu dominio activo no sufrirá ningún tiempo de inactividad. Para obtener más información sobre los certificados, consulte Using SSL/TLS certificates.

Utilice el siguiente procedimiento para actualizar el tipo de certificado o el certificado personalizado que se utiliza en un dominio.

Actualización de un certificado de dominio

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación que desee actualizar.
3. En el panel de navegación, elija Alojamiento y Dominios personalizados.
4. En la página Dominios personalizados, seleccione Configuración del dominio.
5. En la página de detalles del dominio, busque la sección Certificados SSL personalizados. El procedimiento para actualizar el certificado varía en función del tipo de cambio que desee llevar a cabo.
   - Cambio de un certificado personalizado al certificado predeterminado administrado por Amplify
     - Elija Certificado administrado por Amplify.
   - Cambio de un certificado administrado a uno personalizado
     a. Elija Certificado SSL personalizado).
     b. Seleccione de la lista el certificado que desee utilizar.
   - Para cambiar un certificado personalizado por un certificado personalizado diferente
     - En el Certificado SSL personalizado, seleccione de la lista el nuevo certificado que desee utilizar.

6. Seleccione Guardar. Los detalles del estado del dominio indicarán que Amplify ha iniciado el proceso de creación de SSL para un certificado administrado o el proceso de configuración de un certificado personalizado.

Administración de subdominios

El subdominio es la parte de la URL que aparece antes del nombre de dominio. Por ejemplo, www es el subdominio de www.amazon.com, y aws es el subdominio de aws.amazon.com. Si ya tiene un sitio web en producción, es posible que desee conectar solo un subdominio. Los subdominios también pueden ser multinivel. Por ejemplo, beta.alpha.ejemplo.com tiene el subdominio multinivel beta.alpha.

Para añadir solo un subdominio

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación a la que desea añadir un subdominio.
3. En el panel de navegación, elija Alojamiento y, a continuación, Dominios personalizados.
4. En la página Dominios personalizados, seleccione Agregar un dominio.
5. Introduzca el nombre de su dominio raíz y, a continuación, elija Configurar dominio. Por ejemplo, si el nombre de su dominio es https://example.com, introduzca example.com.
6. Elija Excluir raíz y modifique el nombre del subdominio. Por ejemplo, si el dominio es ejemplo.com, puede modificarlo para añadir el subdominio alpha.
7. Elija Añadir dominio.

Para añadir un subdominio multinivel

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación a la que desea agregar un subdominio multinivel.
3. En el panel de navegación, elija Alojamiento y, a continuación, Dominios personalizados.
4. En la página Dominios personalizados, seleccione Agregar un dominio.
5. Introduzca el nombre de un dominio con un subdominio, elija Excluir raíz y modifique el subdominio para agregar un nuevo nivel.

   Por ejemplo, si tiene un dominio llamado alpha.ejemplo.com y desea crear el subdominio multinivel beta.alpha.ejemplo.com, deberá introducir beta como valor de subdominio.

6. Elija Añadir dominio.

Para agregar o editar un subdominio

Tras añadir un dominio personalizado a una aplicación, puede editar un subdominio existente o añadir uno nuevo.

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación cuyos subdominios desea gestionar.
3. En el panel de navegación, elija Alojamiento y, a continuación, Dominios personalizados.
4. En la página Dominios personalizados, seleccione Configuración del dominio.
5. En Subdominios, puede editar los subdominios existentes.
6. (Opcional) Para agregar un nuevo subdominio, elija Agregar nuevo.
7. Seleccione Guardar.

Configuración de subdominios comodín

Amplify Hosting ya es compatible con subdominios comodín. Un subdominio comodín es un subdominio general que le permite apuntar los subdominios existentes y no existentes a una ramificación específica de la aplicación. Al usar un comodín para asociar todos los subdominios de una aplicación a una ramificación específica, puede ofrecer el mismo contenido a los usuarios de la aplicación en cualquier subdominio. También evita tener que configurar cada subdominio de forma individual.

Para crear un subdominio comodín, introduzca un asterisco (*) como nombre del subdominio. Por ejemplo, si introduce el subdominio comodín `*.example.com` para una ramificación específica de su aplicación, cualquier URL que termine en ejemplo.com se redirigirá a dicha ramificación. En este caso, las solicitudes de `dev.example.com` y `prod.example.com` se redirigirán al subdominio `*.example.com`.

Tenga en cuenta que Amplify solo admite subdominios comodín en dominios personalizados. No es posible usar esta característica con el dominio predeterminado `amplifyapp.com`.

Los subdominios comodín deben cumplir los siguientes requisitos:

- El nombre del subdominio debe especificarse únicamente con un asterisco (*).

- No puede utilizar un comodín para reemplazar parte de un nombre de subdominio, como este: *dominio.ejemplo.com.
- No puede sustituir un subdominio en el medio de un nombre de dominio, como este: subdominio.*.ejemplo.com.
- De forma predeterminada, todos los certificados aprovisionados por Amplify abarcan todos los subdominios de un dominio personalizado.

Para agregar o eliminar un subdominio comodín

Tras añadir un dominio personalizado a una aplicación, puede añadir un subdominio comodín a una ramificación de la aplicación.

1. Inicia sesión en la consola de Amplify Hosting AWS Management Console y ábrela.
2. Elija la aplicación cuyos subdominios comodín desea administrar.
3. En el panel de navegación, elija Alojamiento y, a continuación, Dominios personalizados.
4. En la página Dominios personalizados, seleccione Configuración del dominio.
5. En la sección Subdominios, puede agregar o eliminar subdominios comodín.
    - Para agregar un nuevo subdominio comodín
        a. Elija Add new (Añadir nuevo).
        b. En el subdominio, introduzca un *****.
        c. En la ramificación de la aplicación, seleccione un nombre de ramificación de la lista.
        d. Seleccione Guardar.
    - Para eliminar un subdominio comodín
        a. Elija Eliminar junto al nombre del subdominio. El tráfico al subdominio no configurado explícitamente se detiene y Amplify Hosting devuelve un código de estado 404 a dichas solicitudes.
        b. Seleccione Guardar.

Configuración de subdominios automáticos para un dominio personalizado de Amazon Route 53

Tras conectar una aplicación a un dominio personalizado en Route 53, Amplify le permite crear subdominios automáticamente para las ramificaciones recién conectadas. Por ejemplo, si conecta su ramificación de desarrollo, Amplify puede crear automáticamente dev.ejemplodominio.com. Al eliminar una ramificación, se eliminarán automáticamente todos los subdominios asociados.

Para configurar la creación automática de subdominios para ramificaciones recién conectadas

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija una aplicación conectada a un dominio personalizado gestionado en Route 53.
3. En el panel de navegación, elija Alojamiento y, a continuación, Dominios personalizados.
4. En la página Dominios personalizados, seleccione Configuración del dominio.
5. En la sección Creación automática de subdominios, active la característica.

> **Note**
>
> Esta característica solo está disponible para dominios raíz, por ejemplo, ejemplodominio.com. La consola de Amplify no mostrará esta casilla si su dominio es ya un subdominio, como dev.ejemplodominio.com.

Vistas previas de web con subdominios

Tras activar la creación automática de subdominios siguiendo las instrucciones anteriores, podrá acceder a las vistas previas web de las solicitudes de extracción de su aplicación mediante subdominios creados automáticamente. Cuando se cierra una solicitud de extracción, la ramificación y el subdominio asociados se eliminan automáticamente. Para obtener más información sobre cómo configurar las vistas previas web para las solicitudes de extracción, consulte Vistas previas web para solicitudes de extracción.

Solución de problemas de dominios personalizados

Si tiene algún problema al agregar un dominio personalizado a una aplicación en la consola de AWS Amplify , consulte Solución de problemas de dominios personalizados en el capítulo sobre

solución de problemas de Amplify. Si no ve una solución a su problema ahí, póngase en contacto con Support. Para obtener más información, consulte Creación de un caso de soporte en la Guía del usuario de AWS Support .

Ajuste de la configuración de compilación de una aplicación

Al implementar una aplicación, Amplify detecta automáticamente el marco de frontend y la configuración de compilación asociada al inspeccionar el archivo `package.json` de la aplicación en el repositorio de Git. Tiene las siguientes opciones de almacenamiento de configuración de compilación de la aplicación:

- Guarde la configuración de compilación en la consola de Amplify: la consola de Amplify detecta automáticamente la configuración de la compilación y la guarda de forma que se pueda acceder a través de la consola de Amplify. Amplify aplica esta configuración en todas las ramificaciones a menos que exista un archivo `amplify.yml` guardado en el repositorio.
- Guarde la configuración de compilación en el repositorio: descargue el archivo `amplify.yml` y añádalo a la raíz de su repositorio.

> **Note**
>
> La configuración de compilación está visible en el menú Alojamiento de la consola de Amplify solo cuando se configura una aplicación para una implementación continua y conectada a un repositorio de Git. Para obtener instrucciones sobre este tipo de implementación, consulte Introducción.

Descripción de la especificación de compilación

La especificación de compilación de una aplicación de Amplify es un conjunto de comandos de compilación y configuración de YAML relacionados que Amplify utiliza para ejecutar la compilación. En la siguiente lista se describen estas configuraciones y cómo se utilizan.

versión

El número de versión del archivo YAML de Amplify.

AppRoot

La ruta dentro del repositorio en el que reside esta aplicación. Se omite a menos que se definan varias aplicaciones.

env

Añada variables de entorno a esta sección. También puede añadir variables de entorno a través de la consola.

backend

Ejecute comandos de Amplify CLI para suministrar un backend, actualizar las funciones de Lambda, o los esquemas de GraphQL en el marco de una implementación continua.

frontend

Ejecute los comandos de compilación de frontend.

prueba

Ejecute comandos durante una fase de prueba. Aprenda a añadir pruebas a su aplicación.

fases de compilación

Tanto el frontend como el backend tienen tres fases que representan la ejecución de comandos durante cada secuencia de la compilación.

- preBuild: el script preBuild se ejecuta antes de que se inicie la compilación real, pero después de que Amplify instale dependencias.
- build: los comandos de compilación.
- postBuild: el script postBuild se ejecuta una vez que ha finalizado la compilación y Amplify ha copiado todos los artefactos necesarios en el directorio de salida.

buildpath

La ruta que se utilizará para ejecutar la compilación. Amplify utiliza esta ruta para localizar sus artefactos de compilación. Si no especifica una ruta, Amplify utiliza la raíz de la aplicación monorepo, por ejemplo. `apps/app`

artifacts>base-directory

El directorio en el que están los artefactos de compilación.

artifacts>files

Especifique los archivos de los artefactos que desee implementar. Introduzca `**/*` para incluir todos los archivos.

memoria caché

Especifica las dependencias en tiempo de compilación, como la carpeta node_modules. Durante la primera compilación, las rutas que se proporcionan aquí se almacenan en caché. En compilaciones posteriores, Amplify restaura la caché en las mismas rutas antes de ejecutar tus comandos.

Amplify considera que todas las rutas de caché proporcionadas son relativas a la raíz del proyecto. Sin embargo, Amplify no permite salir de la raíz del proyecto. Por ejemplo, si especificas una ruta absoluta, la compilación se realizará correctamente sin errores, pero la ruta no se almacenará en caché.

Referencia de la especificación de compilación de la sintaxis de YAML

En el siguiente ejemplo de especificación de compilación se muestra la sintaxis básica de YAML.

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

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

Edición de la especificación de compilación en la consola de Amplify

Puede personalizar la configuración de compilación de una aplicación al editar la especificación de compilación en la consola de Amplify. La configuración de compilación se aplica a todas las ramificaciones de la aplicación, excepto a las ramificaciones que tienen un archivo `amplify.yml` guardado en el repositorio de Git.

Edición de la configuración de compilación en la consola de Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación cuya configuración de compilación quiera modificar.
3. En el panel de navegación, elija Alojamiento y, a continuación, Configuración de compilación.

4. En la página Configuración de compilación, en la sección Configuración de compilación de la aplicación, elija Editar.
5. En la ventana Editar especificación de compilación, introduzca sus actualizaciones.
6. Seleccione Guardar.

Puede usar los ejemplos que se describen en los siguientes temas para actualizar la configuración de compilación de escenarios específicos.

Temas

Configuración de compilación específica de ramificación con scripting

Puede utilizar scripts del intérprete de comandos Bash para establecer la configuración de compilación específica de ramificación. Por ejemplo, el siguiente script usa la variable de entorno del sistema $ AWS_BRANCH para ejecutar un conjunto de comandos si el nombre de la rama es main y un conjunto de comandos diferente si el nombre de la rama es dev.

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

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

Configurar un comando para navegar a una subcarpeta

Para monorepos, los usuarios quieren poder utilizar `cd` en una carpeta para ejecutar la compilación. Después de ejecutar el comando `cd`, se aplica a todas las etapas de la compilación para que no sea necesario repetir el comando en fases independientes.

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

Implementación del backend con el frontend de una aplicación de Gen 1

> **Note**
>
> Esta sección se aplica únicamente a las aplicaciones de Amplify Gen 1. Se crea un backend de Gen 1 con Amplify Studio y la interfaz de la línea de comandos (CLI) de Amplify.

El comando `amplifyPush` es un script auxiliar que le ayuda con las implementaciones del backend. La configuración de compilación siguiente determina automáticamente el entorno de backend correcto que se va a implementar para la ramificación actual.

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

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

Configuración de la carpeta de salida

La siguiente configuración de compilación establece el directorio de salida en la carpeta pública.

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

Instalación de paquetes como parte de una compilación

Puede utilizar los comandos `npm` o `yarn` para instalar paquetes durante la compilación.

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

Uso de un registro npm privado

Puede añadir referencias a un registro privado en la configuración de compilación o añadirlo como variable de entorno.

```
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
```

Instalación de paquetes de SO

La imagen AL2 023 de Amplify ejecuta el código con un nombre de usuario sin privilegios. `amplify` Amplify otorga a este usuario privilegios para ejecutar comandos del sistema operativo mediante el comando `sudo` de Linux. Si desea instalar paquetes de sistema operativo para las dependencias que faltan, puede utilizar comandos como `yum` y `rpm` mediante `sudo`.

En la siguiente sección de compilación de ejemplos se muestra la sintaxis para instalar un paquete de sistema operativo mediante el comando `sudo`.

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

Configuración del almacenamiento clave-valor para todas las compilaciones

El `envCache` ofrece almacenamiento de un valor clave en el momento de la compilación. Los valores almacenados en `envCache` solo se pueden modificar durante una compilación y se pueden volver a utilizar durante la siguiente compilación. Mediante `envCache`, podemos almacenar información en el entorno implementado y hacer que esté disponible para el contenedor de compilación en compilaciones sucesivas. A diferencia de los valores almacenados en `envCache`, los cambios en las variables de entorno durante una compilación no se almacenan de forma persistente en futuras compilaciones.

Ejemplo de uso:

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

Omitir la compilación de una confirmación

Para omitir la compilación automática de una confirmación concreta, incluya el texto [skip-cd] al final del mensaje de confirmación.

Desactivar las compilaciones automáticas en cada confirmación

Puede configurar Amplify para desactivar las compilaciones automáticas en todas las confirmaciones de código. Para la configuración, elija Configuración de la aplicación y Configuración de ramificación y, a continuación, desplácese hasta la sección Ramificaciones con todas las ramificaciones conectadas. Elija una ramificación y, a continuación, elija Acciones y Deshabilitar compilación automática. Las nuevas confirmaciones en esa ramificación dejarán de iniciar una nueva compilación.

Configuración de la compilación e implementación de frontend basada en diferencias

Puede configurar Amplify para utilizar compilaciones de frontend basadas en diferencias. Si está habilitada, Amplify intentará ejecutar una diferencia en su `appRoot`, o en la carpeta `/src/` de forma predeterminada al inicio de cada compilación. Si Amplify no encuentra ninguna diferencia, omite la compilación de frontend, prueba (si se configura) e implementa los pasos y no actualiza la aplicación alojada.

Para configurar la compilación e implementación de frontend basada en diferencias

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea configurar la creación e implementación de frontend basada en diferencias.
3. En el panel de navegación, elija Alojamiento y Variables de entorno.
4. En la sección Variables de entorno, elija Administrar variables.
5. El procedimiento de configuración de la variable de entorno varía en función de si se habilita o deshabilita la creación e implementación de frontend basada en diferencias.
   - Para habilitar la creación e implementación de frontend basada en diferencias
     a. En la sección Administrar variables de Variable, introduzca `AMPLIFY_DIFF_DEPLOY`.
     b. En Valor, introduzca `true`.
   - Para deshabilitar la creación e implementación de frontend basada en diferencias
     - Lleve a cabo una de las siguientes acciones:
       - En la sección Administrar variables, busque `AMPLIFY_DIFF_DEPLOY`. En Valor, introduzca `false`.

- Elimine la variable de entorno `AMPLIFY_DIFF_DEPLOY`.

6. Elija Guardar.

Opcionalmente, puede configurar la variable de entorno `AMPLIFY_DIFF_DEPLOY_ROOT` para anular la ruta predeterminada con una ruta relativa a la raíz de su repositorio, como `dist`.

Configuración de compilaciones de backend basadas en diferencias para una aplicación de Gen 1

> **Note**
>
> Esta sección se aplica únicamente a las aplicaciones de Amplify Gen 1. Se crea un backend de Gen 1 con Amplify Studio y la interfaz de la línea de comandos (CLI) de Amplify.

Puede configurar Amplify Hosting para utilizar compilaciones de backend basadas en diferencias mediante la variable de entorno `AMPLIFY_DIFF_BACKEND`. Al habilitar las compilaciones de backend basadas en diferencias, al comienzo de cada compilación, Amplify intenta ejecutar una diferencia en la carpeta `amplify` de su repositorio. Si Amplify no encuentra ninguna diferencia, omitirá el paso de compilación del backend y no actualizará los recursos del backend. Si su proyecto no tiene la carpeta `amplify` en el repositorio, Amplify ignorará el valor `AMPLIFY_DIFF_BACKEND` de la variable de entorno.

Si actualmente tiene comandos personalizados especificados en la configuración de compilación de la fase de backend, las compilaciones de backend condicionales no funcionarán. Si desea que esos comandos personalizados se ejecuten, deberá moverlos a la fase de frontend de la configuración de compilación en el archivo `amplify.yml` de su aplicación.

Para configurar compilaciones de backend basadas en diferencias

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea configurar las compilaciones de backend basadas en diferencias.
3. En el panel de navegación, elija Alojamiento y Variables de entorno.
4. En la sección Variables de entorno, elija Administrar variables.

5. El procedimiento de configuración de la variable de entorno varía en función de si se habilitan o deshabilitan las compilaciones de backend basadas en diferencias.
   - Para habilitar las compilaciones de backend basadas en diferencias
     a. En la sección Administrar variables de Variable, introduzca `AMPLIFY_DIFF_BACKEND`.
     b. En Valor, introduzca `true`.
   - Para deshabilitar las compilaciones de backend basadas en diferencias
     - Lleve a cabo una de las siguientes acciones:
       - En la sección Administrar variables, busque `AMPLIFY_DIFF_BACKEND`. En Valor, introduzca `false`.
       - Elimine la variable de entorno `AMPLIFY_DIFF_BACKEND`.
6. Seleccione Guardar.

Modificar la configuración de compilación de monorepo

Cuando se almacenan varios proyectos o microservicios en un único repositorio, se denomina monorepo. Puede utilizar Amplify Hosting para implementar aplicaciones en un monorepo sin crear múltiples configuraciones de compilación o configuraciones de ramificación.

Amplify admite aplicaciones en monorepos genéricos, así como aplicaciones en monorepos creadas con npm workspace, pnpm workspace, Yarn workspace, Nx y Turborepo. Al implementar su aplicación, Amplify detecta automáticamente la herramienta de compilación de monorepo que está utilizando. Amplify aplica automáticamente la configuración de compilación a las aplicaciones en un npm workspace, Yarn workspace o Nx. Las aplicaciones Turborepo y pnpm requieren una configuración adicional. Para obtener más información, consulte Configuración de aplicaciones Turborepo y pnpm monorepo.

Puede guardar la configuración de compilación de un monorepo en la consola de Amplify o descargar el archivo de `amplify.yml` y añadirlo a la raíz de su repositorio. Amplify aplica la configuración guardada en la consola a todas tus ramificaciones, a menos que encuentre un archivo de `amplify.yml` en su repositorio. Cuando hay un archivo de `amplify.yml`, su configuración anula cualquier configuración de compilación guardada en la consola de Amplify.

Referencia de la especificación de compilación de la sintaxis de YAML para monorepo

La sintaxis de YAML de una especificación de compilación de monorepo es diferente de la sintaxis de YAML de un repositorio que contiene una sola aplicación. En el caso de un monorepo, se declara cada proyecto en una lista de aplicaciones. Debe proporcionar la siguiente clave `appRoot` adicional para cada aplicación que declare en la especificación de compilación de monorepo:

appRoot

La raíz, dentro del repositorio, en la que se inicia la aplicación. Esta clave debe existir y tener el mismo valor que la variable de entorno `AMPLIFY_MONOREPO_APP_ROOT`. Para obtener instrucciones sobre cómo configurar esta variable de entorno, consulteConfiguración de la variable de entorno AMPLIFY_MONOREPO_APP_ROOT.

En el siguiente ejemplo de especificación de compilación de monorepo se muestra cómo declarar varias aplicaciones de Amplify en el mismo repositorio. Las dos aplicaciones, `react-app` y `angular-app`, se declaran en la lista `applications`. La clave `appRoot` de cada aplicación indica que la aplicación se encuentra en la carpeta raíz `apps` del repositorio.

El atributo `buildpath` se configura en `/` para ejecutar y compilar la aplicación desde la raíz del proyecto monorepo. El `baseDirectory` atributo es la ruta relativa de`buildpath`.

Sintaxis de YAML de especificación de compilación de monorepo

```
version: 1
applications:
  - appRoot: apps/react-app
    env:
      variables:
        key: value
    backend:
      phases:
        preBuild:
          commands:
            - *enter command*
        build:
          commands:
            - *enter command*
        postBuild:
```

```
          commands:
          - *enter command*
    frontend:
      buildPath: / # Run install and build from the monorepo project root
      phases:
        preBuild:
          commands:
            - *enter command*
            - *enter command*
        build:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        discard-paths: yes
        baseDirectory: location
      cache:
        paths:
            - path
            - path
    test:
      phases:
        preTest:
          commands:
            - *enter command*
        test:
          commands:
            - *enter command*
        postTest:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        configFilePath: *location*
        baseDirectory: *location*
  - appRoot: apps/angular-app
    env:
      variables:
        key: value
    backend:
```

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

```
        configFilePath: *location*
        baseDirectory: *location*
```

Una aplicación que utilice el siguiente ejemplo de especificación de compilación se compilará en la raíz del proyecto y los artefactos de compilación se ubicarán en `/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
```

Configuración de la variable de entorno AMPLIFY_MONOREPO_APP_ROOT

Al implementar una aplicación almacenada en un monorepo, la variable de entorno `AMPLIFY_MONOREPO_APP_ROOT` de la aplicación debe tener el mismo valor que la ruta de la raíz de la aplicación, en relación con la raíz de su repositorio. Por ejemplo, un monorepo denominado `ExampleMonorepo` con una carpeta raíz denominada `apps`, que contenga `app1`, `app2` y `app3`, tiene la siguiente estructura de directorios:

```
ExampleMonorepo
  apps
    app1
    app2
    app3
```

En este ejemplo, el valor de la variable de entorno `AMPLIFY_MONOREPO_APP_ROOT` para `app1` es `apps/app1`.

Al implementar una aplicación monorepo mediante la consola de Amplify, la consola establece automáticamente la variable de entorno `AMPLIFY_MONOREPO_APP_ROOT` con el valor que especifique para la ruta a la raíz de la aplicación. Sin embargo, si su aplicación monorepo ya existe en Amplify o se implementa utilizando AWS CloudFormation, debe configurar manualmente la variable de entorno en la sección Variables de **`AMPLIFY_MONOREPO_APP_ROOT`** entorno de la consola de Amplify.

Configuración automática de la variable de entorno AMPLIFY_MONOREPO_APP_ROOT durante la implementación

Las siguientes instrucciones muestran cómo implementar una aplicación monorepo con la consola de Amplify. Amplify establece automáticamente la variable de entorno `AMPLIFY_MONOREPO_APP_ROOT` mediante la carpeta raíz de la aplicación que especifique en la consola.

Para implementar una aplicación monorepo con la consola de Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija Crear nueva aplicación en la esquina superior derecha.
3. En la página Comenzar a crear con Amplify, seleccione el proveedor de Git y, a continuación, elija Siguiente.
4. En la página Añadir ramificación de repositorio, haga lo siguiente:
    a. Elija el nombre del repositorio de la lista.
    b. Elija el nombre de la ramificación que desea usar.
    c. Seleccione Mi aplicación es un monorepo
    d. Introduzca la ruta a su aplicación en su monorepo, por ejemplo, **`apps/app1`**.
    e. Elija Next (Siguiente).
5. En la página de Configuración de la aplicación, puede utilizar la configuración predeterminada o personalizar la configuración de compilación de su aplicación. En la sección Variables de entorno, Amplify establece `AMPLIFY_MONOREPO_APP_ROOT` en la ruta que se especificó en el paso 4d.
6. Elija Next (Siguiente).
7. En la página Revisar, elija Guardar e implementar.

Configuración de la variable de entorno AMPLIFY_MONOREPO_APP_ROOT para una aplicación existente

Siga las siguientes instrucciones para configurar manualmente la variable de `AMPLIFY_MONOREPO_APP_ROOT` entorno de una aplicación que ya esté implementada en Amplify o que se haya creado con ella. CloudFormation

Para configurar la variable de entorno AMPLIFY_MONOREPO_APP_ROOT para una aplicación existente

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija el nombre de la aplicación para la que desea establecer la variable de entorno.
3. En el panel de navegación, elija Alojamiento y, a continuación, Variables de entorno.
4. En Variables de entorno, elija Administrar variables.
5. En la sección Gestionar variables, haga lo siguiente:
   a. Elija Add new (Añadir nuevo).
   b. En Variable, introduzca la clave `AMPLIFY_MONOREPO_APP_ROOT`.
   c. En Valor, introduzca la ruta a la aplicación, por ejemplo **`apps/app1`**.
   d. En Ramificación, Amplify aplica de forma predeterminada la variable de entorno a todas las ramificaciones.
6. Elija Guardar.

Configuración de aplicaciones Turborepo y pnpm monorepo

Las herramientas de compilación Turborepo y pnpm workspace monorepo obtienen información de configuración de los archivos `.npmrc`. Al implementar una aplicación monorepo creada con una de estas herramientas, debe tener un archivo `.npmrc` en el directorio raíz del proyecto.

En el archivo `.npmrc`, configure el enlazador para la instalación de los paquetes de Node en `hoisted`. Puede copiar la siguiente línea en su archivo.

```
node-linker=hoisted
```

Para obtener más información sobre los archivos *`.npmrc`* y la configuración, consulte pnpm .npmrc en la documentación de pnpm.

Pnpm no se incluye en el contenedor de compilación predeterminado de Amplify. En el caso de las aplicaciones pnpm workspace y Turborepo, debe añadir un comando para instalar pnpm en la fase `preBuild` de configuración de compilación de la aplicación.

El siguiente extracto de ejemplo de una especificación de compilación muestra una fase `preBuild` con un comando para instalar pnpm.

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

Implementaciones de ramificaciones de características y flujos de trabajo de equipo

Amplify Hosting está diseñado para funcionar con ramas de funciones y GitFlow flujos de trabajo. Amplify usa las ramificaciones de Git para crear una nueva implementación cada vez que conecta una nueva ramificación en su repositorio. Después de conectar la primera ramificación, se crean ramificaciones de características adicionales.

Para agregar una ramificación a una aplicación

1. Elija la aplicación a la que desea agregar una ramificación.
2. Seleccione Configuración de la aplicación y, a continuación, Configuración de ramificación.
3. En la página Configuración de ramificación, seleccione Agregar ramificación.
4. Seleccione una ramificación de su repositorio.
5. Seleccione Agregar ramificación.
6. Vuelva a implementar la aplicación.

Después de agregar una sucursal, tu aplicación tiene dos implementaciones disponibles en los dominios predeterminados de Amplify, como https://main.appid.amplifyapp.comy. https://dev.appid.amplifyapp.com Esto puede variar team-to-team, pero normalmente la sucursal principal rastrea el código de lanzamiento y es la rama de producción. La ramificación de desarrollo se usa como ramificación de integración para probar nuevas características. Esto permite que los evaluadores beta puedan probar características que todavía no se han publicado en la implementación de ramificaciones de desarrollo, sin que ello afecte a ningún usuario final de producción en la implementación de ramificaciones principales.

Temas

- Flujos de trabajo en equipo con aplicaciones full stack de Amplify Gen 2
- Flujos de trabajo en equipo con aplicaciones full stack de Amplify Gen 1
- Implementaciones de ramificaciones de características basadas en patrones
- Generación automática de configuración de Amplify en tiempo de compilación (solo para aplicaciones de Gen 1)
- Compilaciones de backend condicionales (solo para aplicaciones de Gen 1)

- Use los backends de Amplify en todas las aplicaciones (solo aplicaciones de Gen 1)

Flujos de trabajo en equipo con aplicaciones full stack de Amplify Gen 2

AWS Amplify Gen 2 presenta una experiencia de desarrollador TypeScript basada en el código para definir los backends. Para obtener más información sobre los flujos de trabajo completos con las aplicaciones de Amplify Gen 2, consulte Fullstack workflows en Amplify Docs.

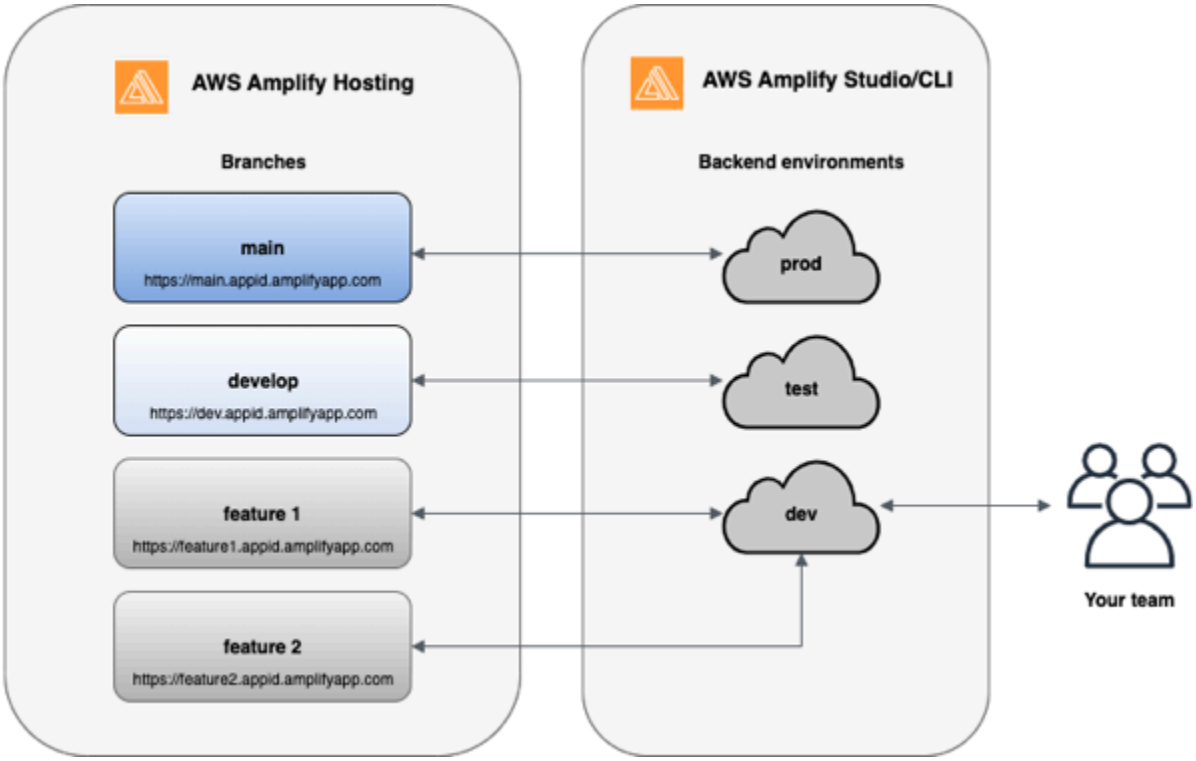
Flujos de trabajo en equipo con aplicaciones full stack de Amplify Gen 1

La implementación de una ramificación de características consta de un entorno frontend y otro backend opcional. El frontend se compila e implementa en una red de entrega de contenido (CDN), mientras que Amplify Studio o la CLI de Amplify implementan el backend en AWS. Para obtener información sobre cómo configurar este escenario de implementación, consulte Creación del backend de una aplicación.

Amplify Hosting implementa continuamente recursos de backend, como las funciones de GraphQL APIs y Lambda, con sus despliegues de sucursales de funciones. Puede usar los siguientes modelos de ramificación para implementar su backend y frontend con Amplify Hosting.

Flujo de trabajo de ramificación de característica

- Cree entornos de backend deproducción, pruebas y desarrollo con Amplify Studio o la CLI de Amplify.
- Asigne el backend de producción a la ramificación principal.
- Asigne el backend de pruebas a la ramificación de desarrollo.
- Los miembros del equipo pueden usar el entorno de backend de desarrollo para probar ramificaciones de características individuales.



1. Instale la CLI de Amplify para inicializar un nuevo proyecto de Amplify.

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

2. Inicialice un entorno de backend de producción para su proyecto. Si no tienes un proyecto, crea uno con herramientas de arranque como Gatsby. create-react-app

```
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. Añada entornos de backend de pruebas y desarrollo.

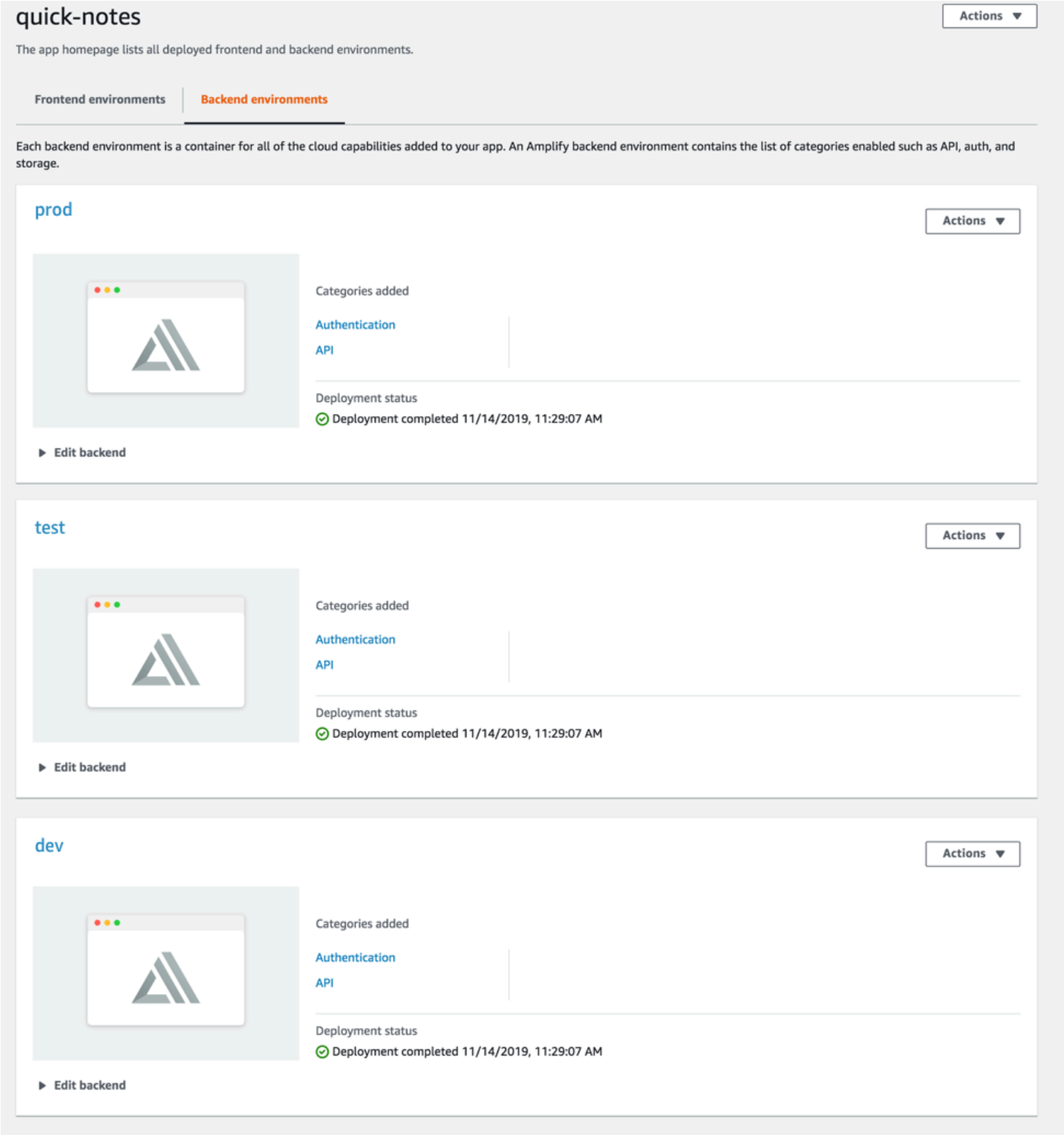
```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: test
...
amplify push
```

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

4. Inserte código en el repositorio de Git que elija (en este ejemplo supondremos que lo ha insertado en el principal).

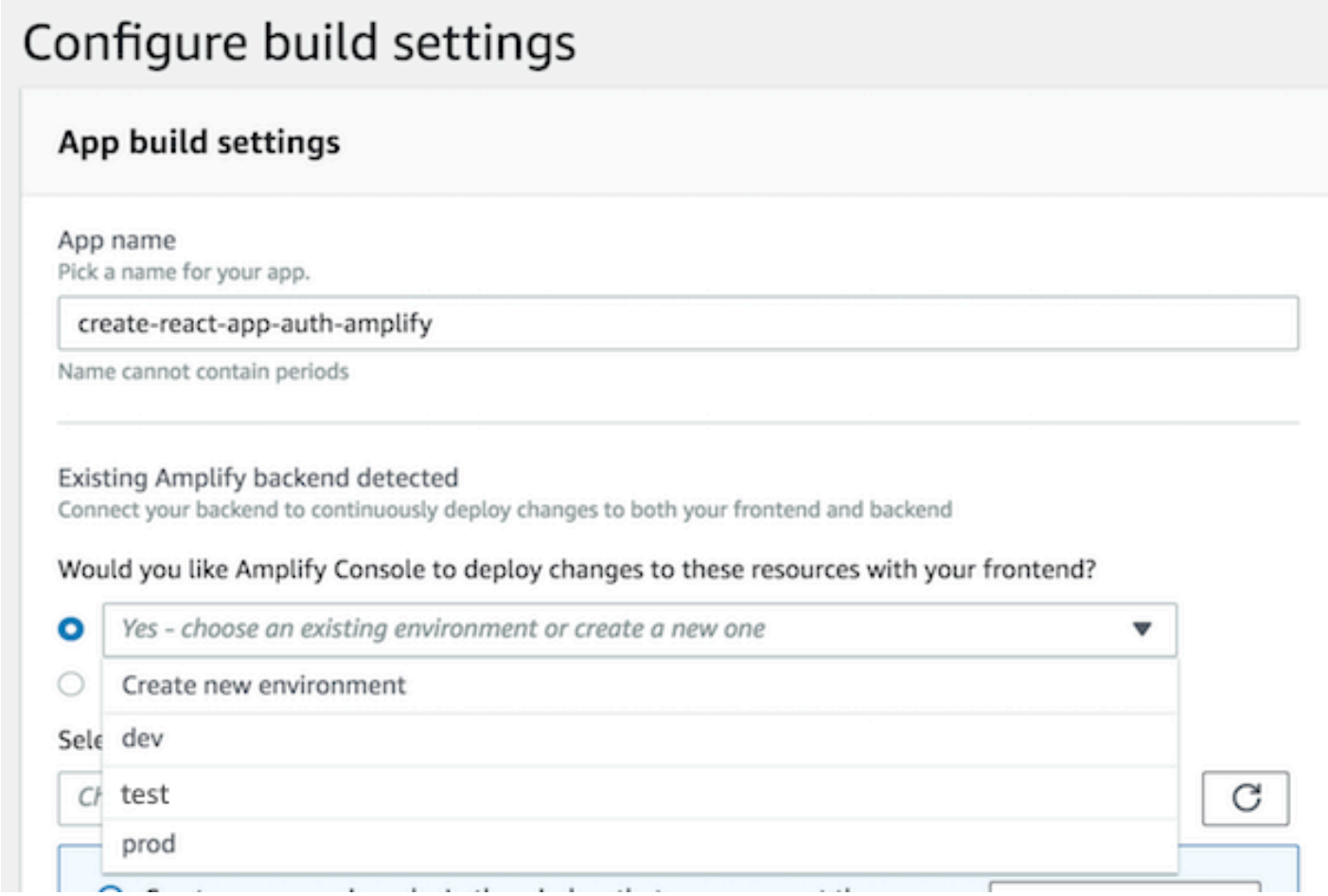
```
git commit -am 'Added dev, test, and prod environments'
git push origin main
```

5. Visite Amplify en AWS Management Console para ver su entorno de backend actual. Desplácese un nivel hacia arriba desde la ruta de navegación para ver una lista de todos los entornos de backend creados en la pestaña de Entornos de backend.

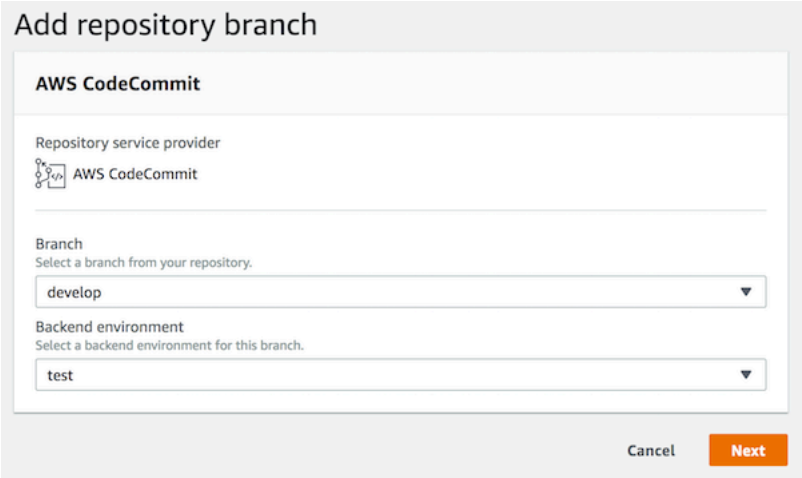


6. Pase a la pestaña de Entornos de frontend y conecte su proveedor de repositorios con la ramificación principal.
7. En la página de configuración de compilación, seleccione un entorno de backend existente para configurar la implementación continua con la ramificación principal. Elija producción en la

lista y conceda el rol de servicio a Amplify. Elija Guardar e implementar. Una vez completada la compilación, obtendrá una implementación en la sucursal principal disponible en. https://main.appid.amplifyapp.com



8. Conecte la ramificación de desarrollo en Amplify (suponga que las ramificaciones de desarrollo y principales son las mismas en este momento). Elija el entorno de backend de pruebas.



9. Amplify ya está configurado. Puede empezar a trabajar en nuevas características en una ramificación de características. Añada la funcionalidad de backend mediante el entorno de backend de desarrollo desde su estación de trabajo local.

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

10. Una vez que termine de trabajar en la característica, confirme su código y cree una solicitud de extracción para realizar una revisión interna.

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

11. Para obtener una vista previa de cómo se verán los cambios, vaya a la consola de Amplify y conecte su ramificación de características. Nota: Si lo tiene AWS CLI instalado en su sistema (no en la CLI de Amplify), puede conectar una sucursal directamente desde su terminal. Encontrará su appid yendo a App settings > General > AppARN (Configuración de la aplicación > General > 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. Podrás acceder a tu función en https://newinternet.appid.amplifyapp.compara compartirla con tus compañeros de equipo. Si todo parece correcto, combine las relaciones públicas en la ramificación de desarrollo.

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

13. Esto dará inicio a una compilación que actualizará el backend y el frontend de Amplify con un despliegue en una sucursal en. https://dev.appid.amplifyapp.com Puede compartir este enlace con las partes interesadas internas para que puedan revisar la nueva característica.

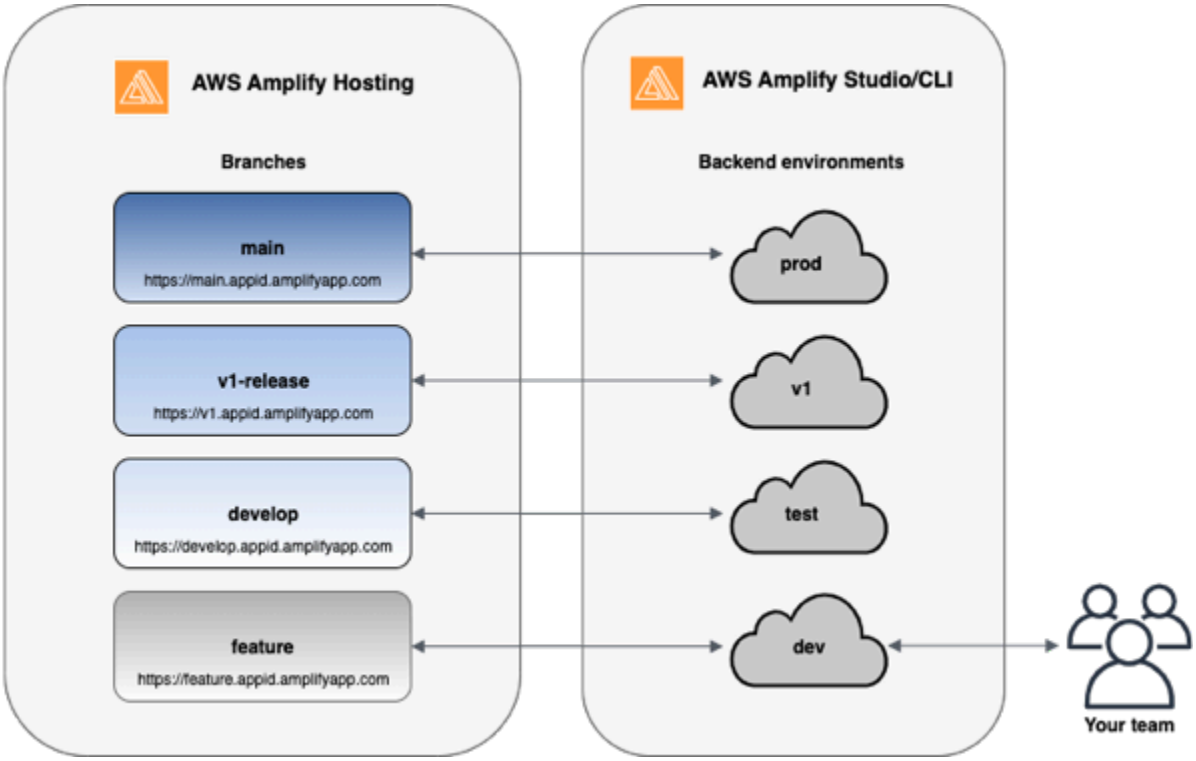
14. Elimine su ramificación de características de Git, Amplify y quite el entorno de backend de la nube (siempre puede poner en marcha uno nuevo basado en la ejecución de “amplify env checkout prod” y de “amplify env add”).

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

GitFlow flujo de trabajo

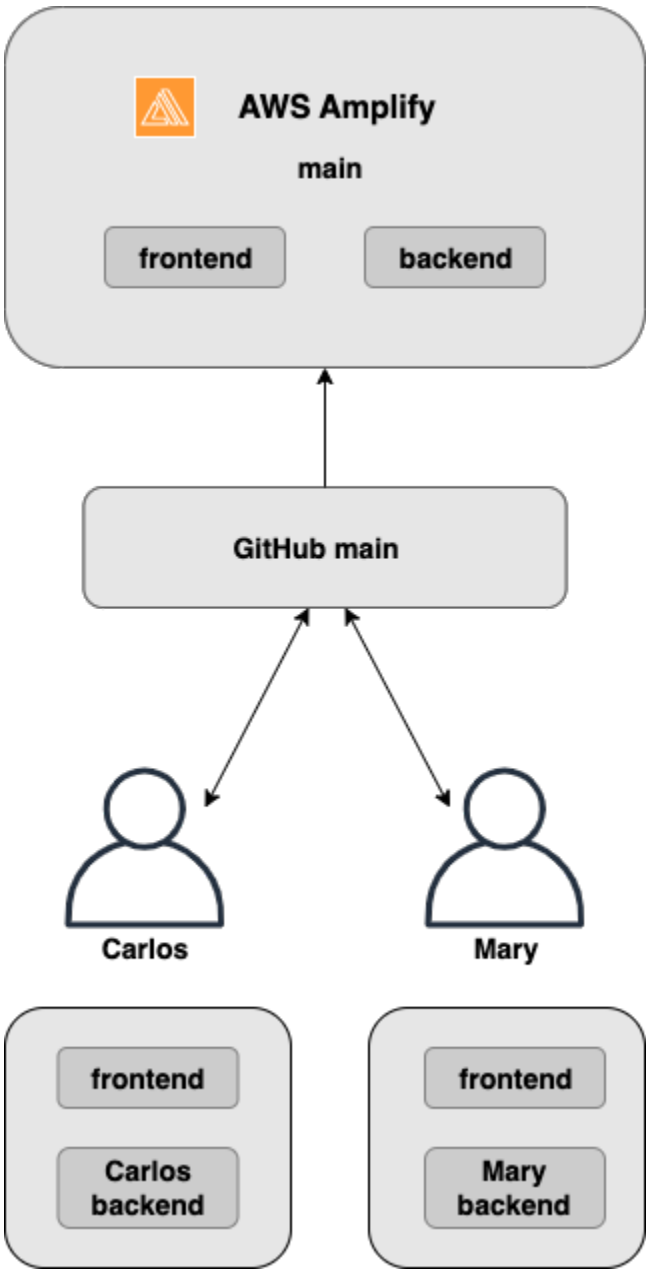
GitFlow utiliza dos ramas para registrar el historial del proyecto. La rama principal solo rastrea el código de la versión, y la rama de desarrollo se usa como rama de integración para las nuevas funciones. GitFlow simplifica el desarrollo paralelo al aislar el desarrollo nuevo del trabajo terminado. El nuevo desarrollo (como correcciones de características y de errores no urgentes) se lleva a cabo en las ramificaciones de características. Cuando el desarrollador considera que el código está listo para lanzamiento, la ramificación de características se combina de nuevo en la ramificación de desarrollo de integración. Las únicas confirmaciones en la ramificación principal son combinaciones desde ramificaciones de lanzamiento y ramificaciones de correcciones (para corregir errores de emergencia).

El siguiente diagrama muestra una configuración recomendada con. GitFlow Puede seguir el mismo proceso tal como se describe en la sección sobre el flujo de trabajo de ramificación de característica anterior.



Entorno de pruebas por desarrollador

- Cada desarrollador de un equipo crea un entorno de pruebas en la nube que es independiente de su equipo local. Esto permite a los desarrolladores trabajar de forma aislada entre sí sin sobrescribir los cambios de los demás miembros del equipo.
- Cada ramificación de Amplify tiene su propio backend. Esto garantiza que Amplify utilice el repositorio de Git como una fuente única desde la que implementar cambios, en lugar de confiar a los desarrolladores del equipo que envíen manualmente su backend o frontend a producción desde sus equipos locales.



1. Instale la CLI de Amplify para inicializar un nuevo proyecto de Amplify.

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

2. Inicialice el entorno de backend mary para su proyecto. Si no tienes un proyecto, crea uno con herramientas de bootstrap como create-react-app 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. Inserte código en el repositorio de Git que elija (en este ejemplo supondremos que lo ha insertado en el principal).

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

4. Conecte su repositorio > principal a Amplify.
5. La consola de Amplify detectará los entornos de backend que ha creado la CLI de Amplify. Elija Crear nuevo entorno en el menú desplegable y conceda el rol de servicio a Amplify. Elija Guardar e implementar. Una vez completada la compilación, dispondrás de una implementación en la sucursal principal https://main.appid.amplifyapp.comcon un nuevo entorno de backend vinculado a la sucursal.
6. Conecte la ramificación de desarrollo en Amplify (suponga que las ramificaciones de desarrollo y principales son las mismas en este momento) y elija Crear.

Implementaciones de ramificaciones de características basadas en patrones

Las implementaciones de ramificaciones basadas en patrones le permiten implementar automáticamente ramificaciones que coinciden con un patrón específico en Amplify. Los equipos de productos que utilizan la rama de funciones o los GitFlow flujos de trabajo para sus lanzamientos ahora pueden definir patrones, por ejemplo, **`release**`** implementar automáticamente las ramas de Git que comienzan con «release» en una URL que se puede compartir.

1. Seleccione Configuración de la aplicación y, a continuación, Configuración de ramificación.
2. En la página de configuración de Branch, selecciona Editar.
3. Seleccione Detección automática de ramificaciones para conectar automáticamente a Amplify las ramificaciones que coincidan con un conjunto de patrones.
4. En el cuadro Detección automática de ramificaciones: patrones, introduzca los patrones para implementar automáticamente las ramificaciones.
   - **`*`**: implementará todas las ramificaciones en su repositorio.
   - **`release*`**: implementará todas las ramificaciones que comiencen con la palabra “release”.

- **`release*/`**: implementará todas las ramificaciones que coincidan con un patrón “release /”.
- Especifique varios patrones en una lista separados por comas. Por ejemplo, **`release*, feature*`**.

5. Seleccione Detección automática de ramificaciones: control de acceso para configurar la protección de contraseñas automática para todas las ramificaciones creadas automáticamente.
6. En el caso de las aplicaciones de Gen 1 compiladas con un backend de Amplify, puede elegir crear un nuevo entorno para cada ramificación conectada o apuntar todas las ramificaciones a un backend existente.
7. Seleccione Guardar.

Implementación de ramificaciones de características basadas en patrones para una aplicación conectada a un dominio personalizado

Puede usar implementaciones de ramificaciones de características basadas en patrones para una aplicación conectada a un dominio personalizado de Amazon Route 53.

- Para obtener más información sobre la configuración de implementaciones de ramificaciones de características basadas en patrones, consulte Configuración de subdominios automáticos para un dominio personalizado de Amazon Route 53
- Para obtener más información sobre cómo conectar una aplicación de Amplify a un dominio personalizado gestionado en Route 53, consulte Agregar un dominio personalizado administrado en Amazon Route 53
- Para obtener más información sobre el funcionamiento de Route 53, consulte Qué es Amazon Route 53.

Generación automática de configuración de Amplify en tiempo de compilación (solo para aplicaciones de Gen 1)

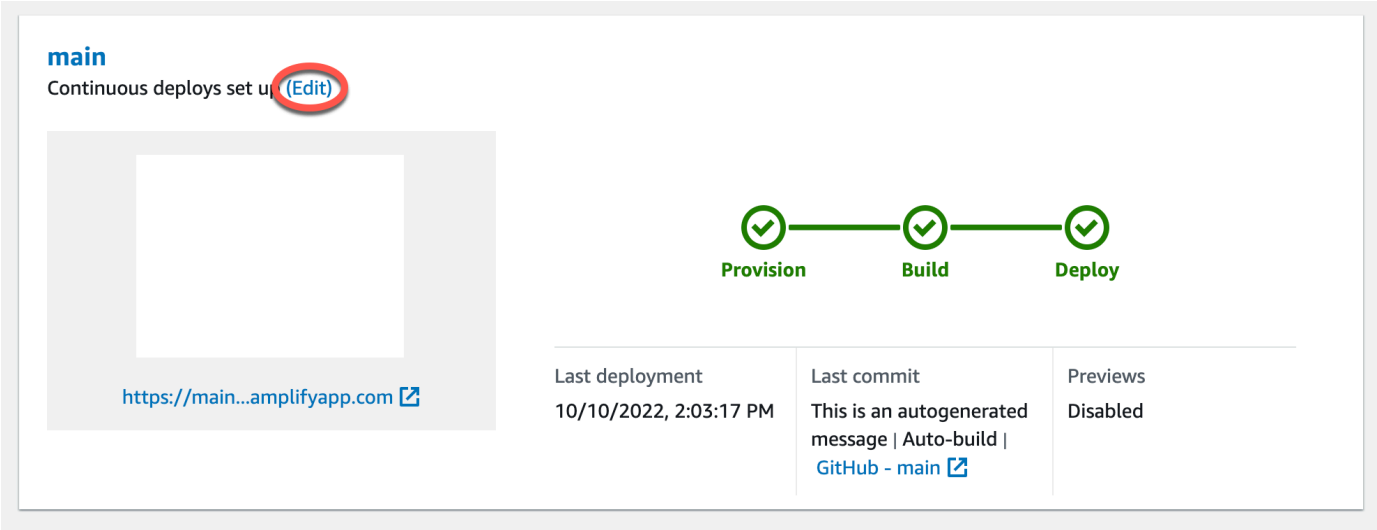
> **Note**
>
> La información de esta sección es solo para aplicaciones de Gen 1. Si quiere implementar automáticamente los cambios en la infraestructura y el código de las aplicaciones desde

las ramificaciones de características de una aplicación de Gen 2, consulte Fullstack branch deployments en Amplify Docs.

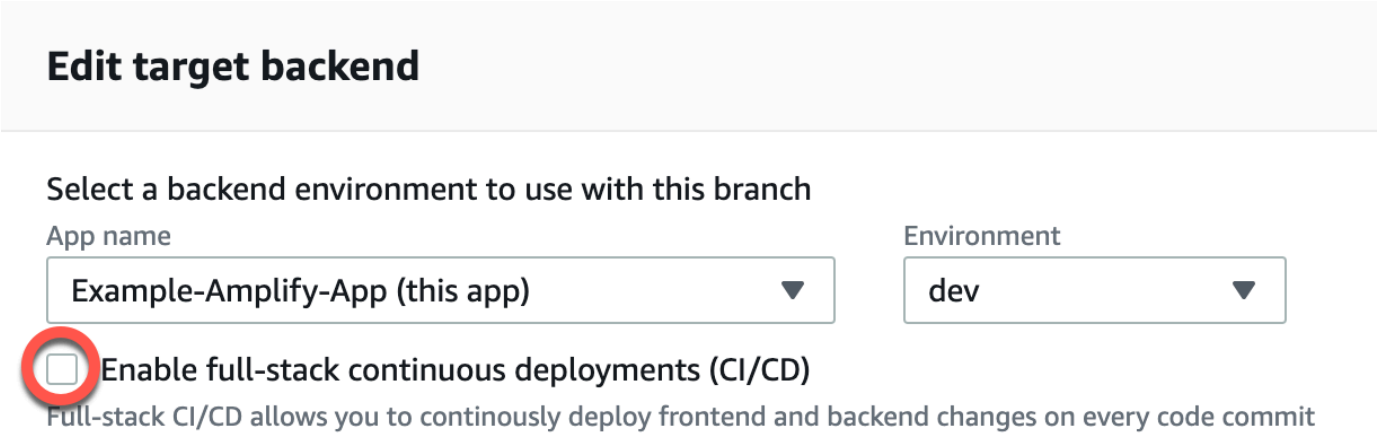
En el caso de las aplicaciones Gen1, Amplify permite generar automáticamente el archivo `aws-exports.js` de configuración de Amplify en tiempo de compilación. Al desactivar las implementaciones de CI/CD de pila completa, permite que su aplicación genere automáticamente el archivo `aws-exports.js` y garantiza que no se lleven a cabo actualizaciones en el backend durante el tiempo de compilación.

Para generar automáticamente el archivo **`aws-exports.js`** en tiempo de compilación

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación que desea editar.
3. Elija la pestaña Entornos de alojamiento.
4. Busque la ramificación que desea editar y eligir Editar.



5. En la página Editar backend de destino, desmarque Habilitar implementaciones continuas de pila completa (CI/CD) para desactivar el CI/CD de pila completa en este backend.



6. Seleccione un rol de servicio existente para conceder a Amplify los permisos necesarios para modificar el backend de su aplicación. Si necesita crear un rol de servicio, elija Crear un rol. Para obtener más información sobre cómo crear un rol de servicio, consulte Añadir un rol de servicio con permisos para implementar recursos de backend.
7. Elija Guardar. Amplify aplicará estos cambios la próxima vez que compile la aplicación.

Compilaciones de backend condicionales (solo para aplicaciones de Gen 1)

> **Note**
>
> La información de esta sección es solo para aplicaciones de Gen 1. Amplify Gen 2 presenta una experiencia de desarrollador TypeScript basada en el código. Por lo tanto, esta característica no es necesaria para los backends de Gen 2.

Amplify admite compilaciones de backend condicionales en todas las ramificaciones de una aplicación de Gen 1. Para configurar las compilaciones de backend condicionales, defina la variable del entorno `AMPLIFY_DIFF_BACKEND` como `true`. Habilitar las compilaciones de backend condicionales ayudará a acelerar aquellas compilaciones en las que solo se realicen cambios en el frontend.

Cuando habilite las compilaciones de backend basadas en diferencias, Amplify intentará ejecutar una diferencia en la carpeta `amplify` de su repositorio al inicio de cada compilación. Si Amplify no encuentra ninguna diferencia, omitirá el paso de compilación del backend y no actualizará los

recursos del backend. Si su proyecto no tiene la carpeta `amplify` en el repositorio, Amplify ignorará el valor `AMPLIFY_DIFF_BACKEND` de la variable de entorno. Para obtener más información sobre cómo configurar la variable de entorno `AMPLIFY_DIFF_BACKEND`, consulte Configuración de compilaciones de backend basadas en diferencias para una aplicación de Gen 1.

Si actualmente tiene comandos personalizados especificados en la configuración de compilación de la fase de backend, las compilaciones de backend condicionales no funcionarán. Si desea que esos comandos personalizados se ejecuten, deberá moverlos a la fase de frontend de la configuración de compilación en el archivo `amplify.yml` de su aplicación. Para obtener más información acerca de la actualización del archivo `amplify.yml`, consulte Descripción de la especificación de compilación.

Use los backends de Amplify en todas las aplicaciones (solo aplicaciones de Gen 1)

> **Note**
>
> La información de esta sección es solo para aplicaciones de Gen 1. Si quiere compartir recursos de backend para una aplicación de Gen 2, consulte Share resources across branches en Amplify Docs

Amplify le permite reutilizar los entornos de backend existentes en todas sus aplicaciones de Gen 1 de una determinada región. Puede hacerlo al crear una nueva aplicación, al conectar una nueva ramificación a una aplicación existente o al actualizar un frontend existente para que apunte a un entorno de backend distinto.

Reutilice backends para crear una nueva aplicación

Para reutilizar un backend al crear una nueva aplicación en Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Para crear el nuevo backend que usaremos en este ejemplo, haga lo siguiente:
   a. En el panel de navegación, elija Todas las aplicaciones.
   b. Elija Nueva aplicación, Crear una aplicación.
   c. Escriba un nombre para su aplicación, como **`Example-Amplify-App`**.
   d. Elija Confirmar implementación.

3. Para conectar un frontend a su nuevo backend, elija la pestaña Entornos de alojamiento.
4. Elija su proveedor de git y, a continuación, elija Conectar ramificación.
5. En la página Añadir ramificación de repositorio, elija el nombre de su repositorio en Repositorios actualizados recientemente. En Ramificación, seleccione la ramificación de su repositorio para conectarla.
6. En la página Configuración de compilaciones, haga lo siguiente:

   a. En Nombre de aplicación, seleccione la aplicación que desea usar para agregar un entorno de backend. Puede elegir la aplicación actual o cualquier otra aplicación de la región actual.

   b. En Entorno, seleccione el nombre del entorno de backend que desea añadir. Puede usar un entorno existente o crear uno nuevo.

   c. De forma predeterminada, la pila completa CI/CD is turned off. Turning off full-stack CI/CD hace que la aplicación se ejecute en modo de solo extracción. En el momento de la compilación, Amplify generará automáticamente el archivo `aws-exports.js` sin modificar el entorno de backend.

   d. Seleccione un rol de servicio existente para conceder a Amplify los permisos necesarios para modificar el backend de su aplicación. Si necesita crear un rol de servicio, elija Crear un rol. Para obtener más información sobre cómo crear un rol de servicio, consulte Añadir un rol de servicio con permisos para implementar recursos de backend.

   e. Elija Siguiente.
7. Elija Guardar e implementar.

Reutilice los backends al conectar una ramificación a una aplicación existente

Para reutilizar un backend al conectar una ramificación a una aplicación de Amplify existente

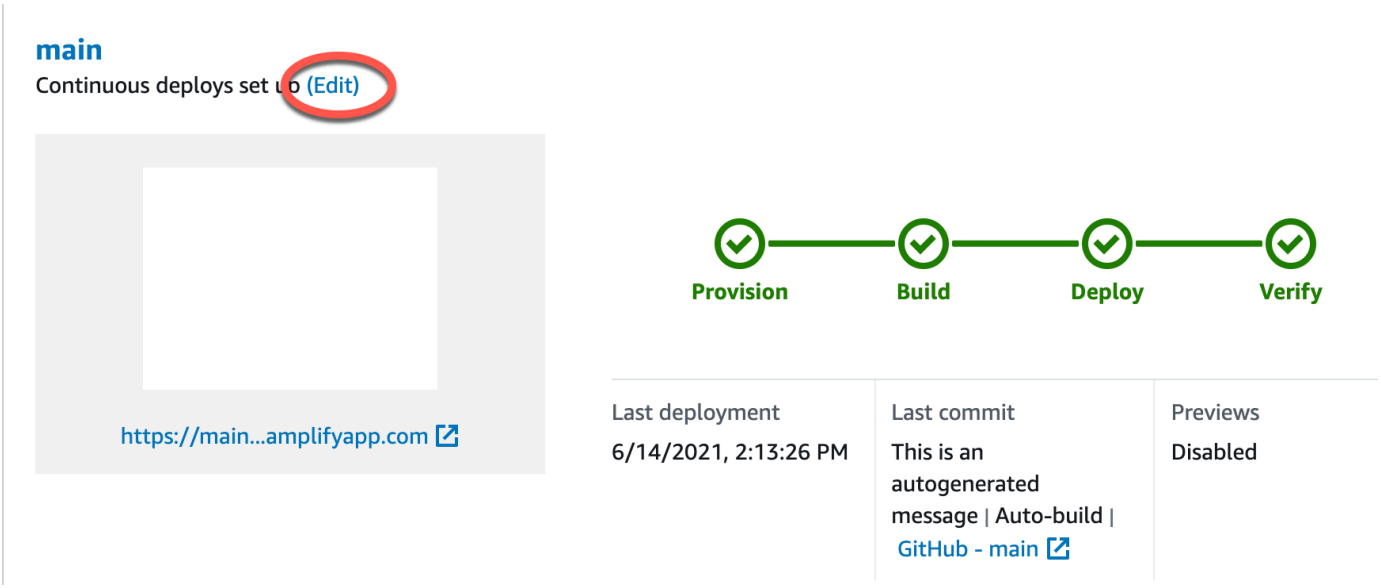
1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación a la que desea conectar una nueva ramificación.
3. En el panel de navegación, elija Configuración de la aplicación y General.
4. En la sección Ramificaciones, elija Conectar una ramificación.
5. En la página Añadir ramificación de repositorio, en Ramificación, seleccione la ramificación de su repositorio a la que desea conectar.

6. En Nombre de aplicación, seleccione la aplicación que desea usar para agregar un entorno de backend. Puede elegir la aplicación actual o cualquier otra aplicación de la región actual.
7. En Entorno, seleccione el nombre del entorno de backend que desea añadir. Puede usar un entorno existente o crear uno nuevo.
8. Si tiene que configurar un rol de servicio para conceder a Amplify los permisos necesarios para realizar cambios en el backend de su aplicación, la consola se lo solicitará. Para obtener más información sobre cómo crear un rol de servicio, consulte Añadir un rol de servicio con permisos para implementar recursos de backend.
9. De forma predeterminada, la pila completa CI/CD is turned off. Turning off full-stack CI/CD hace que la aplicación se ejecute en modo de solo extracción. En el momento de la compilación, Amplify generará automáticamente el archivo `aws-exports.js` sin modificar el entorno de backend.
10. Elija Siguiente.
11. Elija Guardar e implementar.

Edite un frontend existente para que apunte a un backend distinto

Para editar una aplicación de frontend de Amplify de modo que apunte a un backend distinto

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación cuyo backend desea editar.
3. Elija la pestaña Entornos de alojamiento.
4. Busque la ramificación que desea editar y eligir Editar.



5. En la página Seleccione el entorno de backend a usar con esta ramificación, en Nombre de la aplicación, seleccione la aplicación de frontend para la que quiere editar el entorno de backend. Puede elegir la aplicación actual o cualquier otra aplicación de la región actual.
6. En Entorno de backend, seleccione el nombre del entorno de backend que desea añadir.
7. De forma predeterminada, la pila completa CI/CD is enabled. Uncheck this option to turn off full-stack CI/CD for this backend. Turning off full-stack CI/CD hace que la aplicación se ejecute en modo de solo extracción. En el momento de la compilación, Amplify generará automáticamente el archivo `aws-exports.js` sin modificar el entorno de backend.
8. Elija Guardar. Amplify aplicará estos cambios la próxima vez que compile la aplicación.

Creación del backend de una aplicación

Con él AWS Amplify , puede crear una aplicación completa con alojamiento de datos, autenticación, almacenamiento y frontend que se despliega en ella. AWS

AWS Amplify Gen 2 presenta una experiencia de desarrollador TypeScript basada en el código para definir los backends. Para obtener información sobre cómo usar Amplify Gen 2 para crear y conectar un backend a su aplicación, consulte Build & connect backend en Amplify Docs.

Si busca la documentación para crear el backend de una aplicación de primera generación mediante la CLI y Amplify Studio, consulte Build & connect backend en Amplify Docs para Gen 1.

Temas

Creación de un backend de una aplicación de Gen 2

Para ver un tutorial que lo guía a través de los pasos para crear una aplicación completa de Amplify Gen 2 con un backend TypeScript basado, consulte Comenzar en los documentos de Amplify.

Creación de un backend de una aplicación de Gen 1

En este tutorial podrá configurar un flujo de CI/CD de pila completa con Amplify. Implementará una aplicación frontend en Amplify Hosting. A continuación, creará un backend con Amplify Studio. Y, finalmente, conectará el backend en la nube a la aplicación frontend.

Requisitos previos

Antes de comenzar este tutorial, complete los siguientes requisitos previos.

Inscríbase en un Cuenta de AWS

> Si aún no es AWS cliente, debe crear una Cuenta de AWS siguiendo las instrucciones en línea. Al registrarte, podrás acceder a Amplify y a otros AWS servicios que puedes usar con tu aplicación.

Creación de un repositorio de Git

Amplify admite GitHub Bitbucket y. GitLab AWS CodeCommit Envíe su aplicación en su repositorio de Git.

Instalación de la interfaz de la línea de comandos (CLI) de Amplify

Para obtener más instrucciones, consulte Instalar la CLI de Amplify en la Documentación de Amplify Framework.

Paso 1: implementar un frontend

Si ya tiene una aplicación frontend en un repositorio de git y quiere usarla en este tutorial, proceda con las instrucciones para implementar una aplicación de frontend.

Si necesita crear una nueva aplicación frontend para usarla en este ejemplo, puede seguir las instrucciones Create React App en la documentación de Create React App.

Para implementar una aplicación frontend

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, elija Nueva aplicación y, a continuación, Alojar aplicación web en la esquina superior derecha.
3. Selecciona tu proveedor GitHub, Bitbucket o AWS CodeCommit repositorio y GitLab, a continuación, selecciona Continuar.
4. Amplify autorizará el acceso a su repositorio de git. Para GitHub los repositorios, Amplify ahora usa GitHub la función Aplicaciones para autorizar el acceso de Amplify.

   Para obtener más información sobre la instalación y autorización de la GitHub aplicación, consulte. Configuración del acceso de Amplify a los repositorios GitHub
5. En la página Añadir ramificación de repositorio, siga estos pasos:

   a. En la lista de Repositorios actualizados recientemente, seleccione el nombre del repositorio que desea conectar.

   b. En la lista de Ramificaciones, seleccione el nombre de la ramificación del repositorio que desea conectar.

   c. Elija Siguiente.
6. En la página Configurar los ajustes de compilación, elija Siguiente.

7. En la página Revisar, elija Guardar e implementar. Una vez completada la implementación, podrá ver su aplicación en el dominio predeterminado `amplifyapp.com`.

> **Note**
>
> Para aumentar la seguridad de las aplicaciones de Amplify, el dominio amplifyapp.com se ha registrado en la lista de sufijos públicos (PSL). Para una mayor seguridad, le recomendamos que utilice cookies con un prefijo `__Host-` si alguna vez necesita configurar cookies confidenciales en el nombre de dominio predeterminado de las aplicaciones de Amplify. Esta práctica le ayudará a proteger su dominio de los intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de configuración de cookies en la red de desarrolladores de Mozilla.

Paso 2: crear un backend

Ahora que ha implementado una aplicación frontend en Amplify Hosting, puede crear un backend. Siga estas instrucciones para crear un backend con una base de datos simple y un punto de conexión de API GraphQL.

Para crear un backend

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, seleccione la aplicación que ha creado en el Paso 1.
3. En la página de inicio de la aplicación, elija la pestaña Entornos de backend y, a continuación, elija Comenzar. Se iniciará el proceso de configuración en un entorno de ensayo predeterminado.
4. Una vez finalizada la configuración, elija Launch Studio para acceder al entorno de backend de ensayo de Amplify Studio.

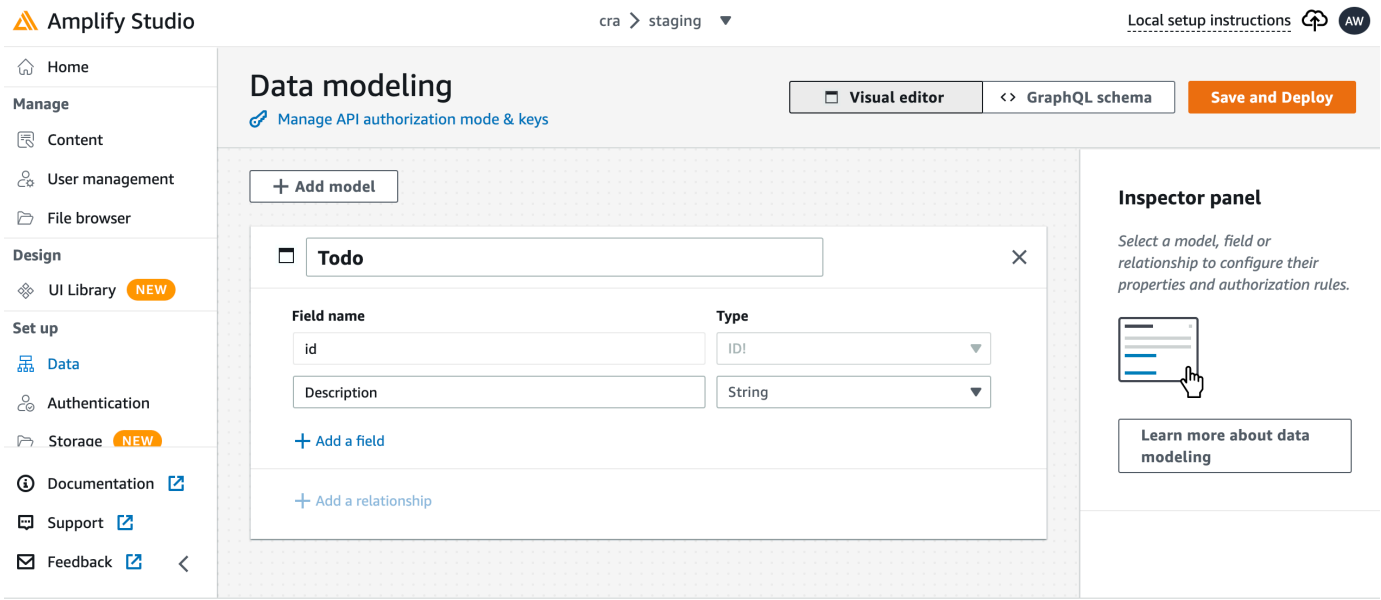
Amplify Studio es una interfaz visual que le permite crear y gestionar el backend, además de acelerar el desarrollo de la interfaz de usuario de frontend. Para obtener más información acerca de Amplify Studio, consulte la documentación de Amplify Studio.

Siga estas instrucciones para crear una base de datos simple usando la interfaz del compilador visual de backend de Amplify Studio.

Crear un modelo de datos

1. En la página de inicio del entorno de ensayo de su aplicación, elija Crear modelo de datos. Se abrirá el diseñador del modelo de datos.
2. En la página Modelado de datos, elija Añadir modelo.
3. Para el título, indique **`Todo`**.
4. Elija Añadir un campo.
5. En Nombre de campo, indique **`Description`**.

   La siguiente captura de pantalla es un ejemplo del aspecto que tendrá su modelo de datos en el diseñador.



6. Elija Guardar e implementar.
7. Regrese a la consola de Amplify Hosting. La implementación del entorno de ensayo estará ya en marcha.

Durante la implementación, Amplify Studio crea todos los AWS recursos necesarios en el backend, incluida una API de AWS AppSync GraphQL para acceder a los datos y una tabla de Amazon DynamoDB para alojar los elementos de Todo. Amplify utiliza AWS CloudFormation para implementar su backend, lo que le permite almacenar su definición de backend como. infrastructure-as-code

Paso 3: conectar el backend al frontend

Ahora que ha implementado un frontend y ha creado un backend en la nube que contiene un modelo de datos, debe conectarlos. Siga estas instrucciones para conectar su definición de backend con su proyecto de aplicación local mediante la CLI de Amplify.

Para conectar un backend en la nube a un frontend local

1. Abra una ventana de terminal y acceda al directorio raíz de su proyecto local.
2. Ejecute el siguiente comando en la ventana del terminal, sustituyendo el texto rojo por el identificador único de aplicación y el nombre del entorno de backend de su proyecto.

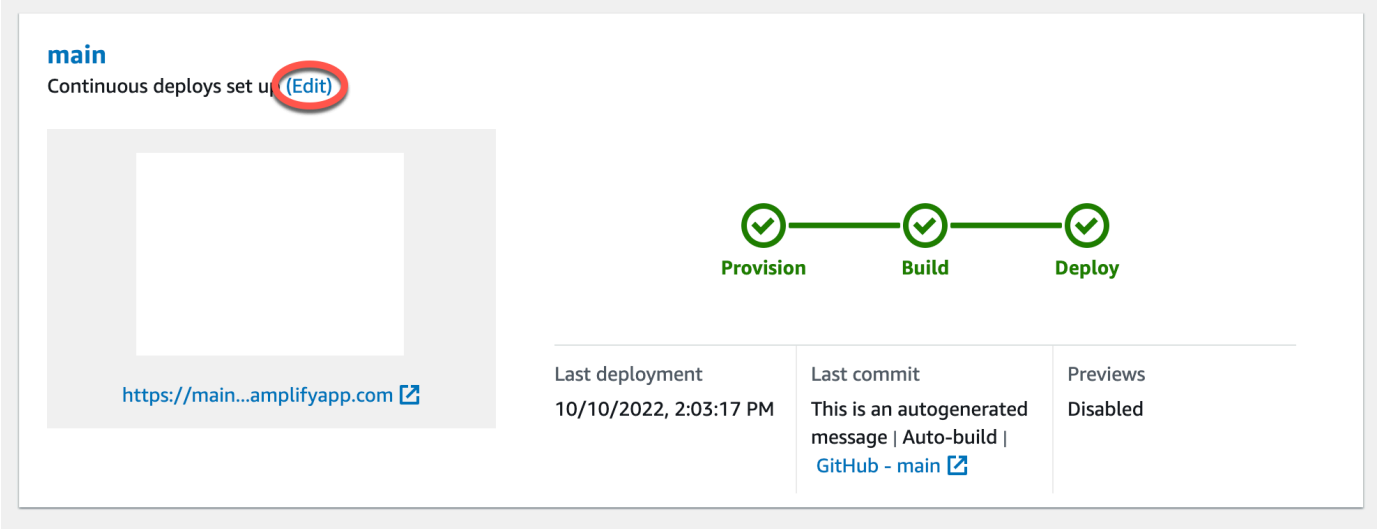
```
amplify pull --appId abcd1234 --envName staging
```

3. Siga las instrucciones de la ventana del terminal para completar la configuración del proyecto.

Ahora puede configurar el proceso de compilación para añadir el backend al flujo de trabajo de implementación continua. Siga estas instrucciones para conectar una ramificación de frontend con un backend en la consola de Amplify Hosting.

Para conectar una ramificación de aplicación de frontend y un backend en la nube

1. En la página de inicio de la aplicación, elija la pestaña Entornos de alojamiento.
2. Localice la ramificación principal y elija Editar.



3. En la ventana Editar backend de destino, en Entorno, seleccione el nombre del backend que desea conectar. En este ejemplo, elegiremos el backend de ensayo que ha creado en el Paso 2.

De forma predeterminada, la pila completa CI/CD is enabled. Uncheck this option to turn off full-stack CI/CD for this backend. Turning off full-stack CI/CD hace que la aplicación se ejecute en modo de solo extracción. En el momento de la compilación, Amplify generará automáticamente el archivo `aws-exports.js` sin modificar el entorno de backend.

4. A continuación, deberá configurar un rol de servicio que conceda a Amplify los permisos necesarios para realizar cambios en el backend de su aplicación. Puede usar un rol de servicio existente o crear uno nuevo. Para obtener instrucciones, consulte Añadir un rol de servicio con permisos para implementar recursos de backend.
5. Tras añadir un rol de servicio, vuelva a la ventana Editar backend de destino y elija Guardar.
6. Para finalizar la conexión del backend de ensayo a la ramificación principal de la aplicación frontend, compile su proyecto.

   Realice una de las siguientes acciones:

   - Desde su repositorio de git, inserte un código para iniciar una compilación en la consola de Amplify.
   - En la consola de Amplify, desplácese a la página de detalles de compilación de la aplicación y elija Volver a implementar esta versión.

Pasos a seguir a continuación

Configurar implementaciones de ramificaciones de características

Siga nuestro flujo de trabajo recomendado para configurar implementaciones de ramificación de características con múltiples entornos de backend.

Cree una interfaz de usuario de frontend en Amplify Studio

Usa Studio para crear la interfaz de usuario de la interfaz de usuario con un conjunto de componentes de la ready-to-use interfaz de usuario y, a continuación, conéctala al servidor de la aplicación. Para obtener más información y ver tutoriales, consulte la guía del usuario de Amplify Studio en la Documentación de Amplify Framework.

Uso del botón Implementar para Amplificar para compartir un proyecto GitHub

> **Important**
>
> La implementación con un solo clic mediante el botón Implementar para Amplificar el alojamiento ya no está disponible. Para implementar desde un repositorio, crea una nueva aplicación en Amplify Hosting. Para obtener instrucciones, consulte Introducción a la implementación de una aplicación en Amplify Hosting.

El botón Implementar para Amplificar el alojamiento te permite compartir GitHub proyectos públicamente o dentro de tu equipo. La siguiente es una imagen del botón:

Agregar el botón Implementación en Amplify Hosting en un repositorio o blog

Agrega el botón a tu archivo GitHub README.md, entrada de blog o cualquier otra página de marcado que muestre HTML. El botón incluye los siguientes dos componentes:

1. Una imagen SVG situada en la dirección URL `https://oneclick.amplifyapp.com/button.svg`
2. La URL de la consola de Amplify con un enlace a tu GitHub repositorio. Puede copiar la dirección URL del repositorio, por ejemplo `https://github.com/username/repository`, o puede proporcionar un enlace directo a una carpeta específica, por ejemplo `https://github.com/username/repository/tree/branchname/folder`. Amplify Hosting implementará la ramificación predeterminada en el repositorio. Las ramificaciones adicionales se pueden conectar una vez conectada la aplicación.

Usa el siguiente ejemplo para añadir el botón a un archivo markdown, como tu GitHub archivo README.md. Sustituya `https://github.com/username/repository` por la dirección URL del repositorio.

```
[![amplifybutton](https://oneclick.amplifyapp.com/button.svg)](https://
console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/username/
repository)
```

Utilice el siguiente ejemplo para añadir el botón a cualquier documento HTML. Sustituya `https://github.com/username/repository` por la dirección URL del repositorio.

```
<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>
```

Configuración del acceso de Amplify a los repositorios GitHub

Amplify ahora usa la función GitHub Aplicaciones para autorizar a Amplify el acceso de solo lectura a los repositorios. GitHub Con la GitHub aplicación Amplify, los permisos están más ajustados, lo que te permite conceder acceso a Amplify solo a los repositorios que especifiques. Para obtener más información sobre GitHub las aplicaciones, consulte Acerca GitHub de las aplicaciones en el sitio web. GitHub

Cuando conectas una nueva aplicación almacenada en un GitHub repositorio, Amplify usa GitHub la aplicación de forma predeterminada para acceder al repositorio. Sin embargo, las aplicaciones de Amplify existentes a las que se conectó anteriormente desde los GitHub repositorios se utilizan para OAuth acceder. El CI/CD seguirá funcionando para estas aplicaciones, pero le recomendamos encarecidamente que las migre para usar la nueva aplicación Amplify. GitHub

Al implementar una nueva aplicación o migrar una aplicación existente mediante la consola Amplify, se le redirige automáticamente a la ubicación de instalación de la aplicación Amplify. GitHub Para acceder manualmente a la página de instalación de la aplicación, abra un navegador web y acceda a la aplicación según su región. Usa el formato `https://github.com/apps/aws-amplify-`*`REGION`* y *`REGION`* sustitúyelo por la región en la que implementarás tu aplicación Amplify. Por ejemplo, para instalar la GitHub aplicación Amplify en la región EE.UU. Oeste (Oregón), vaya a https://github.com/apps/ aws-amplify-us-west -2.

Temas

- Instalación y autorización de la aplicación GitHub Amplify para una nueva implementación
- Migración de una existente OAuth aplicación a la aplicación Amplify GitHub
- Configuración de la GitHub aplicación Amplify para las implementaciones de AWS CloudFormation CLI y SDK
- Configuración de vistas previas web con la aplicación Amplify GitHub

Instalación y autorización de la aplicación GitHub Amplify para una nueva implementación

Cuando implementes una nueva aplicación en Amplify a partir del código existente en un GitHub repositorio, sigue las siguientes instrucciones para instalar y autorizar la aplicación. GitHub

Para instalar y autorizar la aplicación Amplify GitHub

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, elija Nueva aplicación y, a continuación, Alojar aplicación web.
3. En la página Comenzar con Amplify Hosting, selecciona y, a continuación GitHub, selecciona Continuar.
4. Si es la primera vez que se conecta a un GitHub repositorio, se abrirá una nueva página en su navegador en GitHub .com en la que se solicitará permiso para iniciar sesión AWS Amplify en su GitHub cuenta. Seleccione Autorizar.
5. A continuación, debe instalar la GitHub aplicación Amplify en su GitHub cuenta. Se abre una página en GitHub.com solicitando permiso para instalar y autorizar AWS Amplify en tu cuenta. GitHub
6. Selecciona la GitHub cuenta en la que deseas instalar la aplicación Amplify GitHub .
7. Realice una de las siguientes acciones:
   - Para aplicar la instalación a todos los repositorios, elija Todos los repositorios.
   - Para limitar la instalación solo a repositorios específicos, elija Solo los repositorios seleccionados. Asegúrese de incluir el repositorio de la aplicación que está migrando en los repositorios que seleccione.
8. Elija Instalar y autorizar.
9. Se le redirigirá a la página Añadir ramificación de repositorio de su aplicación en la consola de Amplify.
10. En la lista de Repositorios actualizados recientemente, seleccione el nombre del repositorio que desea conectar.
11. En la lista de Ramificaciones, seleccione el nombre de la ramificación del repositorio que desea conectar.
12. Elija Siguiente.
13. En la página Configurar los ajustes de compilación, elija Siguiente.
14. En la página Revisar, elija Guardar e implementar.

Migración de una existente OAuth aplicación a la aplicación Amplify GitHub

Las aplicaciones de Amplify existentes a las que previamente se conectó desde los repositorios se utilizan OAuth para acceder a GitHub los repositorios. Le recomendamos encarecidamente que migre estas aplicaciones para usar la aplicación Amplify GitHub.

Siga las siguientes instrucciones para migrar una aplicación y eliminar el OAuth webhook correspondiente en su GitHub cuenta. Tenga en cuenta que el procedimiento de migración varía en función de si la aplicación GitHub Amplify ya está instalada. Tras migrar la primera aplicación e instalarla y autorizarla, solo tendrá que actualizar los permisos del repositorio para las siguientes migraciones de aplicaciones. GitHub

Para migrar una aplicación de OAuth a la GitHub aplicación

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación que desea migrar.
3. En la página de información de la aplicación, busca el mensaje azul Migrar a nuestra GitHub aplicación y selecciona Iniciar la migración.
4. En la página Instalar y autorizar GitHub la aplicación, selecciona Configurar GitHub la aplicación.
5. Se abre una nueva página en tu navegador, en GitHub .com, en la que se solicita permiso para autorizar AWS Amplify el acceso a tu GitHub cuenta. Seleccione Autorizar.
6. Selecciona la GitHub cuenta en la que deseas instalar la aplicación Amplify GitHub .
7. Realice una de las siguientes acciones:
   - Para aplicar la instalación a todos los repositorios, elija Todos los repositorios.
   - Para limitar la instalación solo a repositorios específicos, elija Solo los repositorios seleccionados. Asegúrese de incluir el repositorio de la aplicación que desea migrar en los repositorios que seleccione.
8. Elija Instalar y autorizar.
9. Se le redirigirá a la página Instalar y autorizar GitHub la aplicación de su aplicación en la consola Amplify. Si GitHub la autorización se ha realizado correctamente, aparecerá un mensaje de confirmación. Elija Siguiente.
10. En la página Completar instalación, elija Completar instalación. Este paso eliminará el webhook existente, creará uno nuevo y finalizará la migración.

Configuración de la GitHub aplicación Amplify para las implementaciones de AWS CloudFormation CLI y SDK

Las aplicaciones de Amplify existentes a las que previamente se conectó desde los repositorios se utilizan OAuth para acceder a GitHub los repositorios. Esto puede incluir aplicaciones que haya implementado mediante la Interfaz de línea de comandos (CLI) de Amplify o la. AWS CloudFormation SDKs Le recomendamos encarecidamente que migre estas aplicaciones para usar la nueva aplicación Amplify GitHub . La migración debe llevarse a cabo en la consola Amplify de la AWS Management Console. Para obtener instrucciones, consulte Migración de una existente OAuth aplicación a la aplicación Amplify GitHub .

Puede utilizar AWS CloudFormation la CLI de Amplify y la SDKs para implementar una nueva aplicación Amplify que utilice la GitHub aplicación para acceder al repositorio. Este proceso requiere que primero instales la GitHub aplicación Amplify en tu GitHub cuenta. A continuación, deberá generar un token de acceso personal en su GitHub cuenta. Por último, deberá implementar la aplicación y especificar el token de acceso personal.

Instala la GitHub aplicación Amplify en tu cuenta

1. Abre un navegador web y navega hasta la ubicación de instalación de la GitHub aplicación Amplify en la AWS región en la que vas a implementar la aplicación.

   Utilice el formato y `https://github.com/apps/aws-amplify-REGION/installations/new` *REGION* sustitúyalo por su propia entrada. Por ejemplo, si va a instalar la aplicación en la región Oeste de EE. UU. (Oregón), especifique `https://github.com/apps/aws-amplify-us-west-2/installations/new`.
2. Selecciona la GitHub cuenta en la que quieres instalar la aplicación Amplify GitHub .
3. Realice una de las siguientes acciones:
   - Para aplicar la instalación a todos los repositorios, elija Todos los repositorios.
   - Para limitar la instalación solo a repositorios específicos, elija Solo los repositorios seleccionados. Asegúrese de incluir el repositorio de la aplicación que está migrando en los repositorios que seleccione.
4. Elija Instalar.

Genera un token de acceso personal en tu cuenta GitHub

1. Inicia sesión en tu GitHub cuenta.
2. En la esquina superior derecha, busque su foto de perfil y elija Configuración en el menú.
3. En el menú de navegación izquierdo, elija Configuración del desarrollador.
4. En la página de GitHub aplicaciones, en el menú de navegación de la izquierda, selecciona Tokens de acceso personal.
5. En la página Tokens de acceso personal, elija Generar nuevo token.
6. En la página Nuevo token de acceso personal, en Nota, introduzca un nombre descriptivo para el token.
7. En la sección Seleccionar ámbitos, seleccione admin:repo_hook.
8. Elija Generar token.
9. Copie y guarde el token de acceso personal. Deberá proporcionarla cuando implemente una aplicación Amplify con la CLI o AWS CloudFormation la. SDKs

Una vez que la GitHub aplicación Amplify esté instalada en su GitHub cuenta y haya generado un token de acceso personal, puede implementar una nueva aplicación con la CLI de Amplify o la. AWS CloudFormation SDKs En el campo `accessToken`, introduzca el token de acceso personal que creó en el procedimiento anterior. Para obtener más información, consulte CreateAppla referencia de la API Amplify y AWS::Amplify::Appla Guía del AWS CloudFormation usuario.

El siguiente comando CLI implementa una nueva aplicación Amplify que usa la aplicación para acceder GitHub al repositorio. Reemplace *myapp-using-githubapp* *https://github.com/Myaccount/react-app* y *MY_TOKEN* con su propia información.

```
aws amplify create-app --name myapp-using-githubapp --repository https://github.com/
Myaccount/react-app --access-token MY_TOKEN
```

Configuración de vistas previas web con la aplicación Amplify GitHub

Una vista previa web despliega todas las solicitudes de extracción (PR) realizadas en tu GitHub repositorio en una URL de vista previa única. Las vistas previas ahora utilizan la GitHub aplicación

Amplify para acceder a su GitHub repositorio. Para obtener instrucciones sobre cómo instalar y autorizar la GitHub aplicación para las vistas previas web, consulte. Habilita las vistas previas web para las solicitudes de extracción

Vistas previas web para solicitudes de extracción

Las vistas previas web ofrecen a los equipos de desarrollo y control de calidad (QA) una forma de previsualizar los cambios de las solicitudes de cambios (PRs) antes de fusionar el código en una rama de producción o integración. Las solicitudes de extracción le permiten informar a otros sobre los cambios introducidos en una ramificación de un repositorio. Tras abrir una solicitud de extracción, puede analizar y revisar los posibles cambios con sus colaboradores y añadir confirmaciones de seguimiento antes de fusionar los cambios en la ramificación base.

La vista previa web presenta todas las solicitudes de extracción realizadas en el repositorio en una URL de vista previa única. Esta URL es totalmente diferente a la de su sitio principal. En el caso de las aplicaciones con entornos de backend aprovisionados mediante la CLI de Amplify o Amplify Studio, cada solicitud de extracción (solo repositorios Git privados) crea un backend temporal que se elimina al cerrar la PR.

Cuando las vistas previas web están activadas para su aplicación, cada PR cuenta para la cuota de Amplify, que consiste en 50 ramificaciones por aplicación. Para evitar superar esta cuota, asegúrate de cerrar la tuya PRs. Para obtener más información sobre las cuotas, consulte Service Quotas de Amplify Hosting.

> **Note**
>
> Actualmente, la variable de `AWS_PULL_REQUEST_ID` entorno no está disponible cuando se utiliza AWS CodeCommit como proveedor de repositorios.

Seguridad de vista previa web

Por motivos de seguridad, puedes habilitar las vistas previas web en todas las aplicaciones con repositorios privados, pero no en todas las aplicaciones con repositorios públicos. Si su repositorio de Git es público, puede configurar vistas previas solo para las aplicaciones que no requieran un rol de servicio de IAM. Por ejemplo, las aplicaciones con backend y aquellas que se implementan en la plataforma de alojamiento de `WEB_COMPUTE` requieren un rol de servicio de IAM. Por lo tanto, si su repositorio es público, no podrá habilitar las vistas previas web para este tipo de aplicaciones. Amplify aplica esta restricción para evitar que posibles terceros envíen un código arbitrario que se ejecutaría con los permisos de rol de IAM de su aplicación.

Cuando se habilitan las vistas previas web para una aplicación en un repositorio público, con una función de SSR Compute, es necesario gestionar cuidadosamente las sucursales que pueden acceder a esa función. Te recomendamos que no utilices un rol a nivel de aplicación. En su lugar, debes asignar un rol de cómputo a nivel de sucursal. Esto te permite conceder permisos solo a las sucursales que requieren acceso a recursos específicos. Para obtener más información, consulte Añadir un rol de SSR Compute para permitir el acceso a los recursos AWS.

Habilita las vistas previas web para las solicitudes de extracción

En el caso de las aplicaciones almacenadas en un GitHub repositorio, las vistas previas web utilizan la aplicación GitHub Amplify para acceder al repositorio. Si está habilitando las vistas previas web en una aplicación Amplify existente que implementó anteriormente desde un GitHub repositorio utilizando OAuth para acceder, primero debe migrar la aplicación para usar la aplicación Amplify. GitHub Para obtener información sobre cómo realizar la migración, consulte Migración de una existente OAuth aplicación a la aplicación Amplify GitHub .

Para habilitar las vistas previas web de solicitudes de extracción

1. Seleccione Alojamiento y, a continuación, Vistas previas.

   > **Note**
   >
   > Vistas previas solo es visible en el menú Configuración de aplicación cuando una aplicación está configurada para implementación continua y conectada a un repositorio de git. Para obtener instrucciones sobre este tipo de implementación, consulte Primeros pasos con el código existente.

2. Solo para GitHub los repositorios, haz lo siguiente para instalar y autorizar la aplicación GitHub Amplify en tu cuenta:

   a. En la ventana Instalar GitHub aplicación para habilitar las vistas previas, seleccione Instalar GitHub aplicación.

   b. Seleccione la GitHub cuenta en la que desee configurar la aplicación Amplify GitHub.

   c. Se abrirá una página en GitHub.com para configurar los permisos de repositorio de su cuenta.

   d. Realice una de las siguientes acciones:

      - Para aplicar la instalación a todos los repositorios, elija Todos los repositorios.

- Para limitar la instalación solo a repositorios específicos, elija Solo los repositorios seleccionados. Asegúrese de incluir en esta selección el repositorio de la aplicación para la que desea habilitar las vistas previas web.

e. Elija Guardar

3. Tras habilitar las vistas previas para su repositorio, vuelva a la consola de Amplify para habilitar las vistas previas de ramificaciones específicas. En la página Vistas previas, elija una ramificación de la lista y elija Editar configuración.
4. En la ventana Administrar la configuración de la vista previa, active Vistas previas de solicitudes de extracción. A continuación, seleccione Confirm (Confirmar).
5. Para aplicaciones de pila completa, siga uno de estos pasos:
   - Elija Crear nuevo entorno de backend para cada solicitud de extracción. Esta opción le permitirá probar los cambios sin que ello afecte a la producción.
   - Elija Dirigir todas las solicitudes de extracción de esta ramificación a un entorno existente.
6. Elija Confirmar.

La próxima vez que envíe una solicitud de extracción a esa ramificación, Amplify compilará e implementará su PR en una URL de vista previa. Una vez cerrada la solicitud de extracción, se eliminará la URL de vista previa, así como cualquier entorno de backend temporal vinculado a la solicitud de extracción. Solo en el caso de GitHub los repositorios, puedes acceder a una vista previa de tu URL directamente desde la solicitud de extracción de información de tu GitHub cuenta.

Acceso a vista previa web con subdominios

Es posible acceder a vistas previas web de las solicitudes de extracción con los subdominios de una aplicación de Amplify que esté conectada a un dominio personalizado administrado por Amazon Route 53. Una vez cerrada la solicitud de extracción, las ramificaciones y subdominios asociados a la misma se eliminarán automáticamente. Tras configurar la implementación de ramificaciones con características basadas en patrón para su aplicación, este será el comportamiento predeterminado de las vistas previas web. Para obtener más información sobre cómo configurar los subdominios automáticos, consulte Configuración de subdominios automáticos para un dominio personalizado de Amazon Route 53.

Configuración de las pruebas de end-to-end Cypress para su aplicación Amplify

Puedes ejecutar pruebas end-to-end (E2E) en la fase de prueba de tu aplicación Amplify para detectar las regresiones antes de pasar el código a producción. La fase de pruebas se puede configurar en la especificación de compilación YAML. Actualmente, solo es posible ejecutar el marco de pruebas de Cypress durante una compilación.

Cypress es un marco de pruebas JavaScript basado en el que puede ejecutar pruebas E2E en un navegador. Para ver un tutorial que demuestra cómo configurar las pruebas E2E, consulte la entrada del blog Ejecución de pruebas de end-to-end Cypress para una implementación completa de CI/CD con Amplify.

Adición de pruebas de Cypress a una aplicación de Amplify existente

Puede añadir pruebas de Cypress a una aplicación existente actualizando la configuración de compilación de la aplicación en la consola de Amplify. El archivo YAML de especificación de compilación contiene un conjunto de comandos de compilación y ajustes relacionados que Amplify utiliza para ejecutar la compilación. Lleve a cabo el paso `test` para ejecutar cualquier comando de prueba en el momento de la compilación. En el caso de las pruebas E2E, Amplify Hosting ofrece una integración más profunda con Cypress que le permite generar un informe de interfaz de usuario para sus pruebas.

La siguiente lista describe la configuración de pruebas y su utilización.

preTest

Instala las dependencias necesarias para ejecutar las pruebas de Cypress. Amplify Hosting usa mochawesome para generar un informe con los resultados de las pruebas, y wait-on para configurar el servidor localhost durante la compilación.

prueba

Ejecuta los comandos de Cypress para realizar pruebas con mochawesome.

postTest

Se genera un informe de mochawesome a partir del JSON de salida. Tenga en cuenta que, si usa Yarn, debe ejecutar este comando en modo silencioso para generar el informe de mochawesome. Con Yarn puede usar el comando siguiente:

```
yarn run --silent mochawesome-merge cypress/report/mochawesome-report/
mochawesome*.json > cypress/report/mochawesome.json
```

artifacts>baseDirectory

Directorio desde el que se ejecutan las pruebas.

artefactos> configFilePath

Datos del informe de prueba generado.

artifacts>files

Los artefactos generados (capturas de pantalla y vídeos) están disponibles para descargar.

El siguiente extracto de ejemplo de un archivo `amplify.yml` de especificaciones de compilación muestra cómo agregar pruebas de Cypress a su aplicación.

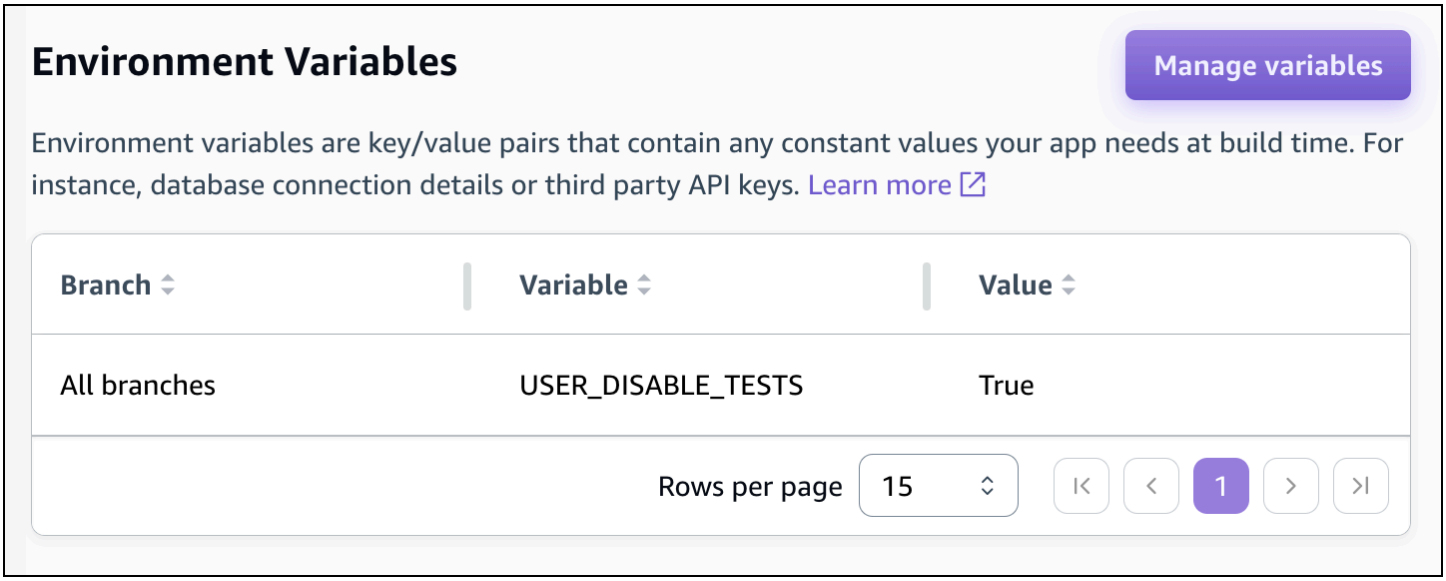
```
test:
  phases:
    preTest:
      commands:
        - npm ci
        - npm install -g pm2
        - npm install -g wait-on
        - npm install mocha mochawesome mochawesome-merge mochawesome-report-generator
        - pm2 start npm -- start
        - wait-on http://localhost:3000
    test:
  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

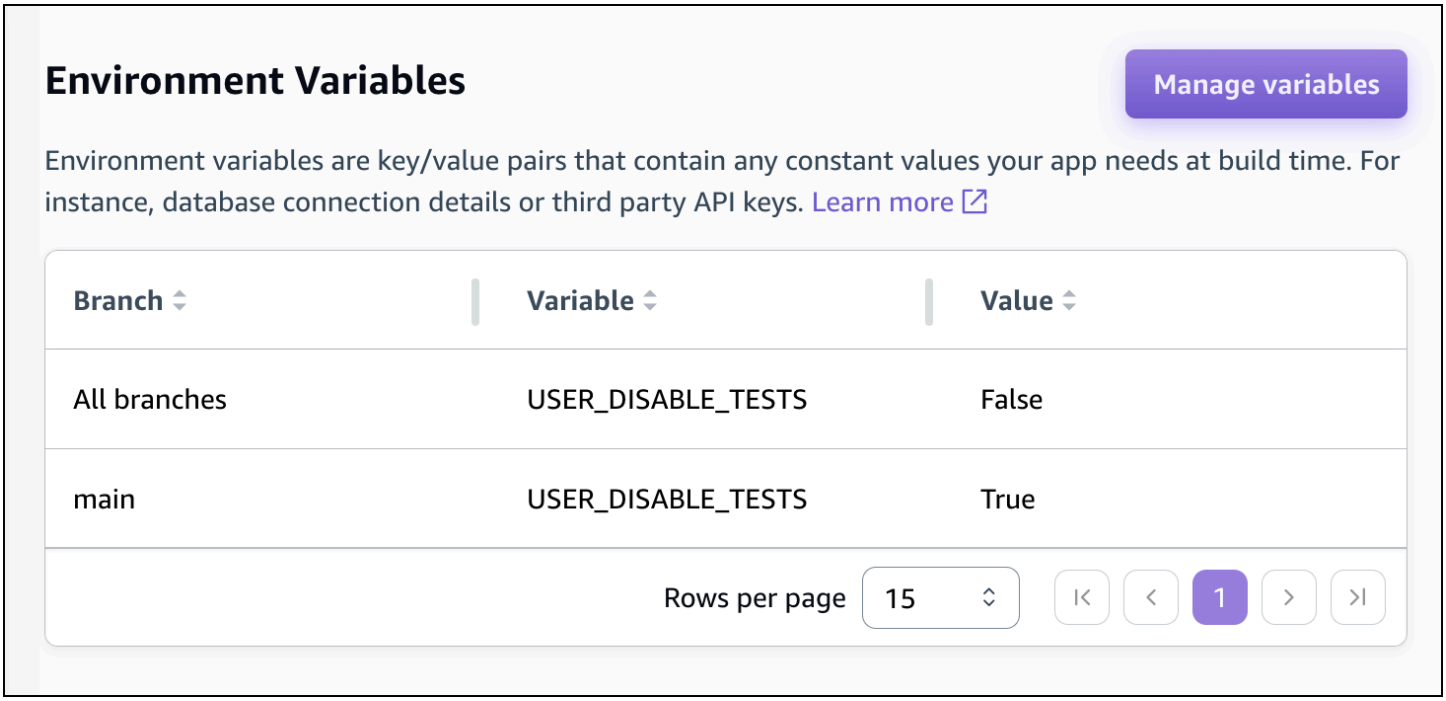
Desactivación de las pruebas de una aplicación o ramificación de Amplify

Una vez añadida la configuración de pruebas a los ajustes de compilación de `amplify.yml`, el paso `test` se ejecutará en cada compilación y ramificación. Si quieres deshabilitar globalmente la ejecución de las pruebas o solo ejecutar pruebas para ramas específicas, puedes usar la USER_DISABLE_TESTS variable de entorno sin modificar la configuración de compilación.

Para deshabilitar globalmente las pruebas en todas las ramas, agrega la USER_DISABLE_TESTS variable de entorno con un valor de `true` para todas las ramas. La siguiente captura de pantalla muestra la sección de Variables de entorno de la consola Amplify con las pruebas deshabilitadas en todas las ramificaciones.



Para deshabilitar las pruebas de una rama específica, añada la USER_DISABLE_TESTS variable de entorno con un valor de `false` para todas las ramas y, a continuación, añada una anulación para cada rama que desee deshabilitar con un valor de`true`. En la siguiente captura de pantalla, las pruebas se desactivan en la ramificación principal y se activan en todas las demás ramificaciones.



Al deshabilitar las pruebas con esta variable, se omitirá por completo el paso de pruebas durante la compilación. Para volver a habilitar las pruebas, defina este valor como `false` o elimine la variable de entorno.

Configuración de redirecciones y reescrituras de una aplicación de Amplify

Los redireccionamientos permiten a un servidor web redirigir la navegación desde una URL a otra. Entre los motivos habituales para el uso de redireccionamientos se incluye: personalizar el aspecto de una dirección URL, evitar enlaces rotos, mover la ubicación de alojamiento de una aplicación o sitio sin cambiar su dirección y cambiar una dirección URL solicitada a la forma que necesita una aplicación web.

Descripción de los redireccionamientos que admite Amplify

Amplify admite los siguientes tipos de redireccionamiento en la consola.

Redireccionamiento permanente (301)

Los redireccionamientos 301 se han diseñado para cambios duraderos en el destino de una dirección web. El historial de clasificación del motor de búsqueda de la dirección original se aplica a la nueva dirección de destino. El redireccionamiento se produce en el lado del cliente, por tanto, una barra de navegación de explorador muestra la dirección de destino tras el redireccionamiento.

Entre los motivos habituales para utilizar redireccionamientos 301 se incluyen:

- Para evitar un enlace que no funciona cuando cambia la dirección de una página.
- Para evitar un enlace que no funciona cuando un usuario comete un error tipográfico predecible en una dirección.

Redireccionamiento temporal (302)

Los redireccionamientos 302 se han diseñado para cambios temporales en el destino de una dirección web. El historial de clasificación del motor de búsqueda de la dirección original no se aplica a la nueva dirección de destino. El redireccionamiento se produce en el lado del cliente, por tanto, una barra de navegación de explorador muestra la dirección de destino tras el redireccionamiento.

Entre los motivos habituales para utilizar redireccionamientos 302 se incluyen:

- Proporcionar un destino alternativo mientras se llevan a cabo reparaciones en una dirección original.

- Proporcionar páginas de prueba para una comparación A/B de interfaz de usuario.

> **Note**
>
> Si su aplicación devuelve una respuesta 302 inesperada, es probable que el error se deba a los cambios que has realizado en la configuración de redireccionamiento y de los encabezados personalizados de la aplicación. Para solucionar este problema, compruebe que los encabezados personalizados sean válidos y, a continuación, vuelva a activar la regla de reescritura 404 predeterminada de su aplicación.

Reescritura (200)

Los redireccionamientos 200 (reescrituras) se han diseñado para mostrar contenido desde la dirección de destino como si se sirviera desde la dirección original. El historial de clasificación del motor de búsqueda se sigue aplicando a la dirección original. El redireccionamiento se produce en el lado del servidor, por tanto, una barra de navegación de explorador muestra la dirección original tras el redireccionamiento. Entre los motivos habituales para utilizar redireccionamientos 200 se incluyen:

- Para redirigir un sitio completo a una nueva ubicación de alojamiento sin cambiar la dirección del sitio.
- Para redireccionar todo el tráfico a una aplicación de web de página única (SPA) a su página index.html para gestión por parte de una función de router del lado del cliente.

No encontrado (404)

Los redireccionamientos 404 se producen cuando una solicitud apunta a una dirección que no existe. Se muestra la página de destino de un error 404 en lugar de la solicitada. Entre los motivos habituales por los que se produce un redireccionamiento 404 se incluyen:

- Para evitar un mensaje de enlace que no funciona cuando un usuario introduce una dirección URL incorrecta.
- Para apuntar solicitudes a páginas no existentes de una aplicación web a su página index.html para gestión por parte de una función de router del lado del cliente.

Descripción del orden de los redireccionamientos

Los redireccionamientos se aplican desde la parte superior de la lista hacia abajo. Asegúrese de que el orden tenga el efecto previsto. Por ejemplo, el siguiente orden de redireccionamientos hace que todas las solicitudes de una ruta determinada en /docs/ se redirijan a la misma ruta en /documents/, excepto /docs/specific-filename.html que redirige a /documents/different-filename.html:

```
/docs/specific-filename.html /documents/different-filename.html 301
/docs/<*> /documents/<*>
```

El siguiente orden de redireccionamientos omite el redireccionamiento de specific-filename.html a different-filename.html:

```
/docs/<*> /documents/<*>
/docs/specific-filename.html /documents/different-filename.html 301
```

Descripción de cómo Amplify reenvía los parámetros de consulta

Puede utilizar parámetros de consulta para tener un mayor control sobre las coincidencias de dirección URL. Amplify reenvía todos los parámetros de consulta a la ruta de destino para los redireccionamientos 301 y 302, con las siguientes excepciones:

- Si la dirección original incluye una cadena de consulta establecida en un valor específico, Amplify no reenvía los parámetros de la consulta. En este caso, el redireccionamiento solo se aplica a las solicitudes a la dirección URL de destino con el valor de consulta especificado.
- Si la dirección de destino de la regla coincidente tiene parámetros de consulta, los parámetros de consulta no se reenvían. Por ejemplo, si la dirección de destino del redireccionamiento es `https://example-target.com?q=someParam`, los parámetros de consulta no se transfieren.

Creación y edición de redireccionamientos en la consola de Amplify

Puede crear y editar redireccionamientos de una aplicación en la consola de Amplify. Antes de comenzar, necesita la siguiente información sobre las partes de un redireccionamiento.

Una dirección original

La dirección que solicitó el usuario.

Una dirección de destino

La dirección que realmente ofrece el contenido que el usuario ve.

Un tipo de redireccionamiento

Entre los tipos se incluye un redireccionamiento permanente (301), un redireccionamiento temporal (302), una reescritura (200) o no encontrado (404).

Un código de país de dos letras (opcional)

Un valor que puede incluir para segmentar la experiencia de usuario de su aplicación por región geográfica.

Para crear un redireccionamiento en la consola de Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea crear un redireccionamiento.
3. En el panel de navegación, elija Alojamiento y, a continuación, elija Reescrituras y redireccionamientos.
4. En la página Reescrituras y redireccionamientos, seleccione Administrar redireccionamientos.
5. El procedimiento para añadir un redireccionamiento varía en función de si desea añadir las reglas de forma individual o realizar una edición masiva:
   - Para crear un redireccionamiento individual, elija Agregar regla.
     a. En Dirección de origen, introduzca la dirección original solicitada por el usuario.
     b. En Dirección de destino, introduzca la dirección de destino que muestra el contenido al usuario.
     c. En Tipo, elija el tipo de redireccionamiento de la lista.
     d. (Opcional) En Código de país, introduzca una condición de código de país de dos letras.
   - Para editar redireccionamientos masivos, elija Abrir editor de texto.
     - Añada o actualice manualmente los redireccionamientos en el editor JSON de Reescrituras y redireccionamientos.
6. Seleccione Guardar.

Referencia de ejemplo de redireccionamientos y reescrituras

En esta sección se incluye código de ejemplo para una variedad de situaciones de redireccionamiento comunes. Puede usar estos ejemplos a fin de entender la sintaxis necesaria para crear sus propias redirecciones y reescrituras.

> **Note**
>
> La coincidencia de dominios de direcciones originales no distingue entre mayúsculas y minúsculas.

Temas

Redireccionamientos y reescrituras sencillos

Puede utilizar el siguiente código de ejemplo para redirigir permanentemente una página específica a una nueva dirección.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/original.html`	`/destinat ion.html`	`permanent redirect (301)`	

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

Puede utilizar el siguiente código de ejemplo para redirigir cualquier ruta en una carpeta a la misma ruta de una carpeta diferente.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

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

Puede utilizar el siguiente código de ejemplo para redirigir todo el tráfico a index.html como una reescritura. En esta situación, la reescritura hace que aparezca al usuario que ha llegado a la dirección original.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/<*>`	`/index.html`	`rewrite (200)`	

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

Puede utilizar el siguiente código de ejemplo para usar una reescritura para cambiar el subdominio que aparece al usuario.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`https://m ydomain.com`	`https://w ww.mydoma in.com`	`rewrite (200)`	

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

Puede utilizar el siguiente código de ejemplo para redirigir a un dominio diferente con un prefijo de ruta.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`https://m ydomain.com`	`https://w ww.mydoma in.com/do cuments`	`temporary redirect (302)`	

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

Puede utilizar el siguiente código de ejemplo para redirigir rutas de una carpeta que no se puede encontrar a una página 404 personalizada.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/<*>`	`/404.html`	`not found (404)`	

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

Redireccionamientos para aplicaciones web de página única (SPA)

La mayoría de los marcos SPA admiten HTML5 History.pushState () para cambiar la ubicación del navegador sin iniciar una solicitud de servidor. Esto funciona para los usuarios que comienzan su recorrido desde la raíz (o /index.html), pero devuelve un error a los usuarios que van directamente a cualquier otra página.

El ejemplo siguiente utiliza expresiones regulares para configurar una reescritura 200 de todos los archivos en index.html, excepto para las extensiones de archivo concretas especificadas en la expresión regular.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`</^[^.]+$|\.(?! (css|gif|ico`	`/index.html`	`200`	

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`|jpg|js|png| txt|svg|woff| woff2|ttf|map| json|webp)$)([^ .]+$)/>`			

JSON [{"source": "</^[^.]+$|\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|webp)$)([^.]+$)/>", "status": "200", "target": "/index.html", "condition": null}]

Reescritura de proxy inverso

En el siguiente ejemplo se utiliza una reescritura a contenido de proxy desde otra ubicación para que al usuario le parezca que el dominio no ha cambiado.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/images/<*>`	`https://i mages.oth erdomain.com/ <*>`	`rewrite (200)`	

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

Tiras diagonales finales y limpio URLs

Para crear estructuras de direcciones URL limpias como about en lugar de about.html, los generadores de sitios estáticos, como Hugo, generan directorios de páginas con un index.html (/about/index.html). Amplify crea automáticamente la limpieza URLs añadiendo una barra al final cuando es necesario. La tabla siguiente destaca diferentes situaciones:

Entradas de usuario en el navegador	Dirección URL en la barra de dirección	Documento servido
`/about`	`/about`	`/about.html`
`/about (when about.html returns 404)`	`/about/`	`/about/index.html`
`/about/`	`/about/`	`/about/index.html`

Marcadores de posición

Puede utilizar el siguiente código de ejemplo para redirigir rutas en una estructura de carpetas a una estructura coincidente en otra carpeta.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/docs/<year>/<month>/<date>/<itemid>`	`/documents/<year>/<month>/<date>/<itemid>`	`permanent redirect (301)`	

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

Cadenas de consulta y parámetros de ruta

Puede utilizar el siguiente código de ejemplo para redirigir una ruta a una carpeta con un nombre que coincida con el valor de un elemento de cadena de consulta en la dirección original:

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/docs?id=<my-blog-id-value`	`/documents/<my-blog-post-id-value>`	`permanent redirect (301)`	

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

> **Note**
>
> Amplify reenvía todos los parámetros de la cadena de consulta a la ruta de destino para los redireccionamientos 301 y 302. Sin embargo, si la dirección original incluye una cadena de consulta establecida en un valor específico, como se muestra en este ejemplo, Amplify no reenvía los parámetros de la consulta. En este caso, el redireccionamiento solo se aplica a las solicitudes a la dirección de destino con el valor de consulta especificado `id`.

Puede utilizar el siguiente código de ejemplo para redirigir todas las rutas que no se pueden encontrar en un determinado nivel de una estructura de carpetas a index.html de una carpeta especificada.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/documents/<folder>/<child-folder>/<grand-child-folder>`	`/documents/index.html`	`not found (404)`	

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

Redireccionamientos basados en la región

Puede utilizar el siguiente código de ejemplo para redirigir solicitudes según la región.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/documents`	`/documents/us/`	`temporary redirect (302)`	`<US>`

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

Uso de expresiones comodín en las redirecciones y reescrituras

Puede utilizar la expresión comodín (`<*>`) en la dirección original para redirigir o reescribir. Debe colocar la expresión al final de la dirección original, y debe ser única. Amplify ignora las direcciones originales que incluyen más de una expresión comodín o las utiliza en una ubicación diferente.

A continuación, se muestra un ejemplo de un redireccionamiento válido con una expresión comodín.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

Los dos ejemplos siguientes muestran redireccionamientos no válidos con expresiones comodín.

Dirección original	Dirección de destino	Tipo de redirecci onamiento	Código de país
`/docs/<*>/content`	`/documents/<*>/content`	`permanent redirect (301)`	
`/docs/<*>/content/<*>`	`/documents/<*>/content/<*>`	`permanent redirect (301)`	

Restringir el acceso a las ramificaciones de una aplicación de Amplify

Si trabaja en características que aún no se han lanzado, puede proteger con contraseña a las ramificaciones de características para restringir su acceso a determinados usuarios. Cuando se establece el control de acceso en una ramificación, los usuarios deben introducir un nombre de usuario y una contraseña para acceder a la URL de dicha ramificación.

Puede establecer una contraseña que se aplique a una ramificación individual o de manera global a todas las ramificaciones conectadas. Cuando el control de acceso está habilitado en una ramificación individual o de manera global, la contraseña de nivel individual tiene prioridad sobre una contraseña de nivel global (aplicación).

Amplify limita las solicitudes con errores que intentan acceder a recursos protegidos por contraseña. Este comportamiento protege a las aplicaciones contra los ataques de diccionario u otros intentos de leer datos que están detrás de los controles de acceso.

Use el siguiente procedimiento para establecer una contraseña que restrinja el acceso a las ramificaciones de una aplicación de Amplify.

Para establecer contraseñas en las ramificaciones de características

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación cuyas ramificaciones de características desea proteger con contraseña.
3. En el panel de navegación, elija Alojamiento y luego elija Control de acceso.
4. En la sección Configuración de control de acceso, elija Gestionar acceso.
5. En la página Administración de permisos de acceso, haga lo siguiente.
   - Para configurar un nombre de usuario y una contraseña que se aplique en todas las ramificaciones conectadas
     - Active Administrar el acceso a todas las ramificaciones. Por ejemplo, si tiene las ramificaciones principales, de desarrollo y de características conectadas, puede aplicar el mismo nombre de usuario y contraseña para todas las ramificaciones.
   - Para configurar un nombre de usuario y una contraseña que se aplique en todas a una ramificación individual.
     a. Desactive Administrar el acceso a todas las ramificaciones.

b. Ubique la ramificación que quiera administrar. En Configuración de acceso, seleccione Restringido (se requiere contraseña).

c. En Nombre de usuario, introduzca un nombre de usuario.

d. En Contraseña, introduzca una contraseña.

- Seleccione Guardar.

6. Si administra el control de acceso de una aplicación representada en el lado del servidor (SSR), vuelva a implementar la aplicación compilándola de nuevo desde su repositorio Git. Este paso es necesario para que Amplify pueda aplicar la configuración de control de acceso.

Uso de variables de entorno en una aplicación de Amplify

Las variables de entorno son pares de valor clave que se pueden añadir a la configuración de la aplicación para que estén disponibles en Amplify Hosting. Como práctica recomendada, puede utilizar variables de entorno para exponer los datos de configuración de la aplicación. Todas las variables de entorno que añada se cifran para evitar el acceso no autorizado.

Amplify aplica las siguientes restricciones a las variables de entorno que cree.

- Amplify no permite crear nombres de variables de entorno con un prefijo `AWS`. Este prefijo se reserva para uso interno de Amplify.
- El valor de una variable de entorno no puede superar los 5500 caracteres.

> **Important**
>
> No utilice variables de entorno para almacenar claves secretas. Para una aplicación de Gen 2, use la característica Administración de secretos de la consola de Amplify. Para obtener más información, consulte Secrets and environment vars en Amplify Docs. Para una aplicación de primera generación, guarda los secretos en un entorno secreto creado con el almacén de AWS Systems Manager parámetros. Para obtener más información, consulte Administración de los secretos de entorno.

Referencia de variables de entorno de Amplify

Las siguientes variables de entorno son accesibles de forma predeterminada en la consola de Amplify.

Nombre de variable	Descripción	Ejemplo de valor
_BUILD_TIMEOUT	El tiempo de espera de compilación en minutos. El valor mínimo es 5. El valor máximo es 120.	`30`

Nombre de variable	Descripción	Ejemplo de valor
_LIVE_UPDATES	La herramienta se actualizará a la última versión.	`[{"name":"Amplify CLI","pkg":"@aws-amplify/cli","type":"npm","version":"latest"}]`
USER_DISABLE_TESTS	El paso de prueba se omite durante la compilación. Puede deshabilitar las pruebas en todas las ramificaciones o en ramificaciones específicas de una aplicación. Esta variable de entorno se utiliza para las aplicaciones que realizan pruebas durante la fase de compilación. Para obtener más información sobre cómo configurar esta variable, consulte Desactivación de las pruebas de una aplicación o ramificación de Amplify.	`true`
AWS_APP_ID	ID de aplicación de la compilación actual	`abcd1234`
AWS_BRANCH	Nombre de ramificación de la compilación actual	`main`, `develop`, `beta`, `v2.0`
AWS_BRANCH_ARN	El nombre de recurso de Amazon (ARN) de la ramificación de la compilación actual	`aws:arn:amplify:us-west-2:123456789012:appname/branch/...`

Nombre de variable	Descripción	Ejemplo de valor
AWS_CLONE_URL	Dirección URL clonada utilizada para recuperar el contenido del repositorio de Git	`git@github.com:<user-name>/<repo-name>.git`
AWS_COMMIT_ID	ID de confirmación de la compilación actual “HEAD” para las recompilaciones	`abcd1234`
AWS_JOB_ID	ID de trabajo de la compilación actual. Este incluye relleno de ‘0’, por lo que siempre tiene la misma longitud.	`0000000001`
AWS_PULL_REQUEST_ID	ID de solicitud de extracción de la compilación de vista previa web de la solicitud de extracción. Esta variable de entorno no está disponible cuando se utiliza AWS CodeCommit como proveedor de repositorios.	1
AWS_PULL_REQUEST_SOURCE_BRANCH	El nombre de la ramificación de característica de una vista previa de una solicitud de extracción que se envía a una ramificación de aplicación en la consola de Amplify.	`featureA`

Nombre de variable	Descripción	Ejemplo de valor
AWS_PULL_REQUEST_DESTINATION_BRANCH	El nombre de la ramificación de aplicación de la consola de Amplify a la que se envía una solicitud de extracción de una rama de característica.	`main`
AMPLIFY_AMAZON_CLIENT_ID	ID de cliente de Amazon	`123456`
AMPLIFY_AMAZON_CLIENT_SECRET	Secreto del cliente de Amazon	`example123456`
AMPLIFY_FACEBOOK_CLIENT_ID	ID de cliente de Facebook	`123456`
AMPLIFY_FACEBOOK_CLIENT_SECRET	Secreto del cliente de Facebook	`example123456`
AMPLIFY_GOOGLE_CLIENT_ID	ID de cliente de Google	`123456`
AMPLIFY_GOOGLE_CLIENT_SECRET	Secreto del cliente de Google	`example123456`
AMPLIFY_DIFF_DEPLOY	Habilite o deshabilite la implementación de frontend basada en diferencias. Para obtener más información, consulte Configuración de la compilación e implementación de frontend basada en diferencias.	`true`

Nombre de variable	Descripción	Ejemplo de valor
AMPLIFY_DIFF_DEPLOY_ROOT	La ruta que se utilizará para realizar comparaciones de implementaciones de frontend basadas en diferencias en relación con la raíz del repositorio.	`dist`
AMPLIFY_DIFF_BACKEND	Habilite o deshabilite las compilaciones de backend basadas en diferencias. Esto se aplica únicamente a las aplicaciones de Gen 1. Para obtener más información, consulte Configuración de compilaciones de backend basadas en diferencias para una aplicación de Gen 1	`true`
AMPLIFY_BACKEND_PULL_ONLY	Amplify gestiona esta variable de entorno. Esto se aplica únicamente a las aplicaciones de Gen 1. Para obtener más información, consulte Edite un frontend existente para que apunte a un backend distinto	`true`
AMPLIFY_BACKEND_APP_ID	Amplify gestiona esta variable de entorno. Esto se aplica únicamente a las aplicaciones de Gen 1. Para obtener más información, consulte Edite un frontend existente para que apunte a un backend distinto	`abcd1234`

Nombre de variable	Descripción	Ejemplo de valor
AMPLIFY_SKIP_BACKEND_BUILD	Si no tiene una sección de backend en su especificación de compilación y desea deshabilitar las compilaciones de backend, establezca esta variable de entorno en `true`. Esto se aplica únicamente a las aplicaciones de Gen 1.	`true`
AMPLIFY_ENABLE_DEBUG_OUTPUT	Establezca esta variable en `true` para imprimir un seguimiento full stack en los registros. Esto resulta útil para depurar los errores de compilación del backend.	`true`
AMPLIFY_MONOREPO_APP_ROOT	La ruta que se utilizará para especificar la raíz de la aplicación de una aplicación monorepo en relación con la raíz de su repositorio.	`apps/react-app`
AMPLIFY_USERPOOL_ID	ID de grupo de usuarios de Amazon Cognito importado para autenticación	`us-west-2_example`
AMPLIFY_WEBCLIENT_ID	ID de cliente de aplicación que van a utilizar las aplicaciones web El cliente de aplicación debe configurarse con acceso al grupo de usuarios de Amazon Cognito especificado por la variable de entorno AMPLIFY_USERPOOL_ID.	`123456`

Nombre de variable	Descripción	Ejemplo de valor
AMPLIFY_NATIVECLIENT_ID	ID del cliente de aplicación que van a utilizar las aplicaciones nativas El cliente de aplicación debe configurarse con acceso al grupo de usuarios de Amazon Cognito especificado por la variable de entorno AMPLIFY_USERPOOL_ID.	`123456`
AMPLIFY_IDENTITYPOOL_ID	ID de grupo de identidades de Amazon Cognito	`example-identitypool-id`
AMPLIFY_PERMISSIONS_BOUNDARY_ARN	El ARN de la política de IAM que se utilizará como límite de permisos que se aplica a todos los roles de IAM creadas por Amplify.	`arn:aws:iam::123456789012:policy/example-policy`
AMPLIFY_DESTRUCTIVE_UPDATES	Establezca esta variable de entorno como verdadera para permitir que una API de GraphQL se actualice con operaciones de esquema que pueden potencialmente provocar la pérdida de datos.	`true`

> **Note**
>
> Las variables de `AMPLIFY_AMAZON_CLIENT_SECRET` entorno `AMPLIFY_AMAZON_CLIENT_ID` y las variables de entorno son OAuth símbolos, no una clave de AWS acceso ni una clave secreta.

Variables de entorno del marco de frontend

Si está desarrollando su aplicación con un marco de frontend que admite sus propias variables de entorno, es importante que comprenda que no son las mismas que las variables de entorno que configura en la consola de Amplify. Por ejemplo, React (con el prefijo REACT_APP) y Gatsby (con el prefijo GATSBY) permiten crear variables de entorno de tiempo de ejecución que esos marcos agrupan automáticamente en la compilación de producción de frontend. Para comprender los efectos del uso de estas variables de entorno para almacenar valores, consulte la documentación del marco de frontend que esté utilizando.

El almacenamiento de valores confidenciales, como las claves de API, dentro de estas variables de entorno prefijadas en el marco de frontend no es una práctica recomendada y no se recomienda en absoluto. Para ver un ejemplo del uso de las variables de entorno de tiempo de compilación de Amplify para este propósito, consulte Acceso a las variables de entorno en el momento de la compilación.

Configuración de variables de entorno

Utilice las siguientes instrucciones para establecer las variables de entorno de una aplicación mediante la consola de Amplify.

> **Note**
>
> Las variables de entorno se pueden ver en el menú de configuración de la aplicación de la consola de Amplify solo cuando se configura una aplicación para una implementación continua y conectada a un repositorio de git. Para obtener instrucciones sobre este tipo de implementación, consulte Primeros pasos con el código existente.

Cómo configurar variables de entorno

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la consola de Amplify, elija Alojamiento y, a continuación, elija Variables de entorno.
3. En Variables de entorno, elija Administrar variables.
4. En Variable, introduzca su clave. En Valor, especifique su valor. De manera predeterminada, la consola de Amplify aplica las variables de entorno en todas las ramificaciones, de manera que no tenga que volver a introducir las variables cuando se conecta a una nueva ramificación.

5. (Opcional) Para personalizar una variable de entorno específica de una ramificación, añada una anulación de ramificación de la siguiente manera:

   a. Elija Acciones y, a continuación, elija Añadir anulación de variable.

   b. Ahora tiene un conjunto de variables de entorno específicas de su ramificación.

6. Seleccione Guardar.

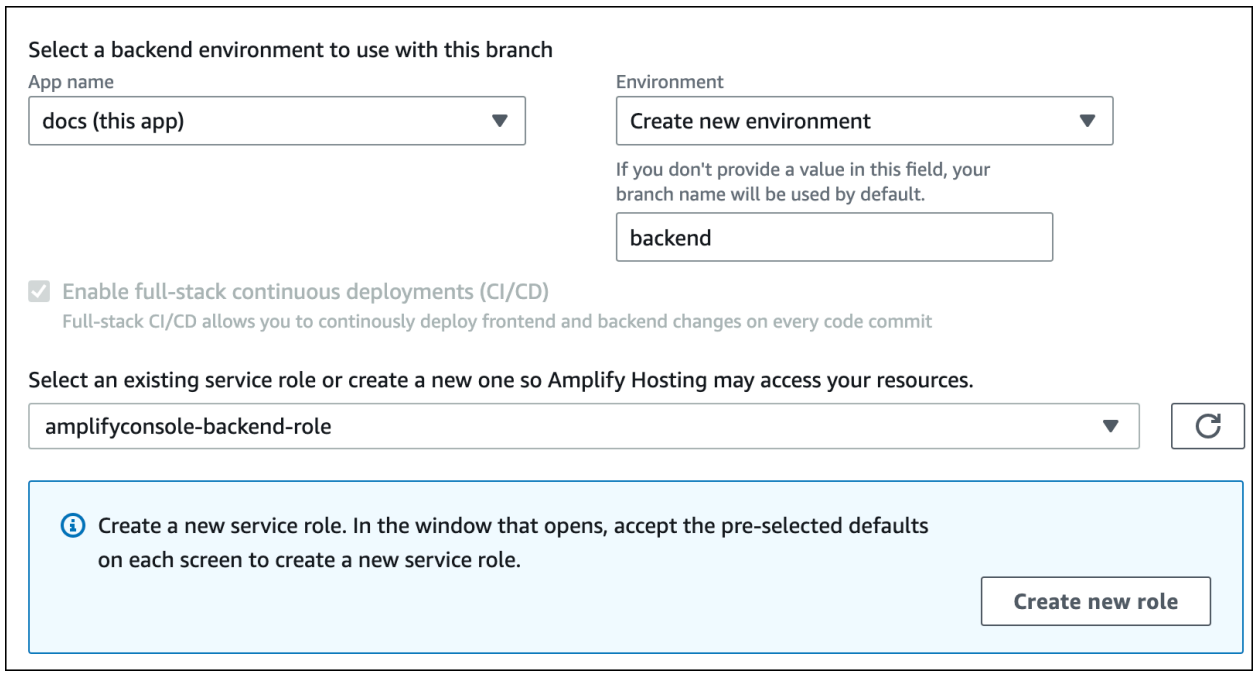
Cree un nuevo entorno de backend con parámetros de autenticación para el inicio de sesión en redes sociales

Para conectar una ramificación a una aplicación

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. El procedimiento para conectar una ramificación a una aplicación varía en función de si se conecta una ramificación a una aplicación nueva o a una aplicación existente.

   - Conexión de una ramificación a una nueva aplicación

     a. En la página de configuración de compilación, busque la sección de selección de un entorno de backend para utilizarlo con esta ramificación. En Entorno, elija Crear un nuevo entorno e introduzca el nombre del entorno de backend. La siguiente captura de pantalla muestra la sección de selección de un entorno de backend para utilizarlo con esta ramificación de la página de configuración de compilación con el nombre de entorno de backend **`backend`** introducido.



b. Amplíe la sección Configuración avanzada de la página Configuración de compilación y añada variables de entorno para las claves de inicio de sesión en redes sociales. Por ejemplo, **`AMPLIFY_FACEBOOK_CLIENT_SECRET`** es una variable de entorno válida. Para ver la lista de variables de entorno del sistema Amplify que están disponibles de forma predeterminada, consulte la tabla de Referencia de variables de entorno de Amplify.

- Conexión de una ramificación a una aplicación existente

a. Si va a conectar una nueva ramificación a una aplicación existente, configure las variables del entorno de inicio de sesión en redes sociales antes de conectar la ramificación. En el panel de navegación, elija Configuración de la aplicación y Variables de entorno.

b. En la sección Variables de entorno, elija Administrar variables.

c. En la sección Administrar variables, elija Añadir variable.

d. En Variable (clave), introduzca su ID de cliente. En Valor, escriba la clave secreta del cliente.

e. Elija Guardar.

Acceso a las variables de entorno en el momento de la compilación

Para acceder a una variable de entorno durante una compilación, edite la configuración de la compilación para incluir la variable de entorno en los comandos de compilación.

Cada comando de la configuración de compilación se ejecuta dentro de un intérprete de comandos Bash. Para obtener más información sobre cómo trabajar con variables de entorno en Bash, consulte las expansiones del intérprete de comandos en el manual de GNU Bash.

Para editar la configuración de compilación con el fin de incluir una variable de entorno

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la consola de Amplify, elija Alojamiento y, a continuación, elija Configuración de compilación.
3. En la sección de especificación de compilación de aplicaciones, elija Editar.
4. Añada la variable de entorno a su comando de compilación. Ahora debe poder acceder a la variable de entorno durante la siguiente compilación. Este ejemplo cambia el comportamiento del npm (BUILD_ENV) y agrega un token de API (TWITCH_CLIENT_ID) para un servicio externo a un archivo de entorno para su uso posterior.

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

5. Seleccione Guardar.

Conversión de las variables de entorno en accesibles para los tiempos de ejecución del servidor

De forma predeterminada, un componente de servidor de Next.js no tiene acceso a las variables de entorno de su aplicación. Este comportamiento tiene como objetivo proteger cualquier secreto almacenado en las variables de entorno que utilice su aplicación durante la fase de compilación.

Para que Next.js pueda acceder a variables de entorno específicas, debe modificar el archivo de especificaciones de compilación de Amplify para establecer las variables de entorno en los archivos de entorno que reconoce Next.js. Esto permite a Amplify cargar las variables de entorno antes de compilar la aplicación. Para obtener más información sobre cómo modificar la especificación de

compilación, consulte los ejemplos de cómo añadir variables de entorno en la sección de comandos de compilación.

Administración de los secretos de entorno

Con el lanzamiento de Amplify Gen 2, el flujo de trabajo de los secretos de entorno se optimiza para centralizar la administración de los secretos y las variables de entorno en la consola de Amplify. Para obtener instrucciones sobre cómo configurar y acceder a los secretos de una aplicación de Amplify Gen 2, consulte Secrets and environment vars en la documentación de Amplify.

Los secretos de entorno de una aplicación de Gen 1 son similares a las variables de entorno, pero son pares de valor clave del almacén de parámetros de AWS Systems Manager que se pueden cifrar. Algunos valores deben estar cifrados, como la clave privada del inicio de sesión con Apple de Amplify.

Se usa AWS Systems Manager para establecer los secretos del entorno para una aplicación Amplify Gen 1

Usa las siguientes instrucciones para establecer un secreto de entorno para una aplicación Amplify de primera generación mediante la AWS Systems Manager consola.

Para establecer un secreto de entorno

1. Inicie sesión en la AWS Systems Manager consola AWS Management Console y ábrala.
2. En el panel de navegación, elija Administración de aplicaciones y, a continuación, elija Almacén de parámetros.
3. En la página Almacén de parámetros de AWS Systems Manager, elija Crear parámetro.
4. En la página Crear parámetro, en la sección Detalles de parámetro, haga lo siguiente:
    a. En Nombre, introduzca un parámetro con el formato **`/amplify/{your_app_id}/{your_backend_environment_name}/{your_parameter_name}`**.
    b. En Type (Tipo), elija SecureString.
    c. En Fuente de clave de KMS, elija Mi cuenta actual para utilizar la clave predeterminada de su cuenta.
    d. En Valor, introduzca el valor secreto para cifrarlo.
5. Elija Crear parámetro.

> **Note**
>
> Amplify solo tiene acceso a las claves de la compilación del entorno específico de `/amplify/{your_app_id}/{your_backend_environment_name}`. Debe especificar el valor predeterminado AWS KMS key para permitir que Amplify descifre el valor.

Acceso a los secretos de entorno de una aplicación de Gen 1

El acceso al secreto de entorno de una aplicación de Gen 1 durante una compilación es similar al acceso a las variables de entorno, excepto que los secretos de entorno se almacenan en una cadena JSON de `process.env.secrets`.

Referencia de secretos de entorno de Amplify

Especifique un parámetro de Systems Manager en el formato `/amplify/{your_app_id}/{your_backend_environment_name}/AMPLIFY_SIWA_CLIENT_ID`.

Puede utilizar los siguientes secretos de entorno a los que se puede acceder de forma predeterminada en la consola de Amplify.

Nombre de variable	Descripción	Ejemplo de valor
AMPLIFY_SIWA_CLIENT_ID	ID de inicio de sesión con ID de cliente SignInWithApple	`com.yourapp.auth`
AMPLIFY_SIWA_TEAM_ID	ID de inicio de sesión con el equipo de Apple	`ABCD123`
AMPLIFY_SIWA_KEY_ID	ID clave de inicio de sesión con clave de Apple	`ABCD123`
AMPLIFY_SIWA_PRIVATE_KEY	Clave privada de inicio de sesión con Apple	-----INICIAR CLAVE PRIVADA----- ****..... -----FINALIZAR CLAVE PRIVADA-----

Configuración de encabezados HTTP personalizados para una aplicación de Amplify

Los encabezados HTTP personalizados le permiten especificar encabezados para todas las respuestas de HTTP. Los encabezados de respuesta se pueden utilizar para fines de depuración, seguridad e información. Puede especificar encabezados en la consola de Amplify, o bien al descargar y editar el archivo `customHttp.yml` de una aplicación y guardarlo en el directorio raíz del proyecto. Para obtener procedimientos detallados, consulte Configuración de encabezados personalizados.

Anteriormente, los encabezados HTTP personalizados de una aplicación se especificaban al editar las especificaciones de compilación (buildspec) en la consola de Amplify o al descargar el archivo `amplify.yml`, actualizarlo y guardarlo en el directorio raíz del proyecto. Recomendamos que migre los encabezados personalizados especificados de esta manera fuera del buildspec y del archivo `amplify.yml`. Para obtener instrucciones, consulte Migrar encabezados personalizados fuera de la especificación de compilación y amplify.yml.

Temas

Referencia de encabezados personalizados YAML

Especifique los encabezados personalizados con el siguiente formato YAML:

```
customHeaders:
  - pattern: '*.json'
    headers:
    - key: 'custom-header-name-1'
      value: 'custom-header-value-1'
    - key: 'custom-header-name-2'
      value: 'custom-header-value-2'
  - pattern: '/path/*'
    headers:
```

```
    - key: 'custom-header-name-1'
      value: 'custom-header-value-2'
```

Para un monorepo, use el siguiente formato YAML:

```
applications:
  - appRoot: app1
    customHeaders:
    - pattern: '**/*'
      headers:
      - key: 'custom-header-name-1'
        value: 'custom-header-value-1'
  - appRoot: app2
    customHeaders:
    - pattern: '/path/*.json'
      headers:
      - key: 'custom-header-name-2'
        value: 'custom-header-value-2'
```

Cuando añada encabezados personalizados a su aplicación, deberá especificar sus propios valores para los siguientes aspectos:

pattern

Los encabezados personalizados se aplican a todas las rutas de archivo de URL que coinciden con el patrón.

headers

Definen los encabezados que coinciden con el patrón de archivo.

clave

El nombre del encabezado personalizado.

valor

El valor del encabezado personalizado.

Para obtener más información acerca de los encabezados HTTP, consulte la lista de encabezados HTTP de Mozilla.

Configuración de encabezados personalizados

Existen dos formas de especificar los encabezados HTTP personalizados en una aplicación de Amplify. Puede especificar los encabezados en la consola de Amplify o bien descargar y editar el archivo `customHttp.yml` de la aplicación, que deberá guardar en el directorio raíz del proyecto.

Cómo configurar encabezados personalizados de una aplicación y guardarlos en la consola

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea configurar encabezados personalizados.
3. En el panel de navegación, seleccione Alojamiento y, a continuación, seleccione Encabezados personalizados.
4. En la página Encabezados personalizados, seleccione Editar.
5. En la ventana Editar encabezados personalizados, introduzca la información de los encabezados personalizados con formato de encabezado personalizado YAML.
   a. En `pattern`, introduzca el patrón de coincidencia.
   b. En `key`, ingrese el nombre del encabezado personalizado.
   c. En `value`, ingrese el valor del encabezado personalizado.
6. Elija Guardar.
7. Vuelva a implementar la aplicación para aplicar los nuevos encabezados personalizados.
   - Para una aplicación de CI/CD, desplácese a la ramificación que desea implementar y eligir Volver a implementar esta versión. También puede realizar una nueva compilación desde su repositorio de Git.
   - Para una aplicación de implementación manual, vuelva a implementar la aplicación en la consola de Amplify.

Cómo configurar encabezados personalizados de una aplicación y guardarlos en la raíz del repositorio

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea configurar encabezados personalizados.
3. En el panel de navegación, seleccione Alojamiento y, a continuación, seleccione Encabezados personalizados.
4. En la página Encabezados personalizados, seleccione Descargar YML.

5. Abra el archivo `customHttp.yml` descargado en su editor de código preferido e introduzca la información de los encabezados personalizados con formato de encabezado personalizado YAML.

   a. En `pattern`, introduzca el patrón de coincidencia.

   b. En `key`, ingrese el nombre del encabezado personalizado.

   c. En `value`, ingrese el valor del encabezado personalizado.

6. Guarde el archivo `customHttp.yml` editado en el directorio raíz de su proyecto. Si está trabajando con un monorepo, guarde el archivo `customHttp.yml` en la raíz de su repositorio.

7. Vuelva a implementar la aplicación para aplicar los nuevos encabezados personalizados.

   - Para una aplicación de CI/CD, lleve a cabo una nueva compilación desde su repositorio de Git que incluya el nuevo archivo `customHttp.yml`.
   - Para una aplicación de implementación manual, vuelva a implementar la aplicación en la consola de Amplify e incluya el nuevo archivo `customHttp.yml` con los artefactos a cargar.

> **Note**
>
> Los encabezados personalizados configurados en el archivo `customHttp.yml` e implementados en el directorio raíz de la aplicación anularán los encabezados personalizados definidos en la sección Encabezados personalizados de la consola de Amplify.

Ejemplo de encabezados personalizados de seguridad

Los encabezados personalizados de seguridad permiten aplicar HTTPS, evitar ataques XSS y defender su navegador frente a ataques tipo clickjack. Use la siguiente sintaxis de YAML para aplicar encabezados de seguridad personalizados a su aplicación.

```
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'"
```

Configuración de encabezados Cache-Control personalizados

Las aplicaciones alojadas en Amplify respetan los encabezados `Cache-Control` que envía el origen, a menos que los anule al definir encabezados personalizados. Amplify solo aplica encabezados personalizados Cache-Control para las respuestas correctas con un código de estado `200 OK`. Esto evita que las respuestas de error se almacenen en caché y se distribuyen a otros usuarios que hagan la misma solicitud.

Puede ajustar manualmente la directiva `s-maxage` para tener más control sobre el rendimiento y la disponibilidad de implementación de la aplicación. Por ejemplo, para aumentar el tiempo que el contenido permanece almacenado en caché en la periferia, puede aumentar manualmente el tiempo de vida (TTL) actualizando `s-maxage` a un valor superior al predeterminado de 600 segundos (10 minutos).

Para especificar un valor personalizado para `s-maxage`, utilice el siguiente formato YAML. Este ejemplo mantiene el contenido asociado en caché en la periferia durante 3600 segundos (una hora).

```
customHeaders:
  - pattern: '/img/*'
    headers:
      - key: 'Cache-Control'
        value: 's-maxage=3600'
```

Para obtener más información sobre cómo controlar el rendimiento de las aplicaciones con encabezados, consulte Uso del encabezado Cache-Control para aumentar el rendimiento de la aplicación.

Migrar encabezados personalizados fuera de la especificación de compilación y amplify.yml

Anteriormente, los encabezados HTTP personalizados de una aplicación se especificaban al editar la especificación de compilación en la consola de Amplify o al descargar el archivo `amplify.yml`,

actualizarlo y guardarlo en el directorio raíz del proyecto. Se recomienda encarecidamente migrar los encabezados personalizados fuera de la especificación de compilación y del archivo `amplify.yml`.

Especifique sus encabezados personalizados en la sección Encabezados personalizados de la consola de Amplify o al descargar y editar el archivo `customHttp.yml`.

Para migrar los encabezados personalizados almacenados en la consola de Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación cuyos encabezados personalizados desee migrar.
3. En el panel de navegación, elija Alojamiento y, a continuación, Configuración de compilación. En la sección Especificaciones de compilación de la aplicación, revise las especificaciones de compilación de su aplicación.
4. Elija Descargar para guardar una copia de sus especificaciones de compilación actuales. Podrá consultar esta copia más adelante si necesita recuperar alguna configuración.
5. Cuando haya terminado la descarga, elija Editar.
6. Tome nota de la información del encabezado personalizado del archivo, ya que deberá usarla en el paso 9. En la ventana Editar, elimine los encabezados personalizados del archivo y elija Guardar.
7. En el panel de navegación, elija Alojamiento y Encabezados personalizados.
8. En la página Encabezados personalizados, seleccione Editar.
9. En la ventana Editar encabezados personalizados, introduzca la información de los encabezados personalizados que eliminó en el paso 6.
10. Seleccione Guardar.
11. Vuelva a implementar aquellas ramificaciones en las que desee aplicar los nuevos encabezados personalizados.

Para migrar los encabezados personalizados de amplify.yml a customHttp.yml

1. Acceda al archivo `amplify.yml`, implementado actualmente en el directorio raíz de su aplicación.
2. Abra el archivo `amplify.yml` con el editor de código que prefiera.
3. Tome nota de la información del encabezado personalizado del archivo, ya que deberá usarla en el paso 8. Elimine los encabezados personalizados del archivo. Guarde y cierre el archivo.
4. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.

5. Elija la aplicación para la que desea configurar encabezados personalizados.
6. En el panel de navegación, elija Alojamiento y Encabezados personalizados.
7. En la página Encabezados personalizados, seleccione Descargar.
8. Abra el archivo `customHttp.yml` que ha descargado en su editor de código favorito e introduzca la información de los encabezados personalizados que eliminó de `amplify.yml` en el paso 3.
9. Guarde el archivo `customHttp.yml` editado en el directorio raíz de su proyecto. Si está trabajando con un monorepo, guarde el archivo en la raíz de su repositorio.
10. Vuelva a implementar la aplicación para aplicar los nuevos encabezados personalizados.
    - Para una aplicación de CI/CD, lleve a cabo una nueva compilación desde su repositorio de Git que incluya el nuevo archivo `customHttp.yml`.
    - Para una aplicación de implementación manual, vuelva a implementar la aplicación en la consola de Amplify e incluya el nuevo archivo `customHttp.yml` con los artefactos que subas.

> **Note**
>
> Los encabezados personalizados configurados en el archivo `customHttp.yml` e implementados en el directorio raíz de la aplicación anularán los encabezados personalizados definidos en la sección Encabezados personalizados de la consola de Amplify.

Requisitos de encabezados personalizados en monorepo

Para especificar encabezados personalizados en una aplicación en monorepo, deberá cumplir los siguientes requisitos de configuración:

- Dispone de un formato YAML específico para monorepo. Para consultar la sintaxis correcta, acceda a Referencia de encabezados personalizados YAML.
- Puede especificar encabezados personalizados para una aplicación en monorepo en la sección Encabezados personalizados de la consola de Amplify. Deberá volver a implementar su aplicación para aplicar los nuevos encabezados personalizados.
- Como alternativa al uso de la consola, puede especificar encabezados personalizados para una aplicación en monorepo en un archivo `customHttp.yml`. Deberá guardar el archivo

`customHttp.yml` en la raíz de su repositorio y, a continuación, volver a implementar la aplicación para aplicar los nuevos encabezados personalizados. Los encabezados personalizados especificados en el archivo `customHttp.yml` anularán los encabezados personalizados especificados en la sección Encabezados personalizados de la consola de Amplify.

Creación de un webhook entrante para iniciar una compilación

Configure un webhook entrante en la consola de Amplify para iniciar una compilación sin confirmar código en su repositorio de Git. Puede usar los webhooks con herramientas CMS sin periféricos (como Contentful o GraphCMS) para iniciar compilaciones cuando haya cambios o compilaciones diarias mediante servicios como Zapier.

Para crear un webhook entrante

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea crear un webhook.
3. En el panel de navegación, elija Hosting y, a continuación, Configuración de compilación.
4. En la página Configuración de compilación, desplácese hasta la sección Webhooks entrantes y elija Crear webhook.
5. En el cuadro de diálogo Crear webhook, haga lo siguiente:
    a. En Nombre de webhook, introduzca un nombre para el webhook.
    b. En Ramificación a compilar, seleccione la ramificación que desea compilar con las solicitudes de webhook entrantes.
    c. Elija Crear webhook.
6. En la sección Webhooks entrantes, lleve a cabo una de las siguientes acciones:
    - Copie la URL del webhook y envíela a una herramienta CMS sin periféricos u otro servicio para iniciar las compilaciones.
    - Ejecute el comando curl en una ventana de terminal para iniciar una nueva compilación.

Supervisión de una aplicación de Amplify

AWS Amplify proporciona dos funciones para supervisar las aplicaciones alojadas desde la consola Amplify.

- Amplify emite métricas a través de Amazon CloudWatch que puedes usar para monitorear el tráfico, los errores, la transferencia de datos y la latencia de tus aplicaciones.
- Amplify ofrece registros de acceso con información detallada sobre las solicitudes realizadas a su aplicación.

Use los temas de esta sección para aprender a usar CloudWatch las métricas y los registros de acceso de Amplify para monitorear sus aplicaciones.

Temas

Supervisión de una aplicación con Amazon CloudWatch

AWS Amplify está integrado con Amazon CloudWatch, lo que te permite monitorizar las métricas de tus aplicaciones de Amplify prácticamente en tiempo real. Puede crear alarmas que envíen notificaciones cuando una métrica supere el umbral que haya establecido. Para obtener más información sobre el funcionamiento del CloudWatch servicio, consulta la Guía del CloudWatch usuario de Amazon.

CloudWatch Métricas compatibles

Amplify admite seis CloudWatch métricas en el espacio de `AWS/AmplifyHosting` nombres para supervisar el tráfico, los errores, la transferencia de datos y la latencia de tus aplicaciones. Estas métricas se agregan en intervalos de un minuto. CloudWatch las métricas de monitoreo son gratuitas y no se tienen en cuenta para las cuotas CloudWatch de servicio.

No todas las estadísticas son aplicables a todas las métricas. En la tabla siguiente, se muestran las estadísticas más relevantes en la descripción de cada métrica.

Métricas	Descripción
Solicitudes	El número total de solicitudes de usuarios recibidas por su aplicación. La estadística más relevante es `Sum`. Utilice la estadística `Sum` para obtener el número total de solicitudes.
BytesDownloaded	La cantidad total de datos transferidos desde su aplicación (descargados) en bytes por los espectadores para las solicitudes `GET`, `HEAD` y `OPTIONS`. La estadística más relevante es `Sum`.
BytesUploaded	La cantidad total de datos transferidos a su aplicación (cargados) en bytes en cualquier solicitud, incluidos los encabezados. Amplify no le cobra por los datos cargados en sus aplicaciones. La estadística más relevante es `Sum`.
4xxErrors	El número de solicitudes que devolvieron un error en el rango de código de estado HTTP 400-499. La estadística más relevante es `Sum`. Utilice la estadística `Sum` para obtener el número total de apariciones de estos errores.
5xxErrors	El número de solicitudes que devolvieron un error en el rango de código de estado HTTP 500-599.

Métricas	Descripción
	La estadística más relevante es `Sum`. Utilice la estadística `Sum` para obtener el número total de apariciones de estos errores.
Latencia	El tiempo transcurrido hasta el primer byte en segundos. Este es el tiempo total entre el momento en que Amplify Hosting recibe una solicitud y cuando devuelve una respuesta a la red. Esto no incluye la latencia de la red para que una respuesta llegue al dispositivo del espectador. Las estadísticas más relevantes son `Average`, `Maximum`, `Minimum`, `p10`, `p50`, `p90`, `p95` y `p100`. Utilice la estadística `Average` para evaluar las latencias previstas.

Amplify proporciona las siguientes dimensiones CloudWatch métricas.

Dimensión	Descripción
Aplicación	Los datos métricos los proporciona la aplicación.
Cuenta de AWS	Los datos métricos se proporcionan en todas las aplicaciones de Cuenta de AWS.

Acceder a CloudWatch las métricas

Puede acceder a CloudWatch las métricas directamente desde la consola de Amplify mediante el siguiente procedimiento.

> **Note**
>
> También puede acceder a CloudWatch las métricas en el AWS Management Console at https://console.aws.amazon.com/cloudwatch/.

Para obtener acceso a las métricas en la consola de Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea ver las métricas.
3. En el panel de navegación, elija Configuración de la aplicación y Supervisión.
4. En la página Supervisión, elija Métricas.

Crear alarmas CloudWatch

Puede crear CloudWatch alarmas en la consola Amplify que envíen notificaciones cuando se cumplan criterios específicos. Una alarma vigila una única CloudWatch métrica y envía una notificación de Amazon Simple Notification Service cuando la métrica supera el umbral durante un número específico de períodos de evaluación.

Puede crear alarmas más avanzadas que utilicen expresiones matemáticas métricas en la CloudWatch consola o mediante el CloudWatch APIs. Por ejemplo, puede crear una alarma que le notifique cuando el porcentaje de 4xxErrors supera el 15% durante tres períodos consecutivos. Para obtener más información, consulte Creación de una CloudWatch alarma basada en una expresión matemática métrica en la Guía del CloudWatch usuario de Amazon.

El CloudWatch precio estándar se aplica a las alarmas. Para obtener más información, consulta los CloudWatchprecios de Amazon.

Utilice el siguiente procedimiento para crear una alarma en la consola de Amplify.

Para crear una CloudWatch alarma para una métrica de Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación en la que desea configurar una alarma.
3. En el panel de navegación, elija Configuración de la aplicación y Supervisión.
4. En la página de supervisión, elija Alarmas.

5. Elija Crear alarma.
6. En la ventana Crear alarma, configure la alarma de la siguiente manera:

   a. En Métrica, elija el nombre de la métrica que se va a supervisar de la lista.

   b. En Nombre de la alarma, escriba un nombre significativo para la alarma. Por ejemplo, si está supervisando Solicitudes, puede asignar un nombre a la alarma**`HighTraffic`**. El nombre solo debe contener caracteres ASCII.

   c. Para Configurar notificaciones, realice una de las siguientes acciones:

      - i. Elija Nuevo para crear un nuevo tema de Amazon SNS.
        ii. En Dirección de correo electrónico, introduzca la dirección de correo electrónico del destinatario de las notificaciones.
        iii. Elija Añadir nueva dirección de correo electrónico para añadir destinatarios adicionales.
      - i. Elija Existente para reutilizar un tema de Amazon SNS.
        ii. En Tema de SNS, seleccione el nombre de un tema Amazon SNS existente de la lista.

   d. En Siempre que la Estadística de la Métrica, configure las condiciones de la alarma de la siguiente manera:

      i. Especifique si la métrica debe ser mayor, menor o igual al valor del umbral.
      ii. Especifique el valor del umbral.
      iii. Especifique el número de periodos de evaluación consecutivos que deben estar en estado de alarma para invocar la alarma.
      iv. Especifique la duración del periodo de tiempo de evaluación.

   e. Elija Crear alarma.

> **Note**
>
> Cada destinatario de Amazon SNS que especifique, recibe un mensaje de correo electrónico de confirmación de notificaciones de AWS . El mensaje de correo electrónico contiene un enlace que el destinatario debe seguir para confirmar su suscripción y recibir notificaciones.

Acceder a CloudWatch los registros de las aplicaciones SSR

Amplify envía información sobre su tiempo de ejecución de Next.js a Amazon CloudWatch Logs en su. Cuenta de AWS Al implementar una aplicación SSR, la aplicación requiere un rol de servicio de IAM que Amplify asume cuando llama a otros servicios en su nombre. Puede permitir que el procesamiento de Amplify Hosting cree automáticamente un rol de servicio en su lugar, o puede especificar un rol que haya creado usted.

Si decides permitir que Amplify cree un rol de IAM para ti, el rol ya tendrá los permisos para crear registros. CloudWatch Si creas tu propia función de IAM, tendrás que añadir los siguientes permisos a tu política para permitir que Amplify acceda a Amazon CloudWatch Logs.

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

Para obtener más información sobre cómo añadir un rol de servicio, consulte. Añadir un rol de servicio con permisos para implementar recursos de backend Para obtener más información acerca cómo implementar aplicaciones renderizadas en el servidor, consulte Implementación de aplicaciones renderizadas del servidor con Amplify Hosting.

Supervisión de los registros de acceso a las aplicaciones

Amplify almacena los registros de acceso de todas las aplicaciones que aloja en Amplify. Los registros de acceso contienen información sobre las solicitudes realizadas a sus aplicaciones alojadas. Amplify retiene todos los registros de acceso de una aplicación hasta que la elimine. Todos los registros de acceso de una aplicación están disponibles en la consola de Amplify. Sin embargo, cada solicitud individual de registros de acceso se limita a un período de dos semanas que especifique.

Amplify nunca reutiliza las CloudFront distribuciones entre clientes. Amplify crea CloudFront distribuciones por adelantado para que no tengas que esperar a que se cree una CloudFront distribución al implementar una nueva aplicación. Antes de asignar estas distribuciones a una aplicación de Amplify, es posible que reciban tráfico de bots. Sin embargo, están configuradas para responder siempre como No encontradas antes de ser asignadas. Si los registros de acceso de la aplicación contienen entradas de un periodo de tiempo anterior a la creación de la aplicación, estas entradas están relacionadas con esta actividad.

> **Important**
>
> Recomendamos utilizar los registros de acceso para comprender la naturaleza de las solicitudes hechas a su contenido y no como una relación exhaustiva de todas las solicitudes. Amplify entrega registros de acceso en la medida en que sea posible. La entrada de registro de una solicitud determinada puede entregarse mucho después de la solicitud se haya procesado realmente y, en casos contados, es probable que una entrada de registro no se entregue en absoluto. Cuando se omite una entrada de registro de los registros de acceso, el número de entradas de los registros de acceso no coincidirá con el uso que aparece en los informes de AWS facturación y uso.

Recuperación de los registros de acceso de una aplicación

Utilice el siguiente procedimiento para recuperar registros de acceso de una aplicación de Amplify.

Para ver los registros de acceso

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea ver los registros de acceso.
3. En el panel de navegación, seleccione Alojamiento y, a continuación, seleccione Supervisión.
4. En la página Supervisión, elija Registros de acceso.
5. Elija Editar rango de tiempo.
6. En la ventana Editar intervalo de tiempo, haga lo siguiente.
    a. En Fecha de inicio, especifique el primer día del intervalo de dos semanas para recuperar los registros.
    b. En Fecha de inicio, elija la hora del primer día para iniciar la recuperación de los registros.
    c. Elija Confirmar.
7. La consola de Amplify muestra los registros del rango de tiempo especificado en la sección Registros de acceso. Elija Descargar para guardar los registros en formato CSV.

Análisis de registros de acceso

Para analizar los registros de acceso, puede guardar los archivos CSV en un bucket de Amazon S3. Una forma de analizar los registros de acceso consiste en utilizar Athena. Athena es un servicio de

consultas interactivo que puede ayudarlo a analizar los datos de los AWS servicios. Puede seguir las step-by-step instrucciones que aparecen aquí para crear una tabla. Una vez creada la tabla, puede consultar los datos del siguiente modo.

```
SELECT SUM(bytes) AS total_bytes
FROM logs
WHERE "date" BETWEEN DATE '2018-06-09' AND DATE '2018-06-11'
LIMIT 100;
```

Registro de llamadas a la API de Amplify mediante AWS CloudTrail

AWS Amplify está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en Amplify. CloudTrail captura todas las llamadas a la API de Amplify como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de Amplify y las llamadas desde el código a las operaciones de la API de Amplify. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de Amplify. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información que CloudTrail recopila, puede determinar la solicitud que se realizó a Amplify, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía del AWS CloudTrail usuario.

Amplify la información en CloudTrail

CloudTrail está activado en tu AWS cuenta de forma predeterminada. Cuando se produce una actividad en Amplify, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS . Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos en la Guía del AWS CloudTrail usuario.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de Amplify, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS . La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos

de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte lo indicado en la Guía del usuario de AWS CloudTrail :

- Crear una ruta para tu AWS cuenta
- CloudTrail servicios e integraciones compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas las operaciones de Amplify se registran CloudTrail y documentan en la Referencia de la API de la AWS Amplify consola, la Referencia de la API de la interfaz de usuario de AWS Amplify Admin y la Referencia de la API de Amplify UI Builder. Por ejemplo, las llamadas a las `CreateApp` `DeleteBackendEnvironment` operaciones `DeleteApp` y generan entradas en los CloudTrail archivos de registro.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- La solicitud se realizó con credenciales de usuario root o AWS Identity and Access Management (IAM).
- Si la solicitud se ha realizado con credenciales de seguridad temporales de un rol o de un usuario federado.
- Fue la solicitud realizada por otro AWS servicio.

Para obtener más información, consulte el elemento CloudTrail UserIdentity en la Guía del usuario.AWS CloudTrail

Interpretación de las entradas de archivos de registro de Amplify

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que muestra la ListAppsoperación de referencia de la API de la AWS Amplify consola.


```
{
    "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"
}
```

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que muestra la ListBackendJobsoperación de referencia de la API de la interfaz de usuario de AWS Amplify administración.


```
{
    "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"
}
```

Notificaciones por correo electrónico para compilaciones

Puedes configurar notificaciones por correo electrónico para una AWS Amplify aplicación para alertar a las partes interesadas o a los miembros del equipo cuando una compilación tenga éxito o fracase. Amplify Hosting crea un tema de Amazon Simple Notification Service (SNS) en su cuenta y lo usa para configurar las notificaciones por correo electrónico. Estas notificaciones se pueden configurar para que se apliquen a todas las ramificaciones o solo a ramificaciones concretas de una aplicación de Amplify.

Configuración de las notificaciones por correo electrónico

Proceda de la siguiente manera para configurar notificaciones por correo electrónico para todas las ramificaciones o ramificaciones concretas de una aplicación de Amplify.

Para configurar las notificaciones por correo electrónico de una aplicación de Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desee configurar las notificaciones por correo electrónico.
3. En el panel de navegación, elija Alojamiento, Notificaciones de compilación. En la página Notificaciones de compilación, elija Habilitar notificaciones.
4. En la página Administrar notificaciones, seleccione Agregar nueva.
5. Realice una de las siguientes acciones:
   - Para enviar notificaciones de una sola ramificación, introduzca en Correo electrónico la dirección de correo a la que desea enviar las notificaciones. En Ramificación, seleccione el nombre de la ramificación cuyas notificaciones desea enviar.
   - Para enviar notificaciones de todas las ramificaciones conectadas, introduzca en Correo electrónico la dirección de correo a la que desea enviar las notificaciones. En Ramificación, elija Todas las ramificaciones.
6. Seleccione Guardar.

Personalización de la imagen de compilación

Puede usar una imagen de compilación personalizada para proporcionar un entorno de compilación personalizado a una aplicación de Amplify. Si tiene dependencias específicas cuya instalación tarda mucho durante una compilación con el contenedor predeterminado de Amplify, puede crear su propia imagen de Docker y hacer referencia a esta durante una compilación. Las imágenes se pueden alojar en Amazon Elastic Container Registry Public.

Para que una imagen de compilación personalizada sirva como imagen de compilación de Amplify, debe cumplir los siguientes requisitos.

Requisitos de las imágenes de compilación personalizadas

1. Una distribución de Linux compatible con la biblioteca GNU C (glibc), como Amazon Linux, compilada para arquitectura x86-64.
2. cURL: al lanzar su imagen personalizada, descargamos nuestro ejecutor de compilación en su contenedor y, por lo tanto, necesitamos que cURL esté presente. Si falta esta dependencia, se produce un error en la compilación de forma instantánea sin ninguna salida, ya que nuestro ejecutor de compilación no puede producir ninguna.
3. Git: para clonar su repositorio de Git, necesitamos que Git se instale en la imagen. Si falta esta dependencia, se producirá un error en el paso Repositorio de clonación.
4. OpenSSH: para clonar de forma segura su repositorio, necesitamos OpenSSH para configurar la clave SSH temporalmente durante la compilación. El paquete OpenSSH proporciona los comandos que necesita el ejecutor de compilación.
5. Bash y The Bourne Shell: estas dos utilidades se utilizan para ejecutar comandos en el momento de la compilación. Si no están instaladas, es posible que las compilaciones fallen antes de empezar.
6. Node.JS+NPM: nuestro ejecutor de compilación no instala Node. Node y NPM deben estar instalados en la imagen. Esto solo es necesario en el caso de las compilaciones que, a su vez, necesitan paquetes NPM o comandos específicos de Node. Sin embargo, recomendamos encarecidamente instalarlos porque, cuando están presentes, el ejecutor de compilación de Amplify puede utilizar estas herramientas para mejorar la ejecución de la compilación. La característica de anulación de paquetes de Amplify usa NPM para instalar el paquete extendido por Hugo cuando establece una anulación para Hugo.

Los siguientes paquetes no son obligatorios, pero recomendamos encarecidamente que los instale.

1. NVM (Node Version Manager): Le recomendamos que instale este administrador de versiones si necesita gestionar diferentes versiones de Node. Cuando configuras una anulación, la función de anulación de paquetes de Amplify utiliza NVM para cambiar las versiones de Node.js antes de cada compilación.
2. Wget: Amplify puede usar el Wget utilidad para descargar archivos durante el proceso de compilación. Le recomendamos que la instale en su imagen personalizada.
3. Tar: Amplify puede usar el Tar utilidad para descomprimir los archivos descargados durante el proceso de compilación. Le recomendamos que la instale en su imagen personalizada.

Configuración de una imagen de compilación personalizada de una aplicación

Utilice el siguiente procedimiento para configurar una imagen de compilación personalizada de una aplicación en la consola de Amplify.

Para configurar una imagen de compilación personalizada alojada en Amazon ECR

1. Consulte Introducción en la Guía del usuario de Amazon ECR Public para configurar un repositorio de Amazon ECR Public con una imagen de Docker.
2. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
3. Elija la aplicación para la que desea configurar una imagen de compilación personalizada.
4. En el panel de navegación, elija Alojamiento y, a continuación, Configuración de compilación.
5. En la página Configuración de compilación, en la sección Configuración de imagen de compilación, elija Editar.
6. En la página Editar configuración de la imagen de compilación, abra el menú Imagen de compilación y elija Imagen de compilación personalizada.
7. Introduzca el nombre del repositorio de Amazon ECR Public que creó en el paso uno. Aquí es donde se aloja la imagen de compilación. Por ejemplo, si el nombre de su repositorio es ecr-examplerepo, deberá introducir **`public.ecr.aws/xxxxxxxx/ecr-examplerepo`**.
8. Seleccione Guardar.

Uso de versiones específicas de paquetes y dependencias en la imagen de compilación

Las actualizaciones de paquete en directo le permiten especificar versiones de paquetes y dependencias para usarlas en la imagen de compilación predeterminada de Amazon. La imagen de compilación predeterminada viene con varios paquetes y dependencias preinstalados (p. ej., Hugo, CLI de Amplify, Yarn, etc.). Las actualizaciones de paquetes en directo le permiten reemplazar la versión de estas dependencias y especificar una versión concreta o garantizar siempre la instalación de la versión más reciente.

Si las actualizaciones de paquetes en directo están habilitadas, antes de que se ejecute su compilación, el ejecutor de compilación actualizará primero las dependencias especificadas (o cambiará a una versión más antigua de las mismas). Esto aumenta el tiempo de compilación, que es proporcional al tiempo que tardan las dependencias en actualizarse, pero tiene la ventaja de que puede garantizar que se use la misma versión de una dependencia para compilar su aplicación.

> **Warning**
>
> Si se establece la versión de Node.js en la más reciente, se producen errores en las compilaciones. En su lugar, debe especificar una versión exacta de Node.js, como `18`, `21.5` o `v0.1.2`.

Para configurar las actualizaciones de paquetes en directo

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea configurar las actualizaciones de paquetes en directo.
3. En el panel de navegación, elija Alojamiento y, a continuación, Configuración de compilación.
4. En la página Configuración de compilación, en la sección Configuración de imagen de compilación, elija Editar.
5. En la página Editar configuración de la imagen de compilación, ubique la lista Actualizaciones de paquetes en vivo y seleccione Agregar nuevo.
6. En Paquete, seleccione la dependencia que desee anular.
7. En Versión, mantenga la última versión predeterminada o introduzca una versión específica de la dependencia. Si usa última, la dependencia siempre se actualizará a la última versión disponible.

8. Elija Guardar.

Administrar la configuración de caché de una aplicación

Amplify utiliza Amazon CloudFront para gestionar la configuración de almacenamiento en caché de las aplicaciones alojadas. Se aplica una configuración de caché a cada aplicación para optimizarla y obtener el mejor rendimiento.

El 13 de agosto de 2024, Amplify publicó mejoras en la eficiencia del almacenamiento en caché de las aplicaciones. Para obtener más información, consulte Mejoras en el almacenamiento en caché de la CDN para mejorar el rendimiento de las aplicaciones con el alojamiento. AWS Amplify

En la siguiente tabla se resume la compatibilidad de Amplify con determinados comportamientos de almacenamiento en caché antes y después de la publicación de mejoras de almacenamiento en caché.

Comportamiento del almacenamiento en caché	Compatibilidad anterior	Con mejoras en el almacenamiento en caché
Puede agregar encabezados personalizados de una aplicación en la consola de Amplify o en un archivo `customHeaders.yaml` . Uno de los encabezados que puede anular es `Cache-Control` . Para obtener más información, consulte Configuración de encabezados HTTP personalizados para una aplicación de Amplify.	Sí	Sí
Amplify respeta los encabezados `Cache-Control` que se definan en un archivo `customHeaders.yaml` y tendrán prioridad sobre la configuración de caché predeterminada de Amplify.	Sí	Sí

Comportamiento del almacenamiento en caché	Compatibilidad anterior	Con mejoras en el almacenamiento en caché
Amplify respeta los encabezados `Cache-Control` establecidos en el marco de una aplicación en el caso de las rutas dinámicas (por ejemplo, las rutas SSR de Next.js). Si se establece un encabezado `Cache-Control` en el archivo `customHeaders.yaml` de la aplicación, esto tendrá prioridad sobre la configuración del archivo `next.config.js` .	Sí	Sí
Cada nueva implementación de una aplicación de CI/CD borra la memoria caché.	Sí	Sí
Puede activar el modo de rendimiento de una aplicación.	Sí	No La configuración del modo de rendimiento ya no está disponible en la consola de Amplify. Sin embargo, puede crear un encabezado `Cache-Control` que establezca la directiva `s-maxage`. Para obtener instrucciones, consulte Uso del encabezado Cache-Control para aumentar el rendimiento de la aplicación.

En la siguiente tabla se enumeran los cambios en los valores predeterminados de una configuración de caché específica.

Configuración de caché	Valor predeterminado anterior	Valor predeterminado con mejoras en el almacenamiento en caché
Duración de la caché de los activos estáticos	Dos segundos	Un año
Duración de la caché para las respuestas de proxy inverso	Dos segundos	Cero segundos (sin almacenamiento en caché)
Periodo de vida máximo (TTL)	Diez minutos	Un año

Para obtener más información sobre cómo Amplify determina la configuración de almacenamiento en caché que se debe aplicar a una aplicación, además de instrucciones sobre cómo administrar la configuración de las claves de caché, consulte los siguientes temas.

Temas

- Cómo Amplify aplica la configuración de caché a una aplicación
- Administración las cookies clave de caché

Cómo Amplify aplica la configuración de caché a una aplicación

Para administrar el almacenamiento en caché de su aplicación, Amplify determina el tipo de contenido que se distribuye al examinar el tipo de plataforma de la aplicación y las reglas de reescritura. En el caso de las aplicaciones `Compute`, Amplify también examina las reglas de enrutamiento del manifiesto de implementación.

> **Note**
>
> Amplify Hosting establece el tipo de plataforma de la aplicación durante la implementación. Una aplicación SSG (estática) se configura en el tipo de plataforma `WEB`. Una aplicación SSR (Next.js 12 o posterior) se establece en el tipo de plataforma `WEB_COMPUTE`.

Amplify identifica los siguientes cuatro tipos de contenido y aplica la política de caché administrada que se especifique.

Estático

El contenido distribuido desde las aplicaciones con la plataforma `WEB` o las rutas estáticas de una aplicación `WEB_COMPUTE`.

Este contenido utiliza la Amplify-StaticContent política de caché.

Optimización de imágenes

Las imágenes distribuidas por las rutas `ImageOptimization` de una aplicación `WEB_COMPUTE`.

Este contenido utiliza la Amplify-ImageOptimization política de caché.

Computación

El contenido distribuido por las rutas `Compute` de una aplicación `WEB_COMPUTE`. Esto incluye todo el contenido renderizado del lado del servidor (SSR).

Este contenido utiliza una de las siguientes opciones: Amplify-Default o Amplify-DefaultNoCookies política de caché en función del valor `cacheConfig.type` que esté establecido en su AmplifyApp.

Proxy inverso

El contenido distribuido por las rutas que coinciden con una regla personalizada de reescritura del proxy inverso. Para obtener más información sobre la creación de esta regla personalizada, consulte Reescritura de proxy inverso en el capítulo Using redirects.

Este contenido utiliza o bien el Amplify-Default o Amplify-DefaultNoCookies política de caché en función del valor `cacheConfig.type` que esté establecido en su AmplifyApp.

Descripción de las políticas de caché administradas de Amplify

Amplify utiliza las siguientes políticas de caché administradas predefinidas para optimizar la configuración de caché predeterminada de las aplicaciones de los clientes:alojadas.

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

- Amplify-StaticContent

Configuración de la política de caché administrada Amplify-Default

Vea esta política en la consola CloudFront

Esta política está diseñada para usarse con un origen que es una aplicación web de AWS Amplify.

Esta política tiene las siguientes opciones:

- TTL mínimo: 0 segundos
- Tiempo de vida máximo: 31 536 000 segundos (un año)
- TTL predeterminado: 0 segundos
- Encabezados incluidos en la clave de caché:
  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`
  - `Host`
- Cookies incluidas en la clave de caché: se incluyen todas las cookies.
- Cadenas de consulta incluidas en la clave de caché: se incluyen todas las cadenas de consulta.
- Configuración de objetos comprimidos en caché: habilitada para Gzip y Brotli.

Amplify: configuración de políticas de caché DefaultNoCookies gestionada

Vea esta política en la consola CloudFront

Esta política está diseñada para usarse con un origen que es una aplicación web de AWS Amplify.

Esta política tiene las siguientes opciones:

- TTL mínimo: 0 segundos
- Tiempo de vida máximo: 31 536 000 segundos (un año)
- TTL predeterminado: 0 segundos
- Encabezados incluidos en la clave de caché:
  - `Authorization`
  - `Accept`

  - `CloudFront-Viewer-Country`
  - `Host`
- Cookies incluidas en la clave de caché: no se incluyen las cookies.
- Cadenas de consulta incluidas en la clave de caché: se incluyen todas las cadenas de consulta.
- Configuración de objetos comprimidos en caché: habilitada para Gzip y Brotli.

Amplify: configuración de políticas de caché ImageOptimization gestionada

Vea esta política en la consola CloudFront

Esta política está diseñada para usarse con un origen que es una aplicación web de AWS Amplify.

Esta política tiene las siguientes opciones:

- TTL mínimo: 0 segundos
- Tiempo de vida máximo: 31 536 000 segundos (un año)
- TTL predeterminado: 0 segundos
- Encabezados incluidos en la clave de caché:
  - `Authorization`
  - `Accept`
  - `Host`
- Cookies incluidas en la clave de caché: no se incluyen las cookies.
- Cadenas de consulta incluidas en la clave de caché: se incluyen todas las cadenas de consulta.
- Configuración de objetos comprimidos en caché: habilitada para Gzip y Brotli.

Amplify: configuración de políticas de caché StaticContent gestionada

Vea esta política en la consola CloudFront

Esta política está diseñada para usarse con un origen que es una aplicación web de AWS Amplify.

Esta política tiene las siguientes opciones:

- TTL mínimo: 0 segundos
- Tiempo de vida máximo: 31 536 000 segundos (un año)
- TTL predeterminado: 0 segundos

- Encabezados incluidos en la clave de caché:
  - `Authorization`
  - `Host`
- Cookies incluidas en la clave de caché: no se incluyen las cookies.
- Cadenas de consulta incluidas en la clave de caché: no se incluyen las cadenas de consulta.
- Configuración de objetos comprimidos en caché: habilitada para Gzip y Brotli.

Administración las cookies clave de caché

Al implementar su aplicación en Amplify, puede elegir si desea incluir o excluir las cookies en la clave de caché. En la consola de Amplify, esta configuración se especifica en la página Encabezados y caché personalizados mediante el conmutador de configuración de la tecla de caché. Para obtener instrucciones, consulte Incluir o excluir cookies de la clave de caché.

Incluir cookies en la clave de caché

Esta es la configuración de caché predeterminada. Con esta configuración, Amplify elige automáticamente una configuración de caché óptima para su aplicación en función del tipo de contenido que distribuya.

Si utiliza la SDKs o AWS CLI, esta configuración corresponde a la configuración `cacheConfig.type` `AMPLIFY_MANAGED` con `CreateApp` o `UpdateApp` APIs.

Excluir las cookies de la clave de caché

Esta configuración de caché es similar a la configuración predeterminada, excepto que excluye todas las cookies de la clave de caché. Debe elegir explícitamente este tipo de configuración de caché.

Si decide excluir las cookies de la clave de caché, puede mejorar el rendimiento de la caché. Sin embargo, antes de elegir esta configuración de caché, es importante tener en cuenta si la aplicación utiliza cookies para ofrecer contenido dinámico.

Si está utilizando la SDKs o AWS CLI, esta configuración corresponde a ajustar la `cacheConfig.type` a `AMPLIFY_MANAGED_NO_COOKIES` con la `CreateApp` o `UpdateApp` APIs.

Para obtener más información sobre la clave de caché, consulte Comprender la clave de caché en la Guía para CloudFront desarrolladores de Amazon;.

Incluir o excluir cookies de la clave de caché

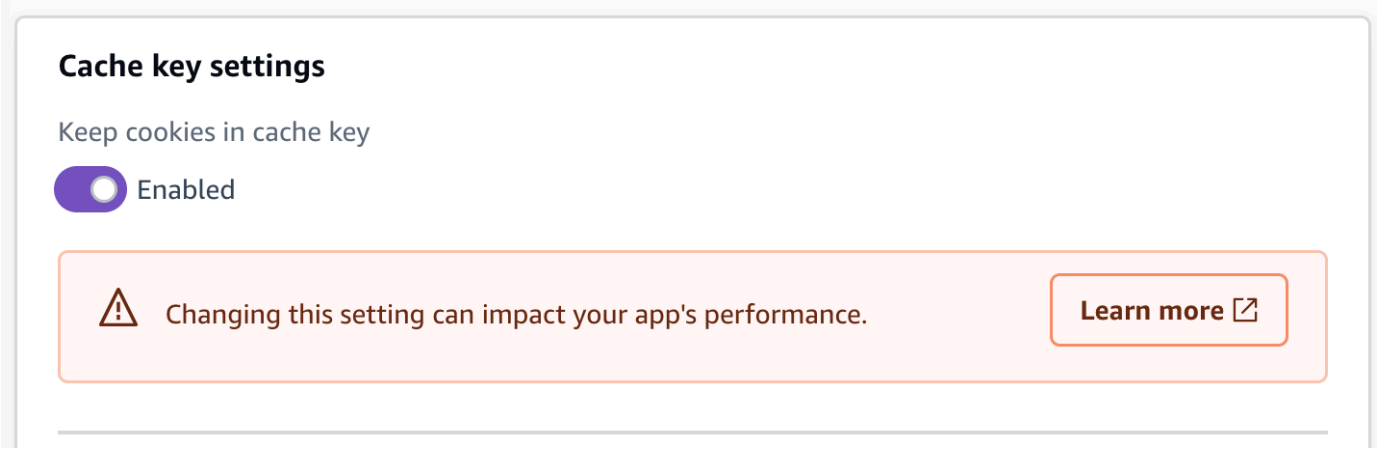
Puede establecer la configuración de cookies de clave de caché para una aplicación en la consola Amplify SDKs, o en. AWS CLI

Utilice el siguiente procedimiento para especificar si desea incluir o excluir las cookies de la clave de caché al implementar una nueva aplicación mediante la consola de Amplify.

Para establecer la configuración de cookies de la clave de caché al implementar una aplicación en Amplify

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, seleccione Crear nueva aplicación.
3. En la página Comenzar a crear con Amplify, seleccione el proveedor de repositorios de Git y, a continuación, elija Siguiente.
4. En la página Añadir ramificación de repositorio, haga lo siguiente:
   a. Seleccione el nombre del repositorio que desea conectar.
   b. Seleccione el nombre de la ramificación del repositorio que desea conectar.
   c. Elija Next (Siguiente).
5. Si la aplicación requiere un rol de servicio de IAM, puede permitir que el procesamiento de Amplify Hosting cree automáticamente uno en su lugar, o puede especificar uno que haya creado usted.
   - Para permitir que Amplify cree automáticamente un rol y lo asocie a su aplicación:
     - Elija Crear y utilizar un nuevo rol de servicio.
   - Para adjuntar un rol de servicio que haya creado anteriormente:
     a. Elija Utilizar un rol de servicio existente.
     b. Seleccione el rol que desea utilizar de la lista.
6. Seleccione Configuración avanzada y, a continuación, busque la sección Configuración de la clave de caché.

7. Seleccione Mantener las cookies en la clave de caché o Eliminar las cookies de la clave de caché. En la siguiente captura de pantalla se muestra el conmutador de configuración de la clave de caché de la consola.



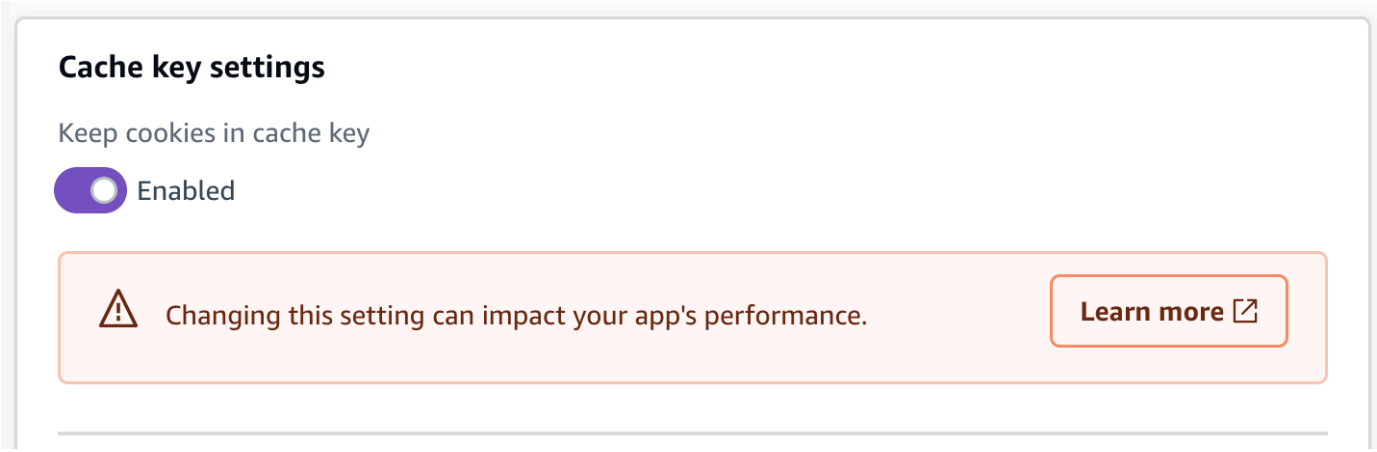
8. Elija Next (Siguiente).
9. En la página Revisar, elija Guardar e implementar.

Cambio de la configuración de cookies de la clave de caché de una aplicación

Puede cambiar la configuración de cookies de clave de caché de una aplicación que ya esté implementada en Amplify. Utilice el siguiente procedimiento para especificar si desea incluir o excluir las cookies de la clave de caché de una aplicación mediante la consola de Amplify.

Cambio de la configuración de cookies de la clave de caché de una aplicación implementada

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la página Todas las aplicaciones, seleccione la aplicación que desea actualizar.
3. En el panel de navegación, seleccione Alojamiento y, a continuación, seleccione Encabezados y caché personalizados.
4. En la página Encabezados y caché personalizados, busque la sección Configuración de la tecla de caché y seleccione Editar.
5. Seleccione Mantener las cookies en la clave de caché o Eliminar las cookies de la clave de caché. En la siguiente captura de pantalla se muestra el conmutador de configuración de la clave de caché de la consola.



6. Seleccione Guardar.

Administración del rendimiento de una aplicación de Amplify

La arquitectura de alojamiento predeterminada de Amplify optimiza el equilibrio entre el rendimiento de alojamiento y la disponibilidad de la implementación. A la mayoría de los clientes le recomendamos que utilicen la arquitectura predeterminada.

Si necesita un control más preciso del rendimiento de una aplicación, puede configurar manualmente el encabezado HTTP `Cache-Control` para optimizar el rendimiento de alojamiento y mantener el contenido en caché en la periferia de la red de entrega de contenido (CDN) durante un intervalo más largo.

Uso del encabezado Cache-Control para aumentar el rendimiento de la aplicación

Las directivas `max-age` y `s-maxage` de los encabezados `Cache-Control` de HTTP afectan a la duración del almacenamiento en caché del contenido de la aplicación. La directiva `max-age` le indica al navegador durante cuánto tiempo (en segundos) desea que el contenido permanezca en la memoria caché antes de que se actualice desde el servidor de origen. La directiva `s-maxage` anula la directiva `max-age` y le permite especificar durante cuánto tiempo (en segundos) desea que el contenido permanezca en la periferia de CDN antes de que se actualice desde el servidor de origen.

Las aplicaciones alojadas en Amplify respetan los encabezados `Cache-Control` que envía el origen, a menos que los anule al definir encabezados personalizados. Amplify solo aplica encabezados `Cache-Control` personalizados para las respuestas correctas con un código de estado `200 OK`. Esto evita que las respuestas de error se almacenen en caché y se distribuyen a otros usuarios que hagan la misma solicitud.

Puede ajustar manualmente la directiva `s-maxage` para tener más control sobre el rendimiento y la disponibilidad de implementación de la aplicación. Por ejemplo, para cambiar la duración en la que el contenido permanece almacenado en caché en la periferia, puede establecer manualmente el tiempo de vida (TTL) al actualizar `s-maxage` a un valor distinto al predeterminado de 31 536 000 segundos (1 año).

Puede definir encabezados personalizados para una aplicación en la sección Encabezados personalizados de la consola de Amplify. Para ver un ejemplo de YAML formato, consulteConfiguración de encabezados Cache-Control personalizados.

Utilice el siguiente procedimiento para configurar la directiva `s-maxage` para mantener el contenido en caché en la periferia de CDN durante 24 horas.

Para establecer una personalización Cache-Control header

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea configurar encabezados personalizados.
3. En el panel de navegación, elija Alojamiento y Encabezados personalizados.
4. En la página Encabezados personalizados, seleccione Editar.
5. En la ventana Editar encabezados personalizados, introduzca la información del encabezado personalizado de la siguiente manera:

   a. En `pattern`, introduzca **`**/*`** para todas las rutas.

   b. En `key`, introduzca **`Cache-Control`**.

   c. En `value`, introduzca **`s-maxage=86400`**.
6. Seleccione Guardar.
7. Vuelva a implementar la aplicación para aplicar el nuevo encabezado personalizado.

Soporte de firewall para sitios alojados

> **Note**
>
> La compatibilidad con firewalls para aplicaciones alojadas es una versión preliminar y está sujeta a cambios. Para obtener más información, consulte Limitaciones de vista previa del firewall.

El soporte de firewall está disponible hoy, en versión preliminar, en todos los sitios Regiones de AWS en los que Amplify Hosting opera, excepto en las regiones de suscripción. Esta integración se enmarca en un recurso AWS WAF global, similar a. CloudFront La web se ACLs puede conectar a varias aplicaciones de Amplify Hosting, pero deben residir en la misma región.

El soporte de firewall para sitios alojados le permite proteger sus aplicaciones web mediante una integración directa con AWS WAF. AWS WAF le permite configurar un conjunto de reglas, denominadas listas de control de acceso a la web (ACL web), que permiten, bloquean o supervisan (cuentan) las solicitudes web en función de las reglas y condiciones de seguridad web personalizables que usted defina. Cuando integras tu aplicación Amplify AWS WAF, obtienes más control y visibilidad del tráfico HTTP aceptado por tu aplicación. Para obtener más información AWS WAF, consulte Cómo AWS WAF funciona en la Guía para AWS WAF desarrolladores.

Puede utilizarla AWS WAF para proteger su aplicación Amplify de las vulnerabilidades web más comunes, como la inyección de SQL y las secuencias de comandos entre sitios. Estos podrían afectar a la disponibilidad y el rendimiento de la aplicación, comprometer la seguridad o consumir recursos excesivos. Por ejemplo, puedes crear reglas para permitir o bloquear las solicitudes de intervalos de direcciones IP específicos, las solicitudes de bloques de CIDR, las solicitudes que se originan en un país o región específicos o las solicitudes que contienen secuencias de comandos o códigos SQL inesperados.

También puede crear reglas que busquen una cadena o un patrón de expresión regular en encabezados HTTP, métodos, cadenas de consulta, URI y el cuerpo de la solicitud (limitado a los primeros 8 KB). Además, puede crear reglas para bloquear eventos de agentes de usuario, bots y rastreadores de contenido específicos. Por ejemplo, puede utilizar reglas basadas en la frecuencia para especificar el número de solicitudes web permitidas por IP de cliente en un periodo de 5 minutos actualizado constantemente.

Para obtener más información sobre los tipos de reglas compatibles y las AWS WAF funciones adicionales, consulta la Guía para AWS WAF desarrolladores y la Referencia de la AWS WAF API.

> **Important**
>
> La seguridad es una responsabilidad compartida entre usted AWS y usted. AWS WAF no es la solución a todos los problemas de seguridad de Internet y debe configurarla para cumplir sus objetivos de seguridad y cumplimiento. Para ayudarle a entender cómo aplicar el modelo de responsabilidad compartida cuando lo utilice AWS WAF, consulte Seguridad en el uso del AWS WAF servicio.

Habilitación AWS WAF de una aplicación Amplify

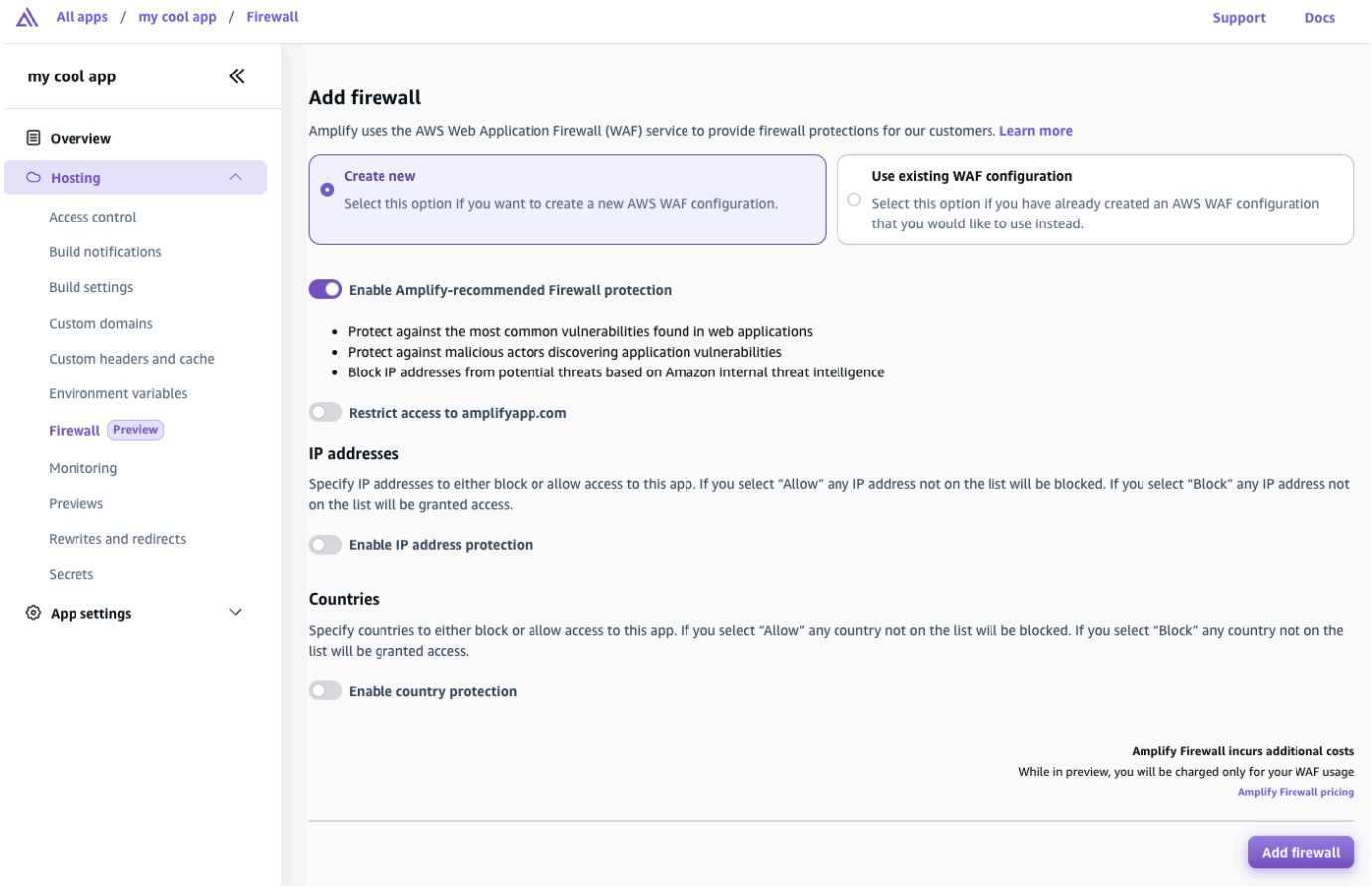
Puede activar las funciones del Firewall al crear una nueva aplicación o al editar la configuración de una aplicación Amplify existente. En ambos flujos de trabajo, asociará una ACL AWS WAF web a su aplicación Amplify Hosting.

Utilice el siguiente procedimiento AWS WAF para habilitar una aplicación existente en la consola de Amplify.

Habilitar AWS WAF para una aplicación Amplify existente

1. Inicie sesión en la consola Amplify AWS Management Console y ábrala en. https://console.aws.amazon.com/amplify/
2. En la página Todas las aplicaciones, elija el nombre de la aplicación implementada para activar la función de firewall.
3. En el panel de navegación, selecciona Alojamiento y, a continuación, Firewall.

   La siguiente captura de pantalla muestra cómo navegar a la página Añadir firewall en la consola Amplify.



4. En la página Agregar firewall, sus acciones dependerán de si desea crear una nueva AWS WAF configuración o usar una existente.

   - Cree una AWS WAF configuración nueva.

     a. Elija Crear nuevo.

     b. Si lo desea, habilite cualquiera de las siguientes configuraciones:

        i. Activa Activar la protección de firewall recomendada por Amplify.

        ii. Activa Restringir el acceso a amplifyapp.com para impedir el acceso a tu aplicación en el dominio Amplify predeterminado.

        iii. Para las direcciones IP, activa Activar la protección de direcciones IP.

           A. En Acción, selecciona Permitir si deseas especificar las direcciones IP a las que se accederá y se bloquearán todas las demás. Elija Bloquear si desea especificar las direcciones IP que se bloquearán y todas las demás tendrán acceso.

           B. Para la versión IP, seleccione una de las dos IPV4opciones IPV6.

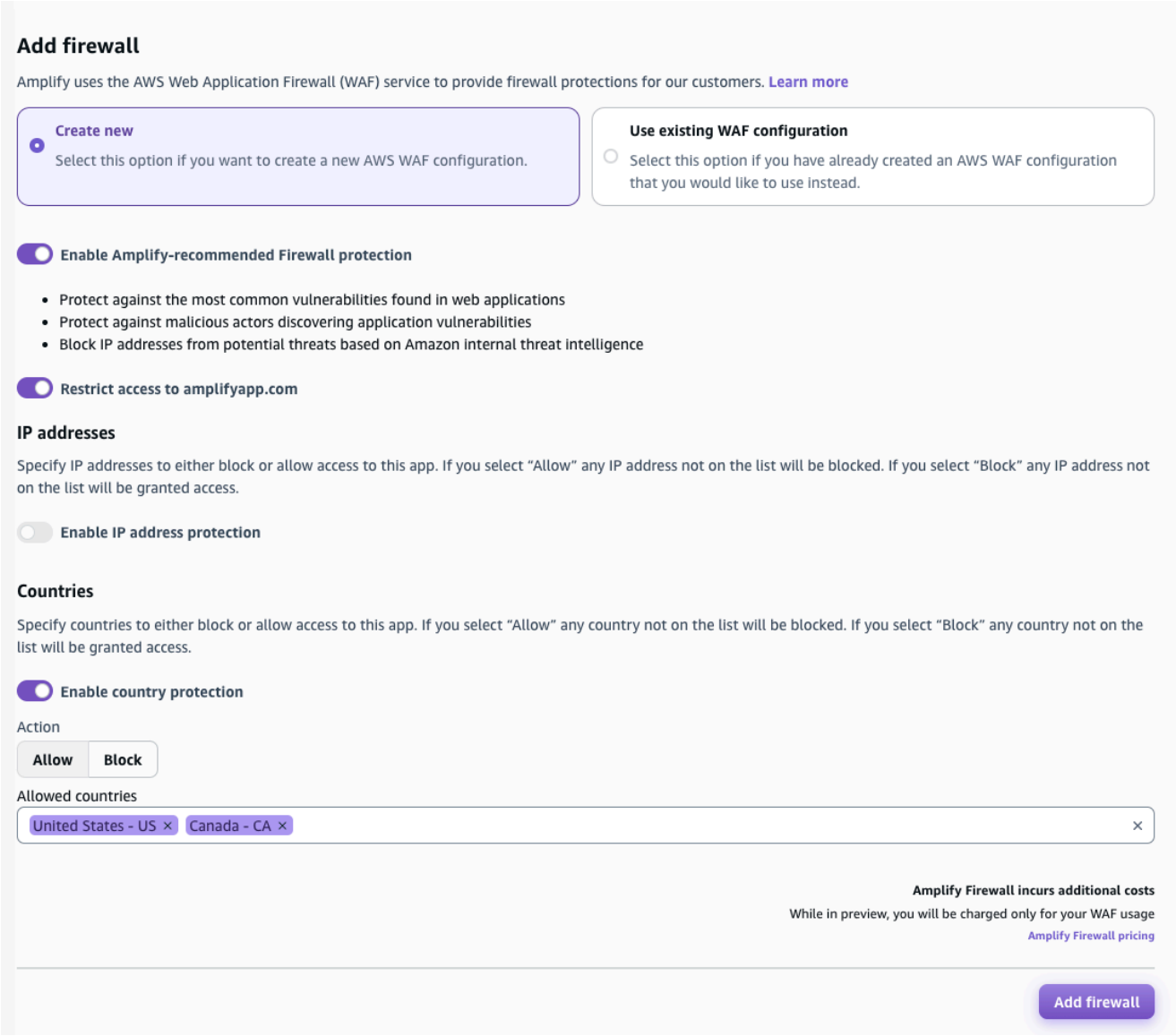
C. En el cuadro de texto de direcciones IP, introduzca las direcciones IP permitidas o bloqueadas, una por línea, en formato CIDR.

iv. En el caso de los países, activa Activar la protección por país.

A. En Acción, selecciona Permitir si quieres especificar los países a los que se les permitirá el acceso y se bloquearán todos los demás. Selecciona Bloquear si quieres especificar los países a los que se bloqueará y a los que tendrán acceso todos los demás.

B. En el caso de los países, selecciona los países permitidos o bloqueados de la lista.

La siguiente captura de pantalla muestra cómo habilitar una nueva AWS WAF configuración para una aplicación.



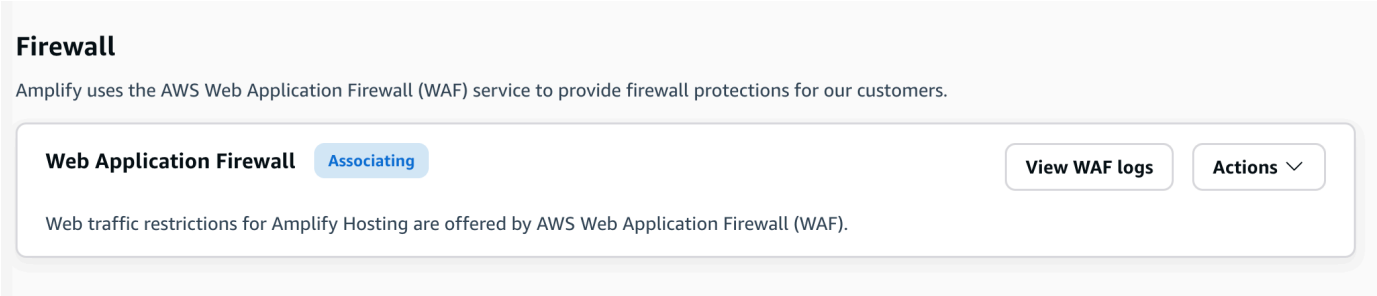
- Utilice una AWS WAF configuración existente.

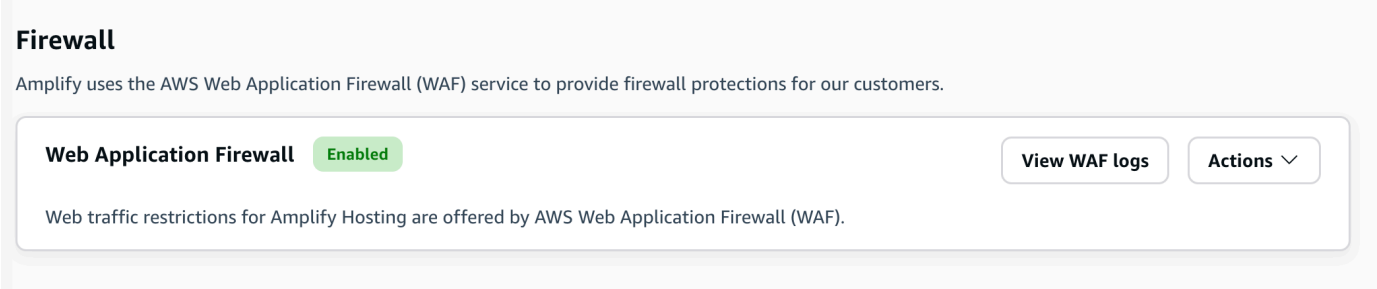
  a. Elija Usar la AWS WAF configuración existente.

  b. Seleccione una configuración guardada de la lista de páginas web ACLs AWS WAF de su Cuenta de AWS.

5. Selecciona Añadir firewall.
6. En la página del firewall, se muestra el estado de asociación para indicar que la AWS WAF configuración se está propagando. Cuando se completa el proceso, el estado cambia a Activado.

   Las siguientes capturas de pantalla muestran el estado del progreso del firewall en la consola Amplify e indican cuándo AWS WAF la configuración está asociada y habilitada.





Desasociar una ACL web de una aplicación Amplify

No puede eliminar una ACL web que esté asociada a una aplicación Amplify. Primero debe desasociar la ACL web de la aplicación en la consola de Amplify. A continuación, puede eliminarla en la AWS WAF consola.

Para desasociar una ACL web de una aplicación Amplify

1. Inicie sesión en la consola Amplify AWS Management Console y ábrala en. https://console.aws.amazon.com/amplify/
2. En la página Todas las aplicaciones, elija el nombre de la aplicación de la que desee desasociar una ACL web.
3. En el panel de navegación, elija Alojamiento y, a continuación, Firewall.
4. En la página del firewall, elija Acciones y, a continuación, elija Desasociar el firewall.
5. En el modo de confirmación, introduce y, a continuación**`disassociate`**, selecciona Desasociar el firewall.
6. En la página del firewall, se muestra el estado de desasociación para indicar que la AWS WAF configuración se está propagando.

   Cuando se complete el proceso, puede eliminar la ACL web de la consola. AWS WAF

Cómo se integra Amplify con AWS WAF

La siguiente lista proporciona detalles específicos sobre cómo se integra la compatibilidad con el firewall AWS WAF y las limitaciones que se deben tener en cuenta al crear sitios web ACLs y asociarlos con las aplicaciones de Amplify.

- Puedes habilitarlo AWS WAF para cualquier tipo de aplicación Amplify. Esto incluye cualquier marco compatible, aplicaciones renderizadas del lado del servidor (SSR) y sitios totalmente estáticos. AWS WAF es compatible con las aplicaciones Amplify Gen 1 y Gen 2.
- Debe crear un sitio web ACLs que desee asociar a una aplicación Amplify en la región Global (CloudFront). ACLs Puede que tu web regional ya exista Cuenta de AWS, pero no es compatible con Amplify.
- La ACL web y la aplicación Amplify deben crearse en la misma. Cuenta de AWS Se puede utilizar AWS Firewall Manager para replicar AWS WAF las reglas en todas partes Cuentas de AWS, a fin de simplificar el mantenimiento de las reglas de la organización centralizadas y distribuidas entre múltiples Cuentas de AWS. Para obtener más información, consulte AWS Firewall Manager en la Guía para desarrolladores de AWS WAF .
- Puede compartir la misma ACL web en varias aplicaciones de Amplify de la misma manera. Cuenta de AWS Todas las aplicaciones deben estar en la misma región.
- Cuando asocias una ACL web a una aplicación Amplify, la ACL web se conecta a todas las sucursales de la aplicación de forma predeterminada. Cuando cree nuevas sucursales, tendrán la ACL web.
- Cuando asocias una ACL web a una aplicación Amplify, se asocia automáticamente a todos los dominios de la aplicación. Sin embargo, puede configurar las reglas que se apliquen a un único nombre de dominio mediante las reglas de coincidencia de encabezados de host.
- No puede eliminar una ACL web que esté asociada a una aplicación Amplify. Antes de eliminar una ACL web en la AWS WAF consola, debe desasociarla de la aplicación.

Amplify la política de recursos de ACL web

Para permitir que Amplify acceda a su ACL web, se adjunta una política de recursos a la ACL web durante la asociación. Amplify crea esta política de recursos automáticamente, pero puedes verla mediante la API. AWS WAFV2 GetPermissionPolicy Se requieren los siguientes permisos de IAM para asociar una ACL web a una aplicación Amplify.

- amplificar: ACL AssociateWeb

- wafv2: ACL AssociateWeb
- wafv2: PutPermissionPolicy
- wafv2: GetPermissionPolicy

Limitaciones de vista previa del firewall

La versión preliminar de Firewall tiene las siguientes limitaciones.

1. Durante el período de vista previa, Amplify admite la integración parcial con. CloudTrail Algunos eventos de administración durante la asociación de ACL web no aparecerán en los CloudTrail registros.
2. Durante la vista previa, cuando su ACL web esté asociada a un recurso de Amplify, este nuevo recurso de Amplify no se mostrará en AWS los recursos asociados de la consola. AWS WAF Puede usar la GetAppAPI Amplify para mostrar la ACL web asociada a una aplicación. Puede asociar y desasociar el recurso Amplify del Firewall navegando a la página Firewall de una aplicación en la consola de Amplify Hosting.
3. Durante la versión preliminar, la AWS Config integración no estará disponible.
4. La función Firewall no está disponible en las regiones en las que Amplify existe actualmente: Asia-Pacífico (Hong Kong) (ap-east-1), Europa (Milán) (eu-south-1) y Oriente Medio (Baréin) (me-south-1).

Precios de los firewalls

Durante la versión preliminar, solo incurrirá en cargos del servicio basados en la utilización. AWS WAF AWS WAF cobra 5$ al mes por ACL web y 1 dólar por regla, entre otros cargos. Como mínimo, pagará 7$ por esta integración, siempre que tenga una ACL web con dos reglas. Para obtener más información sobre precios, consulte precios de AWS WAF.

Además de los cargos por AWS WAF uso, esta integración incurrirá en una tarifa fija por aplicación desde el servicio Amplify una vez que entre en disponibilidad general (GA). Los detalles específicos de los precios se comunicarán en GA.

Seguridad en Amplify

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS Amplify, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le permite comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amplify. En los siguientes temas, se mostrará cómo configurar Amplify para satisfacer sus objetivos de seguridad y conformidad. También aprenderás a usar otros AWS servicios que te ayudan a monitorear y proteger tus recursos de Amplify.

Temas

Administración de identidades y accesos para Amplify

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amplify. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realices en Amplify.

Usuario de servicio: si utiliza el servicio de Amplify para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amplify para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de Amplify, consulte Solución de problemas de identidad y acceso de Amplify.

Administrador de servicio: si está a cargo de los recursos de Amplify de la empresa, probablemente tenga acceso completo a Amplify. El trabajo consiste en determinar a qué características y recursos de Amplify deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su gestionador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información acerca de cómo la empresa puede utilizar IAM con Amplify, consulte Cómo funciona Amplify con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que desee obtener información acerca de cómo escribir políticas para administrar el acceso a Amplify. Para consultar ejemplos

de políticas basadas en la identidad de Amplify que puede utilizar en IAM, consulte Ejemplos de políticas basadas en identidades para Amplify.

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su gestionador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, asumes un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Autenticación multifactor AWS en IAM en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren

que inicie sesión como usuario raíz, consulta Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulta ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulta Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales

temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puedes crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué puedes acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puedes asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puedes utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.

- Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en AWS ellas, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulta Reenviar sesiones de acceso.
- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puedes asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre

la estructura y el contenido de los documentos de política JSON, consulta Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puedes crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utiliza para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puedes asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades puedes clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios puedes utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puedes realizar una entidad

principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas puedes establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puedes conceder a una entidad de IAM (usuario o rol de IAM). Puedes establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulta Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): SCPs son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.

- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también puedes proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulta Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amplify con IAM

Antes de utilizar IAM para administrar el acceso a Amplify, conozca qué características de IAM se pueden utilizar con Amplify.

Características de IAM que puede utilizar con Amplify

Característica de IAM	Compatibilidad de Amplify
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí

Característica de IAM	Compatibilidad de Amplify
Recursos de políticas	Sí
Claves de condición de política	Sí
ACLs	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Sesiones de acceso directo (FAS)	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una visión general de cómo Amplify y otros AWS servicios funcionan con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas de Amplify basadas en identidades

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está asociada. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades para Amplify

Para ver ejemplos de políticas basadas en identidad de Amplify, consulte Ejemplos de políticas basadas en identidades para Amplify.

Políticas basadas en recursos de Amplify

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios puedes utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puedes realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los directores pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Acciones de política para Amplify

Compatibilidad con las acciones de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puedes utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para una lista de las acciones de Amplify, consulte Acciones definidas por AWS Amplify en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Amplify utilizan el siguiente prefijo antes de la acción:

```
amplify
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de políticas basadas en identidad de Amplify, consulte Ejemplos de políticas basadas en identidades para Amplify.

Recursos de políticas para Amplify

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para obtener una lista de los tipos de recursos de Amplify y sus tipos ARNs, consulte los tipos de recursos definidos AWS Amplify en la Referencia de autorización de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS Amplify.

Para ver ejemplos de políticas basadas en identidad de Amplify, consulte Ejemplos de políticas basadas en identidades para Amplify.

Claves de condición de política para Amplify

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para una lista de las claves de condición de Amplify, consulte Claves de condición para AWS Amplify en la Referencia de autorizaciones de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por AWS Amplify.

Para ver ejemplos de políticas basadas en identidad de Amplify, consulte Ejemplos de políticas basadas en identidades para Amplify.

Listas de control de acceso (ACLs) en Amplify

Soportes ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Control de acceso basado en atributos (ABAC) con Amplify

Compatibilidad con ABAC (etiquetas en las políticas): parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulta Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con Amplify

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de roles, consulte Cambio de un usuario a un rol de IAM (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Sesiones de acceso directo para Amplify

Admite sesiones de acceso directo (FAS): sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulta Reenviar sesiones de acceso.

Roles de servicio para Amplify

Compatibilidad con roles de servicio: sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de Amplify. Edite los roles de servicio solo cuando Amplify proporcione orientación para hacerlo.

Roles vinculados a servicios de Amplify

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puedes asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para obtener más información acerca de cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM en la Guía de usuario de IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Elija el vínculo Sí para ver la documentación acerca de los roles vinculados al servicio en cuestión.

Ejemplos de políticas basadas en identidades para Amplify

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amplify. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. Un administrador de IAM puedes crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por Amplify, incluido el formato de cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de la Referencia AWS Amplify de autorización de servicio. ARNs

Temas

- Prácticas recomendadas sobre las políticas

- Usar la consola de Amplify
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de Amplify de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se puedes llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utiliza condiciones en las políticas de IAM para restringir aún más el acceso: puedes agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puedes escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Usar la consola de Amplify

Para acceder a la AWS Amplify consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de Amplify en su. Cuenta de AWS Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Con el lanzamiento de Amplify Studio, es necesario contar con permisos de `amplify` y `amplifybackend` para eliminar una aplicación o un backend. Si la política de IAM solo proporciona permisos de `amplify`, el usuario recibirá un error de permisos al intentar eliminar una aplicación. Los administradores que redacten políticas, determinan qué permisos han de conceder a los usuarios que deban llevar a cabo acciones de eliminación.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de Amplify, adjunte también la política Amplify `ConsoleAccess` o `ReadOnly` AWS gestionada a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. 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 políticas administradas para AWS Amplify

Una política AWS gestionada es una política independiente creada y administrada por. AWS AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen

todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Política gestionada por AWS: AdministratorAccess -Amplify

Puede adjuntar la política `AdministratorAccess-Amplify` a las identidades de IAM. Amplify también asocia esta política a un rol de servicio que permite que Amplify realice acciones en su nombre.

Al implementar un backend en la consola de Amplify, debe crear `Amplify-Backend Deployment` un rol de servicio que Amplify utilice para crear y administrar los recursos. AWS IAM asocia la política administrada por `AdministratorAccess-Amplify` al rol de servicio de `Amplify-Backend Deployment`.

Esta política otorga permisos administrativos a las cuentas y, al mismo tiempo, permite explícitamente el acceso directo a los recursos que necesitan las aplicaciones de Amplify para crear y administrar los backends.

Detalles de los permisos

Esta política proporciona acceso a varios AWS servicios, incluidas las acciones de IAM. Estas acciones permiten que las identidades con esta política se utilicen AWS Identity and Access Management para crear otras identidades con cualquier permiso. Así se facilita la escalabilidad de permisos. Esta política debe considerarse tan eficaz como la política de `AdministratorAccess`.

Esta política concede los permisos de acción de `iam:PassRole` a todos los recursos. Se necesita para admitir la configuración de grupos de usuarios de Amazon Cognito.

Para ver los permisos de esta política, consulte AdministratorAccess-Amplify en la Referencia de políticas AWS gestionadas.

AWS política gestionada: AmplifyBackendDeployFullAccess

Puede adjuntar la política `AmplifyBackendDeployFullAccess` a las identidades de IAM.

Esta política otorga a Amplify permisos de acceso total para implementar los recursos de backend de Amplify utilizando. AWS Cloud Development Kit (AWS CDK) Los permisos se transfieren a las AWS CDK funciones que tienen los permisos de política necesarios. `AdministratorAccess`

Detalles de los permisos

Esta política incluye permisos para hacer lo siguiente.

- `Amplify`: recuperar metadatos sobre las aplicaciones implementadas.
- `AWS CloudFormation`: crear, actualizar y eliminar pilas administradas por Amplify.
- `SSM`: crear, actualizar y eliminar la `String` del almacén de parámetros SSM administrado por Amplify y los parámetros `SecureString`.
- `AWS AppSync`— Actualice y recupere los recursos AWS AppSync del esquema, la resolución y la función. El objetivo es respaldar la funcionalidad de intercambio en caliente del entorno de pruebas de Gen 2.
- `Lambda`: actualizar y recuperar la configuración de las funciones administradas por Amplify. El objetivo es respaldar la funcionalidad de intercambio en caliente del entorno de pruebas de Gen 2.

  Recupere las etiquetas de una función de Lambda. El objetivo es respaldar las funciones de Lambda definidas por los clientes.
- `Amazon S3`: recupere los activos de implementación de Amplify.
- `AWS Security Token Service`— Permite que la AWS Cloud Development Kit (AWS CDK) CLI asuma la función de despliegue.
- `Amazon RDS`: lea los metadatos de instancias de base de datos, clústeres y proxies.
- `Amazon EC2`: lea la información de la zona de disponibilidad de una subred.
- `CloudWatch Logs` recupere los registros de la función de Lambda de un cliente. El objetivo es permitir que un entorno de pruebas limitado de desarrollo en la nube de Amplify transmita los registros de una función de Lambda al terminal de un cliente.

Para ver los permisos de esta política, consulte AmplifyBackendDeployFullAccess en la Referencia de la política administrada de AWS .

Amplify las actualizaciones de las políticas gestionadas AWS

Consulta los detalles sobre las actualizaciones de las políticas AWS gestionadas de Amplify desde que este servicio comenzó a rastrear estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de Historial de documentos para AWS Amplify.

Cambio	Descripción	Fecha
AmplifyBackendDeployFullAccess: actualización de una política actual	Agregue acceso de lectura al `logs:FilterLogEvents` recurso para permitir que Amplify transmita registros desde funciones en las que se creó un grupo de registros personalizado. Se trata de una extensión de la capacidad actual de transmitir los registros de una función Lambda.	14 de noviembre de 2024
AmplifyBackendDeployFullAccess: actualización de una política actual	Agregue acceso de lectura a los recursos de `lambda:ListTags` y `logs:FilterLogEvents` para admitir las funciones de Lambda definidas por los clientes. Esto permite que un entorno de pruebas limitado de desarrollo en la nube de Amplify transmita los registros de una función de Lambda al terminal de un cliente.	18 de julio de 2024

Cambio	Descripción	Fecha
AmplifyBackendDeployFullAccess: actualización de una política actual	Agregue acceso de lectura al recurso de `arn:aws:ssm:*:*:parameter/cdk-bootstrap/*` para permitir que Amplify detecte la versión de arranque del CDK en la cuenta de un cliente.	31 de mayo de 2024

Cambio	Descripción	Fecha
AmplifyBackendDeployFullAccess: actualización de una política actual	Añada una nueva declaración `AmplifyDiscoverRDSVpcConfig` de política con permisos de Amazon RDS y Amazon de EC2 solo lectura sujetos a las condiciones de los recursos y de la cuenta. Estos permisos admiten el comando `npx amplify generate schema-from-database` de Amplify Gen 2, que permite a los clientes generar un esquema de datos de TypeScript a partir de una base de datos SQL existente. Agregue los permisos `rds:DescribeDBProxies`, `rds:DescribeDBInstances`, `rds:DescribeDBClusters`, `rds:DescribeDBSubnetGroups` y `ec2:DescribeSubnets`. El `npx amplify generate schema-from-database` comando requiere estos permisos para comprobar si un host de base de datos específico está alojado en Amazon RDS y generar automáticamente la configuración de Amazon VPC necesaria para aprovisionar los demás recursos necesario	17 de abril de 2024

Cambio	Descripción	Fecha
	s para configurar una AWS AppSync API respaldada por una base de datos SQL.	
AmplifyBackendDeployFullAccess: actualización de una política actual	Agregue la acción de política `cloudformation:DeleteStack` para permitir la eliminación de la pila cuando se llame a la API `DeleteBranch` . Agregue la acción de política `lambda:GetFunction` para admitir las funciones de intercambio en caliente. Agregue la acción de política `lambda:UpdateFunctionConfiguration` para admitir actualizaciones de la función de Lambda.	5 de abril de 2024
AdministratorAccess-Amplify: actualización de una política existente	Agregue los `cloudformation:UnTagResource` permisos `cloudformation:TagResource` y para admitir llamadas a. AWS CloudFormation APIs	4 de abril de 2024

Cambio	Descripción	Fecha
AmplifyBackendDeployFullAccess: actualización de una política actual	Añada la acción `lambda:InvokeFunction` política para respaldar el AWS Cloud Development Kit (AWS CDK) intercambio en caliente. AWS CDK Realiza llamadas directas a una función de Lambda para realizar el intercambio en caliente de activos de Amazon S3. Agregue la acción de política `lambda:UpdateFunctionCode` para admitir las funciones de intercambio en caliente.	2 de enero de 2024
AmplifyBackendDeployFullAccess: actualización de una política actual	Agregue acciones de políticas para admitir la operación `UpdateApiKey` . Esto es necesario para permitir una implementación correcta de la aplicación después de salir del entorno aislado y reiniciar lo sin eliminar los recursos.	17 de noviembre de 2023
AmplifyBackendDeployFullAccess: actualización de una política actual	Agregue el permiso `amplify:GetBackendEnvironment` para admitir la implementación de la aplicación de Amplify.	6 de noviembre de 2023

Cambio	Descripción	Fecha
AmplifyBackendDeployFullAccess: política nueva	Amplify ha agregado una nueva política con los permisos mínimos necesarios para implementar los recursos de backend de Amplify.	8 de octubre de 2023
AdministratorAccess-Amplificar: actualización de una política existente	Agregue el permiso `ecr:DescribeRepositories` , necesario para la interfaz de la línea de comandos (CLI) de Amplify.	1 de junio de 2023

Cambio	Descripción	Fecha
AdministratorAccess-Amplificar: actualización de una política existente	Agregue una acción de política para permitir la eliminación de etiquetas de un recurso de AWS AppSync . Agregue una acción de política para admitir el recurso Amazon Polly. Agregue una acción política para respaldar la actualización de la configuración del OpenSearch dominio. Agregue una acción de política para permitir la eliminación de etiquetas de un rol de AWS Identity and Access Management . Agregue una acción de política para permitir la eliminación de etiquetas de un recurso de Amazon DynamoDB. Agregue los permisos `cloudfront:GetCloudFrontOriginAccessIdentity` y `cloudfront:GetCloudFrontOriginAccessIdentityConfig` al bloque de instrucción `CLISDKCalls` para permitir los flujos de trabajo de	24 de febrero de 2023

Cambio	Descripción	Fecha
	publicación, así como los de alojamiento de Amplify. Agregue el permiso `s3:PutBucketPublicAccessBlock` al bloque de instrucción `CLIManageviaCFNPolicy` para que la AWS CLI permita la práctica recomendada de seguridad de Amazon S3 de habilitar la característica de bloqueo de acceso público de Amazon S3 en buckets internos. Añada el `cloudformation:DescribeStacks` permiso al bloque de `CLISDKCalls` sentencias para permitir la recuperación de las AWS CloudFormation pilas de los clientes al volver a intentarlo en el procesador de fondo Amplify para evitar la duplicación de las ejecuciones si una pila se está actualizando. Añada el permiso `cloudformation:ListStacks` al bloque de instrucción `CLICloudformationPolicy` . Este permiso es necesario para respaldar plenamente la acción.	

Cambio	Descripción	Fecha
	CloudFormation DescribeStacks	
AdministratorAccess-Amplificar: actualización de una política existente	Agregue acciones políticas para permitir que la función de renderización del lado del servidor Amplify incorpore las métricas de las aplicaciones a CloudWatch las del cliente. Cuenta de AWS	30 de agosto de 2022
AdministratorAccess-Amplificar: actualización de una política existente	Agregue acciones de política para bloquear el acceso público al bucket de Amazon S3 de implementación de Amplify.	27 de abril de 2022
AdministratorAccess-Amplificar: actualización de una política existente	Agregue una acción para permitir a los clientes eliminar sus aplicaciones representadas en el lado del servidor (SSR). Esto también permite que la CloudFront distribución correspondiente se elimine correctamente. Agregue una acción para permitir a los clientes especificar una función de Lambda diferente para gestionar los eventos de una fuente de eventos existente mediante la CLI de Amplify. Con estos cambios, AWS Lambda podrá realizar la UpdateEventSourceMappingacción.	17 de abril de 2022

Cambio	Descripción	Fecha
AdministratorAccess-Amplificar: actualización de una política existente	Agregue una acción de política para habilitar las acciones del creador de Amplify UI en todos los recursos.	2 de diciembre de 2021
AdministratorAccess-Amplificar: actualización de una política existente	Agregue acciones de política para admitir la caracterí stica de autenticación de Amazon Cognito que emplea proveedores de identidad social. Agregue una acción de política para admitir las capas de Lambda. Agregue una acción de política para admitir la categoría de Amplify Storage.	8 de noviembre de 2021

Cambio	Descripción	Fecha
AdministratorAccess-Amplificar: actualización de una política existente	Agregue acciones de Amazon Lex para admitir la categoría de Amplify Interactions. Agregue acciones de Amazon Rekognition para admitir la categoría de Amplify Predictions. Agregue una acción de Amazon Cognito para admitir la configuración de MFA en los grupos de usuarios de Amazon Cognito. Añada CloudFormation acciones de apoyo. AWS CloudFormation StackSets Agregue acciones de Amazon Location Service para admitir la categoría de Amplify Geo. Agregue una acción de Lambda para admitir las capas de Lambda en Amplify. Agrega acciones CloudWatch de registro para respaldar CloudWatch los eventos. Agregue acciones de Amazon S3 para admitir la categoría de Amplify Storage. Agregue acciones de política para admitir las aplicaciones	27 de septiembre de 2021

Cambio	Descripción	Fecha
	representadas en el lado del servidor (SSR).	

Cambio	Descripción	Fecha
AdministratorAccess-Amplificar: actualización de una política existente	Consolide todas las acciones de Amplify en una única acción de `amplify:*` . Agregue una acción de Amazon S3 para admitir el cifrado de los buckets de Amazon S3 de los clientes. Agregue acciones de límite de permisos de IAM para admitir las aplicaciones de Amplify que tienen habilitados los límites de permisos. Agregue acciones de Amazon SNS para admitir la visualización de los números de teléfono de origen y la visualización, creación, verificación, así como la eliminación de los números de teléfono de destino. Amplify Studio: añada acciones políticas AWS Lambda, de IAM y de Amazon Cognito para permitir la administración de los backends en la consola Amplify AWS CloudFormation y Amplify Studio. Añada una declaración de política AWS Systems Manager (SSM) para	28 de julio de 2021

Cambio	Descripción	Fecha
	gestionar los secretos del entorno de Amplify. Agregue una AWS CloudFormation `ListResources` acción para admitir las capas Lambda para las aplicaciones de Amplify.	
Amplify comenzó el seguimiento de los cambios	Amplify comenzó a realizar un seguimiento de los cambios en sus políticas AWS gestionadas.	28 de julio de 2021

Solución de problemas de identidad y acceso de Amplify

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con Amplify e IAM.

Temas

- No tengo autorización para realizar una acción en Amplify
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amplify

No tengo autorización para realizar una acción en Amplify

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `amplify:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplify:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `amplify:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Con el lanzamiento de Amplify Studio, es necesario contar con permisos de `amplify` y `amplifybackend` para eliminar una aplicación o un backend. Si un administrador ha redactado una política de IAM que proporciona únicamente permisos de `amplify`, aparecerá un error de permisos al intentar eliminar una aplicación.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio de *`example-amplify-app`*, pero no tiene los permisos ficticios `amplifybackend:`*`RemoveAllBackends`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplifybackend;:RemoveAllBackends on resource: example-amplify-app
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *`example-amplify-app`* mediante la acción `amplifybackend:`*`RemoveAllBackends`*.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, sus políticas deben actualizarse para permitirle pasar un rol a Amplify.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amplify. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amplify

Puedes crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puedes especificar una persona de confianza para que asuma el rol. En el caso de los servicios que respaldan las políticas basadas en recursos o las listas de control de acceso (ACLs), puede usar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si Amplify admite estas características, consulte Cómo funciona Amplify con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad en la Cuenta de AWS Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Protección de datos en Amplify

AWS Amplify se ajusta al modelo de responsabilidad AWS compartida, modelo de , que incluye normas y directrices para la protección de datos. AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los AWS servicios. AWS mantiene el control de los datos alojados en esta infraestructura, incluidos los controles de configuración de seguridad para gestionar el contenido y los datos personales de los clientes. AWS los clientes y los socios de APN, que actúan

como controladores o procesadores de datos, son responsables de cualquier dato personal que coloquen en la AWS nube.

Para proteger los datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, cada usuario recibe únicamente los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de las siguientes maneras:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados de AWS los servicios.
- Utilice avanzados servicios de seguridad administrados, como Amazon Macie, que lo ayuden a detectar y proteger los datos personales almacenados en Amazon S3.

Le recomendamos encarecidamente que nunca introduzca información de identificación confidencial, como, por ejemplo, números de cuenta de sus clientes, en los campos de formato libre, como el campo Nombre. Esto incluye cuando trabaja con Amplify u otros AWS servicios mediante la consola, la API o. AWS CLI AWS SDKs Es posible que cualquier dato que ingrese en Amplify u otros servicios se incluya en los registros de diagnóstico. Cuando proporcione una URL a un servidor externo, no incluya información de credenciales en la URL para validar la solicitud para ese servidor.

Para obtener más información sobre la protección de datos, consulte la entrada de blog relativa al modelo de responsabilidad compartida de AWS y GDPR en el blog de seguridad de AWS .

Cifrado en reposo

El cifrado en reposo hace referencia a la protección de sus datos del acceso no autorizado mediante el cifrado de datos mientras están almacenados. Amplify cifra los artefactos de creación de una aplicación de forma predeterminada mediante Amazon AWS KMS keys S3, que son administrados por. AWS Key Management Service

Amplify usa Amazon CloudFront para ofrecer tu aplicación a tus clientes. CloudFront utiliza volúmenes SSDs cifrados para los puntos de presencia de ubicación perimetral (POPs) y volúmenes EBS cifrados para las cachés perimetrales regionales (). RECs El código de función y la configuración de CloudFront Functions siempre se almacenan en un formato cifrado en la

ubicación POPs cifrada SSDs de la periferia y en otras ubicaciones de almacenamiento utilizadas por. CloudFront

Cifrado en tránsito

El cifrado en tránsito se refiere a proteger sus datos de ser interceptados mientras se mueven entre los extremos de comunicación. De forma predeterminada, Amplify Hosting proporciona cifrado de datos en tránsito. Todas las comunicaciones entre los clientes y Amplify, así como entre Amplify y sus dependencias posteriores están protegidas con conexiones TLS que se firman mediante el proceso de firma de Signature Version 4. Todos los puntos finales de Amplify Hosting utilizan certificados SHA-256 administrados por una autoridad de certificación privada. AWS Certificate Manager Para más información, consulte Proceso de firma de Signature Version 4 y ¿Qué es PCA de ACM?.

Administración de claves de cifrado

AWS Key Management Service (KMS) es un servicio gestionado para crear y controlar AWS KMS keys las claves de cifrado utilizadas para cifrar los datos de los clientes. AWS Amplify genera y administra claves criptográficas para cifrar datos en nombre de los clientes. No hay claves de cifrado para que las administre.

Validación de conformidad para AWS Amplify

Los auditores externos evalúan la seguridad y el cumplimiento AWS Amplify como parte de varios programas de AWS cumplimiento. Estos incluyen SOC, PCI, ISO, HIPAA, MTCS, C5, K-ISMS, ENS High, OSPAR, HITRUST, CSF y FINMA.

Para saber si un programa de cumplimiento Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance by Compliance Servicios de AWS y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la HIPAA.
- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulta la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Seguridad de la infraestructura en AWS Amplify

Como servicio gestionado, AWS Amplify está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a Amplify a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Registro y supervisión de eventos de seguridad en Amplify

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de Amplify y sus demás AWS soluciones. AWS proporciona las siguientes herramientas de monitoreo para ver Amplify, informar cuando algo anda mal y tomar medidas automáticas cuando sea apropiado:

- Amazon CloudWatch supervisa en tiempo real sus AWS recursos y las aplicaciones en las que se ejecuta AWS. Puede recopilar métricas y realizar un seguimiento de ellas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando alguna métrica alcanza el umbral que se especifique. Por ejemplo, puede CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de sus instancias de Amazon Elastic Compute Cloud (Amazon EC2) y lanzar nuevas instancias automáticamente cuando sea necesario. Para obtener más información sobre el uso de CloudWatch métricas y alarmas con Amplify, consulte. Supervisión de una aplicación de Amplify
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde EC2 instancias de Amazon y otras fuentes. AWS CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.
- AWS CloudTrailcaptura las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre y envía los archivos de registro a un depósito de Amazon Simple Storage

Service (Amazon S3) que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron. Para obtener más información, consulte Registro de llamadas a la API de Amplify mediante AWS CloudTrail.

- Amazon EventBridge es un servicio de bus de eventos sin servidor que facilita la conexión de sus aplicaciones con datos de diversas fuentes. EventBridge ofrece un flujo de datos en tiempo real desde sus propias aplicaciones, aplicaciones Software-as-a-Service (SaaS) y AWS servicios, y dirige esos datos a objetivos como. AWS Lambda Esto le permite monitorear los eventos que ocurren en los servicios y crear arquitecturas basadas en eventos. Para obtener más información, consulta la Guía del EventBridge usuario de Amazon.

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puedes producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puedes manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición `aws:SourceAccount`global `aws:SourceArn`y las claves de contexto en las políticas de recursos para limitar los permisos que se AWS Amplify otorgan a otro servicio al recurso. Si se utilizan ambas claves contextuales de condición global, el valor `aws:SourceAccount` y la cuenta del valor `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utilicen en la misma declaración de política.

El valor de `aws:SourceArn` debe ser el ARN de ramificación de la aplicación de Amplify. Especifique este valor en el formato `arn:`*`Partition`*`:amplify:`*`Region`*`:`*`Account`*`:apps/`*`AppId`*`/branches/`*`BranchName`*.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si especifica varios recursos, utiliza la clave de condición de contexto

global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

El siguiente ejemplo muestra una política de confianza de roles que puede aplicar para limitar el acceso a cualquier aplicación de Amplify de su cuenta y evitar problemas de suplente confuso. Para utilizar esta política, sustituya el texto rojo en cursiva del ejemplo de política por su propia información.

```
{
  "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"
      }
    }
  }
}
```

El siguiente ejemplo muestra una política de confianza de roles que puede aplicar para limitar el acceso a una aplicación de Amplify concreta de su cuenta y evitar problemas de suplente confuso. Para utilizar esta política, sustituya el texto rojo en cursiva del ejemplo de política por su propia información.

```
{
  "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"
      }
    }
  }
}
```

Prácticas recomendadas de seguridad para Amplify

Amplify proporciona un número de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no suponen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para su entorno, considérelas como recomendaciones útiles en lugar de como normas.

Uso de cookies con el dominio predeterminado de Amplify

Cuando usa Amplify para implementar una aplicación web, Amplify la aloja en el dominio predeterminado `amplifyapp.com`. Podrá ver su aplicación en una URL con el formato `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`.

Para aumentar la seguridad de las aplicaciones de Amplify, el dominio amplifyapp.com se ha registrado en la lista de sufijos públicos (PSL). Para una mayor seguridad, le recomendamos que utilice cookies con un prefijo `__Host-` si alguna vez necesita configurar cookies confidenciales en

el nombre de dominio predeterminado de las aplicaciones de Amplify. Esta práctica le ayudará a proteger su dominio de los intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de configuración de cookies en la red de desarrolladores de Mozilla.

Service Quotas de Amplify Hosting

Las siguientes son las cuotas de servicio para AWS Amplify Hosting. Las cuotas de servicio (anteriormente denominadas límites) establecen el número máximo de recursos u operaciones de servicio para su Cuenta de AWS.

Cuentas de AWS Los nuevos han reducido las cuotas de aplicaciones y trabajos simultáneos. AWS aumenta estas cuotas automáticamente en función del uso que hagas. También puede solicitar un aumento de cuota.

La consola de Service Quotas ofrece información sobre las cuotas de su cuenta. Puede utilizar la consola de Service Quotas para consultar las cuotas predeterminadas y solicitar aumentos de cuota para las cuotas ajustables. Para obtener más información, consulte este tema acerca de cómo solicitar un aumento de cuota en la Guía del usuario de Service Quotas.

Nombre	Valor predeterminado	Ajusta	Descripción
Aplicaciones	Cada región admitida: 25	Sí	El número máximo de aplicaciones que puedes crear en AWS Amplify Console en esta cuenta en la región actual.
Ramificaciones por aplicación	Cada región admitida: 50	No	El número máximo de ramificaciones por aplicación que puede crear en esta cuenta en la región actual.
Tamaño de artefacto de compilación	Cada región admitida: 5 gigabytes	No	El tamaño máximo (en GB) de un artefacto de compilación de aplicaciones. AWS Amplify Console despliega un artefacto de construcción

Nombre	Valor predeterminado	Ajusta	Descripción
			después de una compilación.
Tamaño del artefacto en la memoria caché	Cada región admitida: 5 gigabytes	No	El tamaño máximo (en GB) de un artefacto en la memoria caché.
Trabajos simultáneos	Cada región admitida: 5	Sí	El número máximo de trabajos simultáneos que puede crear en esta cuenta en la región actual.
Dominios por aplicación	Cada región admitida: 5	Sí	El número máximo de dominios por aplicación que puede crear en esta cuenta en la región actual.
Tamaño del artefacto en la memoria caché del entorno	Cada región admitida: 5 gigabytes	No	El tamaño máximo (en GB) del artefacto en la memoria caché del entorno.
Tamaño del archivo ZIP de implementación manual	Cada región admitida: 5 gigabytes	No	El tamaño máximo (en GB) de un archivo ZIP de implementación manual.
Número máximo de creaciones de aplicaciones por hora	Cada región admitida: 25	No	El número máximo de aplicaciones que puedes crear en AWS Amplify Console por hora en esta cuenta en la región actual.

Nombre	Valor predeterminado	Ajusta	Descripción
Tokens de solicitud por segundo	Cada región admitida: 20 000	Sí	El número máximo de tokens de solicitud por segundo de una aplicación. Amplify Hosting asigna tokens a las solicitudes en función de la cantidad de recursos (tiempo de procesamiento y transferencia de datos) que consumen.
Subdominios por dominio	Cada región admitida: 50	No	El número máximo de subdominios por dominio que puede crear en esta cuenta en la región actual.
Webhooks por aplicación	Cada región admitida: 50	Sí	El número máximo de webhooks por aplicación que puede crear en esta cuenta en la región actual.

Para obtener más información acerca de las Service Quotas de Amplify, consulte los puntos de conexión y cuotas de AWS Amplify en Referencia general de AWS.

Solución de problemas de Amplify Hosting

Si surgen problemas o errores de implementación a la hora de trabajar con Amplify Hosting, consulte los temas de esta sección.

Temas

Solución de problemas generales de Amplify

La siguiente información puede ayudarlo a solucionar problemas generales con Amplify Hosting.

Temas

Código de estado HTTP 429 (demasiadas solicitudes)

Amplify controla la cantidad de solicitudes por segundo (RPS) a su sitio web en función del tiempo de procesamiento y la transferencia de datos que consumen las solicitudes entrantes. Si su aplicación devuelve un código de estado HTTP 429, las solicitudes entrantes exceden el tiempo de procesamiento y transferencia de datos asignado a su aplicación. Este límite de aplicaciones se administra mediante la cuota de servicio `REQUEST_TOKENS_PER_SECOND` de Amplify. Para obtener más información sobre las cuotas, consulte Service Quotas de Amplify Hosting.

Para solucionar este problema, recomendamos optimizar la aplicación para reducir la duración de las solicitudes y la transferencia de datos a fin de aumentar la RPS de la aplicación. Por ejemplo, con los mismos 20 000 tokens, una página SSR altamente optimizada que responda en 100 milisegundos puede admitir una RPS más alta en comparación con una página con una latencia superior a 200 milisegundos.

Del mismo modo, una aplicación que devuelva un tamaño de respuesta de 1 MB consumirá más tokens que una aplicación que devuelva uno de 250 KB.

También le recomendamos que aproveche la CloudFront caché de Amazon configurando Cache-Control encabezados que maximizan el tiempo que una respuesta determinada permanece en la memoria caché. Las solicitudes que se atienden desde la CloudFront memoria caché no se tienen en cuenta para el límite de velocidad. Cada CloudFront distribución puede gestionar hasta 250 000 solicitudes por segundo, lo que te permite escalar tu aplicación a un nivel muy alto utilizando la caché. Para obtener más información sobre la CloudFront caché, consulte Optimización del almacenamiento en caché y la disponibilidad en la Guía para CloudFront desarrolladores de Amazon.

La consola Amplify no muestra el estado de compilación ni la hora de la última actualización de mi aplicación

Al navegar a la página Todas las aplicaciones en la consola de Amplify, se muestra un mosaico para cada una de las aplicaciones de la región actual. Si no ves el estado de compilación de una aplicación, como Implementada, ni la hora de la última actualización, significa que la aplicación no tiene ninguna rama de `Production` fase asociada.

Para enumerar las aplicaciones en la consola, Amplify usa la `ListApps` API. Amplify usa el `ProductionBranch.status` atributo para mostrar el estado de la compilación y el `ProductionBranch.lastDeployTime` atributo para mostrar la hora de la última actualización. Para obtener más información sobre esta API, consulta la documentación ProductionBranchde la API de Amplify Hosting.

Usa las siguientes instrucciones para asociar una `Production` etapa a la rama de tu aplicación.

1. Inicia sesión en la consola de Amplify.
2. En la página Todas las aplicaciones, elige la aplicación que deseas actualizar.
3. En el panel de navegación, selecciona Configuración de la aplicación y, a continuación, Configuración de sucursal.
4. En la sección Configuración de sucursal, selecciona Editar.

5. Para la rama de producción, elige el nombre de la sucursal que deseas usar.
6. Seleccione Guardar.
7. Vuelva a la página Todas las aplicaciones. Ahora deberían mostrarse el estado de compilación y la hora de la última actualización de tu aplicación.

No se crean vistas previas web para las nuevas solicitudes de cambios

La función de vistas previas web te permite previsualizar los cambios de las solicitudes de extracción antes de fusionarlos en una rama de integración. Una vista previa web despliega todas las solicitudes de extracción realizadas en tu repositorio en una URL de vista previa única que es diferente de la URL que usa tu sitio principal.

Si has activado las vistas previas web de tu aplicación, pero no se están creando para una nueva PRs, investiga si alguna de las siguientes razones es la causa del problema.

1. Comprueba si tu aplicación ha alcanzado la cuota máxima `Branches per app` de servicio. Para obtener más información sobre las cuotas, consulte Service Quotas de Amplify Hosting.

   Para mantenerte dentro de la cuota predeterminada de 50 sucursales por aplicación, considera habilitar la eliminación automática de sucursales en tu aplicación. Esto evitará que acumules sucursales en tu cuenta que ya no existen en tu repositorio.
2. Si utilizas un GitHub repositorio público y tu aplicación Amplify tiene una función de servicio de IAM asociada, Amplify no crea vistas previas por motivos de seguridad. Por ejemplo, las aplicaciones con backend y aquellas que se implementan en la plataforma de alojamiento de `WEB_COMPUTE` requieren un rol de servicio de IAM. Por lo tanto, si su repositorio es público, no podrá habilitar las vistas previas web para este tipo de aplicaciones.

   Para permitir que las vistas previas web funcionen en tu aplicación, puedes desasociar la función de servicio (si la aplicación no tiene un backend o no es una `WEB_COMPUTE` aplicación) o puedes hacer que el repositorio sea privado. GitHub

Mi implementación manual está bloqueada con un estado pendiente en la consola Amplify

Las implementaciones manuales le permiten publicar su aplicación web con Amplify Hosting sin necesidad de conectarse a un proveedor de Git. Puedes usar una de las siguientes cuatro opciones de implementación.

1. Arrastra y suelta la carpeta de aplicaciones en la consola de Amplify.
2. Arrastra y suelta un archivo.zip (que contiene los artefactos de construcción de tu sitio) en la consola de Amplify.
3. Cargue un archivo.zip (que contiene los artefactos de creación de su sitio) en un bucket de Amazon S3 y conecte el bucket a una aplicación en la consola de Amplify.
4. Usa una URL pública que apunte a un archivo.zip (que contiene los artefactos de creación de tu sitio) en la consola de Amplify.

Somos conscientes de que hay problemas con la función de arrastrar y soltar cuando se utiliza una carpeta de aplicaciones para una implementación manual en la consola Amplify. Estas implementaciones pueden fallar por las siguientes razones.

- Se producen problemas de red transitorios.
- Se produce un cambio local en los archivos durante la carga.
- La sesión del navegador intenta cargar una gran cantidad de activos estáticos simultáneamente.

Mientras trabajamos para mejorar la fiabilidad de nuestras subidas mediante la función de arrastrar y soltar, te recomendamos que utilices un archivo.zip en lugar de arrastrar y soltar las carpetas de la aplicación.

Recomendamos encarecidamente subir un archivo.zip a un bucket de Amazon S3, ya que esto evita la carga de archivos desde la consola Amplify y proporciona una mayor fiabilidad para las implementaciones manuales. La integración de Amplify con Amazon S3 simplifica este proceso. Para obtener más información, consulte Implementación de un sitio web estático en Amplify desde un bucket de Amazon S3.

Solución de problemas de la imagen de compilación de Amazon Linux 2023

La siguiente información puede ayudarle a solucionar problemas con la imagen de compilación de Amazon Linux 2023 (AL2023).

Temas

- Quiero ejecutar las funciones de Amplify con el tiempo de ejecución de Python
- Quiero ejecutar comandos que requieran privilegios raíz o de superusuario

Quiero ejecutar las funciones de Amplify con el tiempo de ejecución de Python

Amplify Hosting ahora usa la imagen de compilación de Amazon Linux 2023 de forma predeterminada al implementar una nueva aplicación. AL2023 viene preinstalado con las versiones 3.8, 3.9, 3.10 y 3.11 de Python.

Para garantizar la compatibilidad con versiones anteriores de la imagen de Amazon Linux 2, la imagen de compilación AL2 023 tiene preinstalados enlaces simbólicos para versiones anteriores de Python.

De forma predeterminada, la versión 3.10 de Python se usa de manera global. Para crear las funciones con una versión específica de Python, ejecute los siguientes comandos en el archivo de especificaciones de compilación de la aplicación.

```
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
```

Quiero ejecutar comandos que requieran privilegios raíz o de superusuario

Si utiliza la imagen de compilación de Amazon Linux 2023 y recibe un error al ejecutar comandos del sistema que requieren privilegios raíz o de superusuario, debe ejecutar estos comandos con el comando `sudo` de Linux. Por ejemplo, si se produce un error al ejecutar `yum install -y gcc`, utilice `sudo yum install -y gcc`.

La imagen de compilación de Amazon Linux 2 utilizaba el usuario root, pero la imagen AL2 023 de Amplify ejecuta el código con un usuario personalizado`amplify`. Amplify otorga a este usuario

privilegios para ejecutar comandos mediante el comando sudo de Linux. La práctica recomendada consiste en usar comandos sudo que requieren privilegios de superusuario.

Solución de problemas de compilación

Si tiene problemas al crear o crear una aplicación Amplify, consulte los temas de esta sección para obtener ayuda.

Temas

Las nuevas confirmaciones en mi repositorio no activan las compilaciones de Amplify

Si las nuevas confirmaciones en tu repositorio de Git no activan las compilaciones de Amplify, verifica que tu webhook siga presente en tu repositorio. Si está presente, consulta el historial de solicitudes de webhooks para ver si hay algún error. Amplify tiene un límite de tamaño de carga útil de 256 KB para los webhooks entrantes. Si envías una confirmación a tu repositorio que contiene una gran cantidad de archivos modificados, podrías superar este límite y provocar que no se activen las compilaciones.

El nombre de mi repositorio no aparece en la consola de Amplify al crear una nueva aplicación

Al crear una nueva aplicación en la consola de Amplify, puede elegir entre los repositorios disponibles de su organización en la página Agregar repositorio y sucursal. Es posible que tu repositorio de destino no aparezca en la lista si no se ha actualizado recientemente. Esto puede ocurrir si tu organización tiene una gran cantidad de repositorios. Para resolver este problema, envía una confirmación al repositorio y, a continuación, actualiza la lista de repositorios en la consola. Esto debería hacer que se muestre el repositorio.

Mi compilación falla debido al `Cannot find module aws-exports` error (solo aplicaciones de primera generación)

Si tu aplicación no encuentra el `aws-exports.js` archivo durante una compilación, aparece el siguiente error.

```
TS2307: Cannot find module 'aws-exports'
```

La interfaz de línea de comandos (CLI) de Amplify genera el `aws-exports.js` archivo durante la compilación del backend. Para resolver este error, debes crear un `aws-exports.js` archivo para usarlo en la compilación. Agrega el siguiente código a la especificación de compilación para crear el archivo:

```
backend:
  phases:
    build:
      commands:
        - "# Execute Amplify CLI with the helper script"
        - amplifyPush --simple
```

Para ver un ejemplo completo de los ajustes de especificación de compilación de una aplicación Amplify, consulte. Referencia de la especificación de compilación de la sintaxis de YAML

Quiero anular un tiempo de espera de compilación

El tiempo de espera de compilación predeterminado es de 30 minutos. Puede anular el tiempo de espera de compilación predeterminado mediante la `_BUILD_TIMEOUT` variable de entorno. El tiempo de espera mínimo de compilación es de 5 minutos. El tiempo de espera máximo de construcción es de 120 minutos.

Para obtener instrucciones sobre cómo configurar una variable de entorno para una aplicación en la consola de Amplify, consulte. Configuración de variables de entorno

Solución de problemas de dominios personalizados

Si tiene algún problema al conectar un dominio personalizado a una aplicación de Amplify, consulte los siguientes temas de esta sección para obtener ayuda.

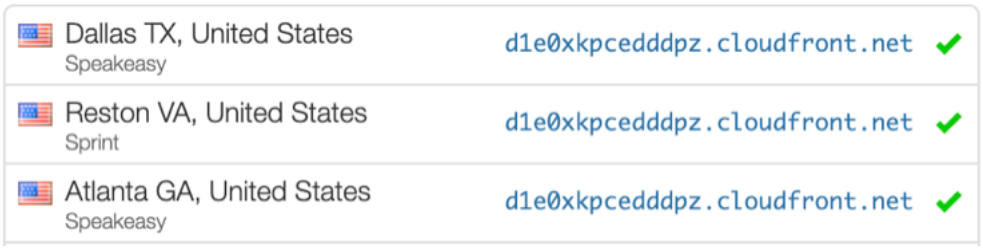
Si no ve una solución a su problema aquí, póngase en contacto con Support. Para obtener más información, consulte Creación de un caso de soporte en la Guía del usuario de AWS Support .

Temas

Necesito comprobar que mi CNAME llega a una resolución

1. Tras actualizar los registros de DNS con su proveedor de dominios externo, puede usar una herramienta como dig o un sitio web gratuito como https://www.whatsmydns.net/ para comprobar que el registro CNAME se resuelve correctamente. En la siguiente captura de pantalla puede ver cómo usar whatsmydns.net para comprobar el registro CNAME del dominio www.ejemplo.com.

2. Elija Buscar y whatsmydns.net mostrará los resultados de su CNAME. La siguiente captura de pantalla es un ejemplo de una lista de resultados que comprueban la resolución correcta del CNAME a una URL de cloudfront.net.



Mi dominio alojado con un tercero está bloqueado en el estado Verificación pendiente

1. Si su dominio personalizado está bloqueado en el estado de pendiente de verificación, compruebe que CNAME los registros se están resolviendo. Consulte el tema de solución de problemas anterior, ¿Cómo puedo comprobar que mi CNAME resuelve, para obtener instrucciones sobre cómo realizar esta tarea.
2. Si las recetas CNAME los registros no se están resolviendo, confirme que CNAME la entrada existe en la configuración de DNS de su proveedor de dominio.

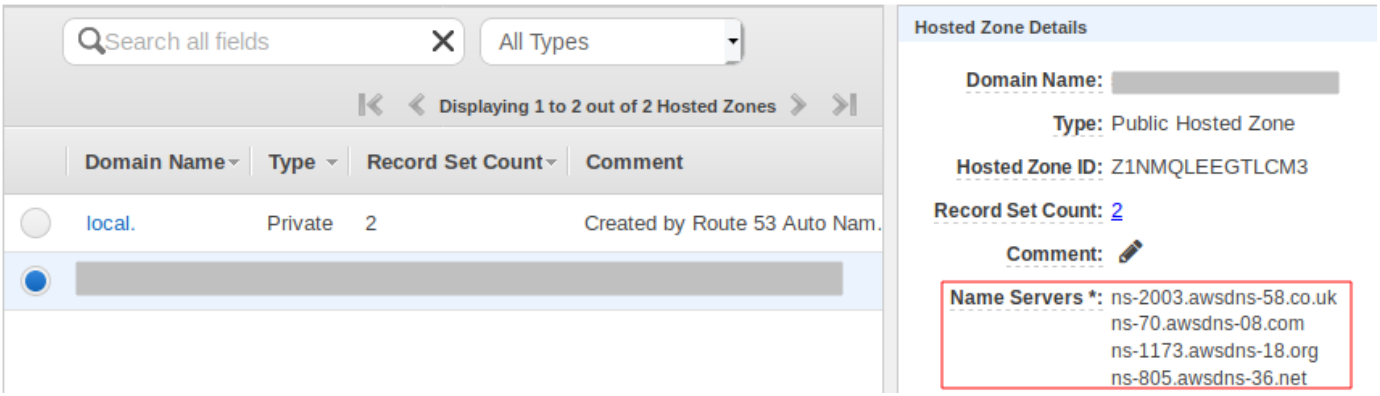
   > **Important**
   >
   > Es importante actualizar tu CNAME registros tan pronto como crees tu dominio personalizado. Una vez creada la aplicación en la consola de Amplify, tu CNAME el registro se comprueba cada pocos minutos para determinar si se resuelve. Si no llega a una resolución transcurrida una hora, la comprobación se realizará cada pocas horas, lo que puede provocar un retraso en que su dominio esté listo para usar. Si agregó o actualizó su CNAME se registra unas horas después de crear la aplicación, lo más probable es que esta sea la causa más probable de que la aplicación se quede atascada en el estado de pendiente de verificación.

3. Si has comprobado que el CNAME el registro existe, es posible que haya un problema con su proveedor de DNS. Puedes ponerte en contacto con el proveedor de DNS para diagnosticar el motivo de la verificación de DNS CNAME no se está resolviendo o puede migrar su DNS a Route 53. Para obtener más información, consulte Establecimiento de Amazon Route 53 como el servicio DNS de un dominio existente.

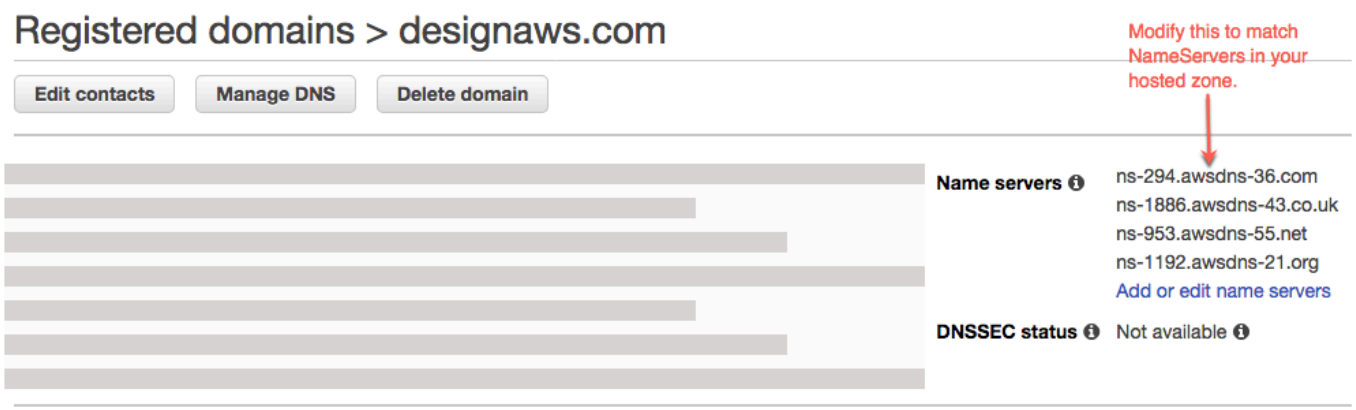
Mi dominio alojado con Amazon Route 53 está bloqueado en estado Verificación pendiente

Si ha transferido su dominio a Amazon Route 53, es posible que su dominio tenga unos servidores de nombres distintos de los emitidos por Amplify al crearse su aplicación. Lleve a cabo los siguientes pasos para diagnosticar la causa del error.

1. Inicie sesión en la consola de Amazon Route 53
2. En el panel de navegación, elija Zonas alojadas y, a continuación, elija el nombre del dominio que desea conectar.
3. Registre los valores del servidor de nombres en la sección Detalles de la zona alojada. Necesita estos valores para completar el siguiente paso. La siguiente captura de pantalla de la consola de Route 53 muestra la ubicación de los valores del servidor de nombres, en la esquina inferior derecha.



4. En el panel de navegación, elija Registered domains. Compruebe que los servidores de nombres que aparecen en la sección Dominios registrados coincidan con los valores de servidor de nombres que ha registrado en la sección Detalles de la zona alojada del paso anterior. Si no coinciden, edite los valores del servidor de nombres para que coincidan con los valores de su Zona alojada. La siguiente captura de pantalla de la consola de Route 53 muestra la ubicación de los valores del servidor de nombres, en el lado derecho.



5. Si así no se resuelve el problema, póngase en contacto con Support. Para obtener más información, consulte Creación de un caso de soporte en la Guía del usuario de AWS Support .

Mi aplicación con subdominios de varios niveles está bloqueada en el estado de verificación pendiente

Si una aplicación con subdominios de varios niveles se bloquea en el estado de verificación pendiente al conectarse a un proveedor de DNS externo, es posible que haya un problema con el formato de tus registros de DNS. Algunos proveedores de DNS añaden automáticamente los sufijos de dominio de segundo nivel (SLD) y dominio de nivel superior (TLD) a tus registros. Si también especificas el dominio en el formato que incluye el SLD y el TLD, esto puede provocar un problema de verificación del dominio.

Cuando conectes un dominio, primero intenta especificar el nombre de dominio con el formato completo proporcionado por Amplify, por ejemplo. `_hash.docs.backend.example.com` Si la configuración de SSL se queda atascada en el estado de verificación pendiente, intenta eliminar el TLD y el SLD de los registros. Por ejemplo, si el formato es completo`_hash.docs.backend.example.com`, especifique. `_hash.docs.backend` Espere de 15 a 30 minutos para permitir que los registros se propaguen. A continuación, utilice una herramienta como MX Toolbox para comprobar si el proceso de verificación funciona.

Mi proveedor de DNS no admite registros A con nombres de dominio totalmente cualificados

Algunos proveedores de DNS no admiten registros A con un nombre de dominio completo (FQDN), como`example.cloudfront.net`. Por ejemplo, Cloudflare A records solo puede escribir IPv4

direcciones y no admiten FQDNs. Para evitar esta limitación, recomendamos utilizar CNAME registros en lugar de A records en tu DNS configuración.

A modo de ejemplo, lo siguiente DNS la configuración utiliza un A record.

```
A     | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Cámbielo a lo siguiente DNS configuración a utilizar CNAME solo registros.

```
CNAME | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Esta solución alternativa le permite apuntar correctamente su dominio de ápex (registro @) a servicios como CloudFront, al mismo tiempo, evitar la limitación exclusiva IPv4 de A records en el sistema de Cloudflare.

Me sale un error CNAMEAlready ExistsException

Si recibes un CNAMEAlreadyExistsExceptionerror, significa que uno de los nombres de host que has intentado conectar (un subdominio o el dominio apex) ya está desplegado en otra distribución de Amazon CloudFront . El origen del error depende de tus proveedores actuales de hosting y DNS.

A CNAME alias, por ejemplo, `example.com` o solo se `sub.example.com` puede asociar a una única CloudFront distribución a la vez. CNAMEAlreadyExistsExceptionEsto indica que su dominio ya está asociado a otra CloudFront distribución, ya sea dentro de la misma o Cuenta de AWS, posiblemente, en una cuenta diferente. El dominio debe estar disociado de la CloudFront distribución anterior antes de que funcione la nueva distribución creada por Amplify Hosting. Es posible que tengas que comprobar más de una cuenta si tú o tu organización poseen varias Cuenta de AWS.

Realice los siguientes pasos para diagnosticar la causa del CNAMEAlreadyExistsExceptionerror.

1. Inicia sesión en la CloudFront consola de Amazon y comprueba que no tienes este dominio implementado en otra distribución. Una sola CNAME el registro se puede adjuntar a una CloudFront distribución a la vez.
2. Si anteriormente implementó el dominio en una CloudFront distribución, debe eliminarlo.
    a. En el menú de navegación izquierdo, elija Distribuciones.
    b. Seleccione el nombre de la distribución que desea editar.

c. Elija la pestaña General. En la sección Settings (Configuración), elija Editar.

d. Elimine el nombre de dominio de Nombre de dominio alternativo (CNAME). A continuación, elija Guardar cambios.

3. Confirme que no existe ninguna otra CloudFront distribución que utilice este dominio en la actual Cuenta de AWS o en otra Cuentas de AWS. Si no interrumpe ningún servicio que se esté ejecutando actualmente, intenta eliminar y volver a crear la zona alojada.

4. Compruebe si este dominio está conectado a una aplicación de Amplify distinta de la que posee. En caso afirmativo, asegúrese de que no intenta reutilizar uno de los nombres de host. Si la utilizas `www.example.com` para otra aplicación, no puedes usarla `www.example.com` con la aplicación a la que te estás conectando actualmente. Puedes usar otros subdominios, como`blog.example.com`.

5. Si este dominio estaba conectado correctamente a otra aplicación y se ha eliminado en la última hora, inténtelo de nuevo cuando haya transcurrido al menos una hora. Si sigues viendo esta excepción después de 6 horas, ponte en contacto con nosotros Support. Para obtener más información, consulte Creación de un caso de soporte en la Guía del usuario de AWS Support .

6. Si administra su dominio a través de Route 53, asegúrese de limpiar todas las zonas alojadas CNAME o ALIAS registros que apuntan a la CloudFront distribución anterior.

7. Tras completar los pasos anteriores, elimina el dominio personalizado de Amplify Hosting y vuelve a empezar con el flujo de trabajo para conectar un dominio personalizado en la consola de Amplify.

Aparece un error de verificación adicional necesaria

Si aparece un error que indica que es necesaria una verificación adicional, significa que AWS Certificate Manager (ACM) necesita información adicional para procesar esta solicitud de certificado. Esto puede suceder como una medida de protección contra el fraude; por ejemplo, cuando el dominio se encuentra dentro de los 1000 mejores sitios web de Alexa. Para proporcionar la información requerida, utilice el Centro de asistencia para contactar con Support. Si no tiene un plan de asistencia técnica, publique un mensaje en el Foro de debate de ACM.

> **Note**
>
> No se puede solicitar un certificado para nombres de dominio propiedad de Amazon como los que terminan en amazonaws.com, cloudfront.net o elasticbeanstalk.com.

Aparece un error 404 en la URL CloudFront

Para atender el tráfico, Amplify Hosting apunta a una CloudFront URL a través de un registro CNAME. En el proceso de conectar una aplicación a un dominio personalizado, la consola de Amplify muestra la CloudFront URL de la aplicación. Sin embargo, no puede acceder a la aplicación directamente mediante esta CloudFront URL. Devuelve un error 404. Su aplicación solo se resuelve mediante la URL de la aplicación de Amplify (por ejemplo, `https://main.d5udybEXAMPLE.amplifyapp.com`) o su dominio personalizado (por ejemplo, `www.example.com`).

Amplify debe enrutar las solicitudes a la ramificación implementada correcta, y para ello usa el nombre de host. Por ejemplo, puede configurar el dominio `www.example.com` para que apunte a la ramificación principal de una aplicación, pero también puede configurar `dev.example.com` para que apunte a la ramificación de desarrollo de la misma aplicación. Por este motivo, deberá visitar su aplicación en función de los subdominios configurados para que Amplify pueda enrutar las solicitudes en consecuencia.

Aparecen errores de certificado SSL o HTTPS cuando visito mi dominio

Si tienes registros DNS de autorización de la autoridad de certificación (CAA) configurados con un proveedor de DNS externo, es posible que AWS Certificate Manager (ACM) no pueda actualizar o volver a emitir los certificados intermedios para tu certificado SSL de dominio personalizado. Para resolver este problema, debe añadir un registro CAA para confiar en, al menos, uno de los dominios de la entidad de certificación de Amazon. El siguiente procedimiento describe los pasos que debe realizar.

Para añadir un registro CAA para confiar en una entidad de certificación de Amazon

1. Configure un registro CAA con su proveedor de dominios para que confíe en, al menos, uno de los dominios de la entidad de certificación de Amazon. Para obtener más información sobre la configuración del registro CAA, consulte la sección Problemas con la autorización de la entidad de certificación (CAA) en la Guía del usuario de AWS Certificate Manager .
2. Use uno de los siguientes métodos para actualizar el certificado SSL:
   - Actualice manualmente mediante la consola de Amplify.

     > **Note**
     >
     > Este método conllevará cierto tiempo de inactividad en su dominio personalizado.

a. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
b. Elija la aplicación a la que desea agregar un registro CAA.
c. En el panel de navegación, elija Configuración de la aplicación, Administración de dominio.
d. En la página Gestión de dominio, elimine el dominio personalizado.
e. Conecte su aplicación de nuevo al dominio personalizado. Este proceso emitirá un nuevo certificado SSL. Ahora, los certificados intermedios pueden ser gestionados por ACM.

   Para volver a conectar la aplicación a su dominio personalizado, realice uno de los siguientes procedimientos según su proveedor de dominio.

   - Agregar un dominio personalizado administrado en Amazon Route 53.
   - Adición de un dominio personalizado administrado por un proveedor de DNS de terceros.
   - Actualizar los registros DNS de un dominio administrado por GoDaddy.

- Póngase en contacto con nosotros Support para que se vuelva a emitir su certificado SSL.

Solución de problemas de renderización del servidor

Si tiene problemas imprevistos al implementar una aplicación SSR con el procesamiento de Amplify Hosting, consulte los siguientes temas de resolución de problemas. Si no encuentras una solución a tu problema aquí, consulta la guía de solución de problemas de computación web SSR en el repositorio de Amplify GitHub Hosting Issues.

Temas

- Necesito ayuda para usar un adaptador de marcos
- Las rutas de la API de periferia permiten que la compilación de Next.js falle
- La regeneración estática incremental bajo demanda no funciona en mi aplicación
- El resultado de compilación de la aplicación supera el tamaño máximo permitido
- Mi compilación falla debido a un error de memoria insuficiente
- El tamaño de respuesta HTTP de mi aplicación es demasiado grande
- ¿Cómo puedo medir el tiempo de inicio de mi aplicación informática a nivel local?

Necesito ayuda para usar un adaptador de marcos

Si tiene problemas para implementar una aplicación de SSR que usa un adaptador de marcos, consulte Uso de adaptadores de código abierto para cualquier marco SSR.

Las rutas de la API de periferia permiten que la compilación de Next.js falle

Actualmente, Amplify no admite las rutas de la API de Edge de Next.js. Debe utilizar middleware APIs y que no sea periférico cuando aloje su aplicación con Amplify.

La regeneración estática incremental bajo demanda no funciona en mi aplicación

A partir de la versión 12.2.0, Next.js admite la regeneración estática incremental (ISR) para purgar manualmente la memoria caché de Next.js de una página específica. Sin embargo, Amplify no admite actualmente ISR bajo demanda. Si su aplicación utiliza la revalidación bajo demanda de Next.js, esta característica no funcionará cuando implemente su aplicación en Amplify.

El resultado de compilación de la aplicación supera el tamaño máximo permitido

Actualmente, el tamaño máximo de salida de compilación que Amplify admite para las aplicaciones SSR es de 220 MB. Si recibe un mensaje de error que indica que el tamaño del resultado de compilación de la aplicación supera el tamaño máximo permitido, debe tomar medidas para reducirlo.

Para reducir el tamaño del resultado de compilación de una aplicación, puede inspeccionar los artefactos de compilación e identificar cualquier dependencia importante que quiera actualizar o eliminar. Primero, descargue los artefactos de compilación en su equipo local. A continuación, compruebe el tamaño de los directorios. Por ejemplo, el directorio `node_modules` puede contener binarios como `@swc` y `@esbuild` a los que hacen referencia los archivos de tiempo de ejecución del servidor Next.js. Como estos binarios no son necesarios en tiempo de ejecución, puede eliminarlos después de la compilación.

Sigue las siguientes instrucciones para descargar el resultado de la compilación de una aplicación e inspeccionar el tamaño de los directorios mediante la AWS Command Line Interface (CLI).

Descarga e inspección del resultado de la compilación de una aplicación Next.js

1. Abra una ventana del terminal y ejecute el siguiente comando. Cambie el identificador de la aplicación, el nombre de la ramificación y el identificador del trabajo e ingrese su propia información. Para el identificador del trabajo, use el número de la compilación fallida que está investigando.

```
aws amplify get-job --app-id abcd1234 --branch-name main --job-id 2
```

2. En la salida del terminal, localice la URL del artefacto prefirmado en la sección `job`, `steps`, `stepName: "BUILD"`. La URL aparece resaltada en rojo en el siguiente ejemplo de salida.

```
"job": {
    "summary": {
        "jobArn": "arn:aws:amplify:us-west-2:111122223333:apps/abcd1234/main/
jobs/0000000002",
        "jobId": "2",
        "commitId": "HEAD",
        "commitTime": "2024-02-08T21:54:42.398000+00:00",
        "startTime": "2024-02-08T21:54:42.674000+00:00",
        "status": "SUCCEED",
        "endTime": "2024-02-08T22:03:58.071000+00:00"
    },
    "steps": [
        {
            "stepName": "BUILD",
            "startTime": "2024-02-08T21:54:42.693000+00:00",
            "status": "SUCCEED",
            "endTime": "2024-02-08T22:03:30.897000+00:00",
            "logUrl": "https://aws-amplify-prod-us-west-2-artifacts.s3.us-
west-2.amazonaws.com/abcd1234/main/0000000002/BUILD/log.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2V...Example
```

3. Copie y pegue la URL en una ventana del navegador. Un archivo `artifacts.zip` se descarga en el equipo local. Este es el resultado de su compilación.

4. Ejecute el comando de uso de disco du para inspeccionar el tamaño de los directorios. El siguiente comando de ejemplo devuelve el tamaño de los directorios `compute` y `static`.

```
du -csh compute static
```

A continuación se muestra un ejemplo del resultado con información sobre el tamaño de los directorios `compute` y `static`.

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

5. Abra el directorio `compute` y localice la carpeta `node_modules`. Revise las dependencias para ver si hay archivos que pueda actualizar o eliminar para reducir el tamaño de la carpeta.
6. Si la aplicación contiene archivos binarios que no son necesarios en el tiempo de ejecución, elimínelos después de la compilación al agregar los siguientes comandos a la sección de compilación del archivo `amplify.yml` de su aplicación.

```
- 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
```

El siguiente es un ejemplo de la sección de comandos de compilación de un archivo `amplify.yml` con estos comandos agregados después de ejecutar una compilación de producción.

```
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
```

Mi compilación falla debido a un error de memoria insuficiente

Next.js le permite almacenar en la memoria caché los artefactos de compilación para mejorar el rendimiento en las compilaciones posteriores. Además, el AWS CodeBuild contenedor de Amplify comprime y carga esta caché en Amazon S3, en su nombre, para mejorar el rendimiento de la compilación posterior. Esto podría provocar un error de compilación debido a un error de memoria insuficiente.

Realice las siguientes acciones para evitar que su aplicación supere el límite de memoria durante la fase de compilación. En primer lugar, elimine `.next/cache/**/*` de la sección cache.paths de su configuración de compilación. A continuación, elimine la variable de entorno `NODE_OPTIONS` de su archivo de configuración de compilación. En su lugar, configure la variable de entorno `NODE_OPTIONS` en la consola de Amplify para definir el límite máximo de memoria del nodo. Para obtener más información sobre cómo configurar variables de entorno utilizando la consola de Amplify, consulte Configuración de variables de entorno.

Después de realizar estos cambios, intente realizar la compilación de nuevo. Si tiene éxito, añada de nuevo `.next/cache/**/*` a la sección cache.paths del archivo de configuración de compilación.

Para obtener más información sobre la configuración de la caché de Next.js para mejorar el rendimiento de la compilación, consulte AWS CodeBuild en el sitio web Next.js.

El tamaño de respuesta HTTP de mi aplicación es demasiado grande

Actualmente, el tamaño máximo de respuesta que Amplify admite para las aplicaciones Next.js 12 y versiones posteriores que utilizan la plataforma de procesamiento web es de 5,72 MB. Las respuestas que superen ese límite devuelven 504 errores sin contenido a los clientes.

¿Cómo puedo medir el tiempo de inicio de mi aplicación informática a nivel local?

Sigue las siguientes instrucciones para determinar el tiempo de inicialización o inicio local de tu aplicación de cómputo Next.js 12 o posterior. Puedes comparar el rendimiento de tu aplicación a nivel local con el de Amplify Hosting y utilizar los resultados para mejorar el rendimiento de tu aplicación.

Para medir el tiempo de inicialización de una aplicación de Next.js Compute de forma local

1. Abre el `next.config.js` archivo de la aplicación y establece la `output` opción de la `standalone` siguiente manera.

```
** @type {import('next').NextConfig} */
const nextConfig = {
  // Other options
  output: "standalone",
};
```

```
module.exports = nextConfig;
```

2. Abre una ventana de terminal y ejecuta el siguiente comando para crear la aplicación.

```
next build
```

3. Ejecuta el siguiente comando para copiar la `.next/static` carpeta en`.next/standalone/.next/static`.

```
cp -r .next/static .next/standalone/.next/static
```

4. Ejecute el siguiente comando para copiar la `public` carpeta a`.next/standalone/public`.

```
cp -r public .next/standalone/public
```

5. Ejecute el siguiente comando para iniciar el servidor Next.js.

```
node .next/standalone/server.js
```

6. Observe el tiempo que transcurre entre la ejecución del comando en el paso 5 y el inicio del servidor. Cuando el servidor escucha en un puerto, debe imprimir el siguiente mensaje.

```
Listening on port 3000
```

7. En el paso 6, observe cuánto tardan en cargarse los demás módulos después de iniciar el servidor. Por ejemplo, bibliotecas como estas `bugsnag` tardan entre 10 y 12 segundos en cargarse. Una vez cargada, mostrará el mensaje de confirmación`[bugsnag] loaded`.
8. Sume las duraciones de los pasos 6 y 7. Este resultado es la hora de inicialización o inicio local de tu aplicación de Compute.

Solución de problemas de redirecciones y reescrituras

Si tienes problemas al configurar redireccionamientos y reescrituras para una aplicación de Amplify, consulta los temas de esta sección para obtener ayuda.

Temas

- Se deniega el acceso a determinadas rutas incluso con la regla de redireccionamiento de SPA.
- Quiero configurar un proxy inverso a una API

Se deniega el acceso a determinadas rutas incluso con la regla de redireccionamiento de SPA.

Si recibes un error de acceso denegado para determinadas rutas con una regla de redireccionamiento de SPA, es `baseDirectory` posible que no esté configurada correctamente en la configuración de compilación de la aplicación. Por ejemplo, si la interfaz de tu aplicación está integrada en el `build` directorio, la configuración de compilación también debe apuntar al `build` directorio. En el siguiente ejemplo de especificación de compilación, se muestra esta configuración.

```
frontend:
  artifacts:
    baseDirectory: build
    files:
        - "**/*"
```

Para ver un ejemplo completo de los ajustes de especificación de compilación de una aplicación Amplify, consulta Referencia de la especificación de compilación de la sintaxis de YAML

Quiero configurar un proxy inverso a una API

Puede usar el siguiente JSON para configurar un proxy inverso para un punto final dinámico.

```
[
  {
    "source": "/documents/<*>",
    "target": "https://otherdomain/resource/<*>",
    "status": "200",
    "condition": null
  }
]
```

Para ver un ejemplo básico de cómo crear un proxy inverso para tu aplicación Amplify en una API de terceros, consulta. Reescritura de proxy inverso

Solución de problemas de almacenamiento en caché

Si tiene problemas de almacenamiento en caché en una aplicación de Amplify, consulte los temas de esta sección para obtener ayuda.

Temas

Quiero reducir el tamaño de la memoria caché de una aplicación

Si utilizas la caché, es posible que estés almacenando en caché archivos intermedios que no se limpian entre compilaciones. El almacenamiento en caché de estos archivos que se utilizan con poca frecuencia aumentará el tamaño de la memoria caché. Para evitarlo, puedes excluir carpetas específicas del almacenamiento en caché mediante la `!` directiva de la `cache` sección de especificaciones de compilación de tu aplicación.

En el siguiente ejemplo de configuración de compilación, se muestra cómo usar la `!` directiva para especificar una carpeta que no quieres almacenar en caché.

```
cache:
  paths:
    - node_modules/**/*
    - "!node_modules/path/not/to/cache"
```

Al almacenar en caché la `node_modules` carpeta, `node_modules/.cache` se omite de forma predeterminada.

Para ver un ejemplo completo de los ajustes de especificación de compilación de una aplicación Amplify, consulta Referencia de la especificación de compilación de la sintaxis de YAML

Quiero deshabilitar la lectura desde la memoria caché de una aplicación

Si quieres deshabilitar la lectura desde la caché de una aplicación, elimina la sección de caché de la especificación de compilación de la aplicación.

AWS Amplify Referencia de hospedaje

Utilice los temas de esta sección para encontrar material de referencia detallado sobre AWS Amplify.

Temas

AWS CloudFormation apoyo

Utilice AWS CloudFormation plantillas para aprovisionar los recursos de Amplify, lo que permite despliegues de aplicaciones web confiables y repetibles. AWS CloudFormation proporciona un lenguaje común para describir y aprovisionar todos los recursos de infraestructura de su entorno de nube y simplifica la implementación en varias AWS cuentas o regiones con solo un par de clics.

Para Amplify Hosting, consulte la documentación de Amplify. CloudFormation Para Amplify Studio, consulta la documentación de Amplify UI Builder. CloudFormation

AWS Command Line Interface soporte

AWS Command Line Interface Utilícela para crear aplicaciones Amplify mediante programación desde la línea de comandos. Para obtener más información, consulte la documentación de AWS CLI .

Servicio de asistencia para el etiquetado de recursos

Puedes usar AWS Command Line Interface para etiquetar los recursos de Amplify. Para obtener más información, consulte la documentación de etiquetado de recursos de AWS CLI .

API de Amplify Hosting

Esta referencia ofrece descripciones de las acciones y tipos de datos de la API de Amplify Hosting. Para obtener más información, consulte la documentación de referencia de la API de Amplify.

Historial de documentos para AWS Amplify

En la siguiente tabla se describen los cambios importantes en la documentación desde la última versión de. AWS Amplify

- Última actualización de la documentación: 17 de febrero de 2025

Cambio	Descripción	Fecha
Nuevo tema: agregar una función de SSR Compute para permitir el acceso a AWS los recursos	Se agregó el Añadir un rol de SSR Compute para permitir el acceso a los recursos AWS tema para describir cómo crear y asociar un rol de Amplify SSR Compute a una aplicación para dar acceso al servicio Amplify Compute a los recursos. AWS	17 de febrero de 2025
Nuevo capítulo sobre AWS WAF Cómo proteger tus aplicaciones de Amplify	Se agregó el Soporte de firewall para sitios alojados capítulo para describir la integración de Amplify con AWS WAF (en versión preliminar), que le permite proteger sus aplicaciones web con una lista de control de acceso a la web (ACL web).	18 de diciembre de 2024
Tema de políticas administr adas actualizado	Se ha actualizado el tema AWS políticas administr adas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	14 de noviembre de 2024

Cambio	Descripción	Fecha
Se actualizó el tema de soporte de Amplify para Next.js	Se actualizó el Compatibilidad de Amplify con Next.js tema para describir la compatibilidad de Amplify con la versión 15 de Next.js.	6 de noviembre de 2024
Nuevo capítulo sobre la implementación de un sitio web estático en Amplify desde Amazon S3	Se agregó el Implementación de un sitio web estático en Amplify desde un bucket de Amazon S3 capítulo para describir la nueva integración de Amplify con Amazon S3, que le permite alojar contenido estático de sitios web almacenado en S3 con solo unos pocos clics.	16 de octubre de 2024
Nuevo capítulo sobre la administración de la configuración de caché	Se agregó el capítulo Administrar la configuración de caché de una aplicación, que describe el comportamiento de almacenamiento en caché predeterminado de Amplify y cómo este aplica las políticas de caché administrada al contenido.	13 de agosto de 2024
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	18 de julio de 2024

Cambio	Descripción	Fecha
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	31 de mayo de 2024
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	17 de abril de 2024
Capítulo de introducción actualizado	Se actualizó el capítulo Introducción a la implementación de una aplicación en Amplify Hosting para usar una aplicación Next.js de ejemplo en el tutorial.	12 de abril de 2024
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	5 de abril de 2024
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	4 de abril de 2024

Cambio	Descripción	Fecha
Nuevo capítulo sobre solución de problemas	Se agregó el capítulo Solución de problemas de Amplify Hosting para describir cómo solucionar los problemas que surjan con las aplicaciones implementadas en Amplify Hosting.	2 de abril de 2024
Nueva compatibilidad con certificados SSL/TLS personalizados	Se agregó el tema Uso de certificados de SSL/TLS al capítulo Configuración de dominios personalizados para describir la compatibilidad de Amplify con certificados SSL/TLS personalizados al conectar una aplicación a un dominio personalizado.	20 de febrero de 2024
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	2 de enero de 2024
Nueva compatibilidad con marcos de SSR	Se actualizó el tema Implementación de aplicaciones renderizadas del servidor con Amplify Hosting para describir la compatibilidad de Amplify con cualquier marco SSR basado en JavaScript con un adaptador de código abierto.	19 de noviembre de 2023

Cambio	Descripción	Fecha
Nueva compatibilidad con el lanzamiento de la caracterí stica de optimización de imágenes	Se agregó el tema Optimizac ión de imágenes para aplicaciones de SSR para describir la compatibilidad integrada para la optimización de imágenes para aplicacio nes representadas del servidor.	19 de noviembre de 2023
Tema de políticas administr adas actualizado	Se ha actualizado el tema AWS políticas administr adas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	17 de noviembre de 2023
Tema de políticas administr adas actualizado	Se ha actualizado el tema AWS políticas administr adas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	6 de noviembre de 2023
Nuevo tema de subdominios comodín	Se ha agregado el tema Configuración de subdomini os comodín para describir la compatibilidad con los subdominios comodín en los dominios personalizados.	6 de noviembre de 2023

Cambio	Descripción	Fecha
Nueva política administrada	Se ha actualizado el AWS políticas administradas para AWS Amplify tema para describir la nueva política AmplifyBackendDeployFullAccess AWS gestionada de Amplify.	8 de octubre de 2023
Lanzamiento de una nueva característica de compatibilidad con marcos monorepo	Se ha actualizado el tema Modificar la configuración de compilación de monorepo para describir la compatibilidad con la implementación de aplicaciones en monorepos creados con npm workspace, pnpm workspace, Yarn workspace, Nx y Turborepo.	19 de junio de 2023
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	1 de junio de 2023
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	24 de febrero de 2023

Cambio	Descripción	Fecha
Se ha actualizado el capítulo de representación en el lado del servidor	Se ha actualizado el capítulo Implementación de aplicaciones renderizadas del servidor con Amplify Hosting para describir los cambios recientes en la compatibilidad de Amplify con las versiones 12 y 13 de Next.js.	17 de noviembre de 2022
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	30 de agosto de 2022
Tema de políticas administradas actualizado	Se ha actualizado el tema Creación del backend de una aplicación para describir cómo implementar un backend con Amplify Studio.	23 de agosto de 2022
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	27 de abril de 2022
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	17 de abril de 2022

Cambio	Descripción	Fecha
Lanzamiento de una nueva función GitHub de la aplicación	Se agregó el Configuración del acceso de Amplify a los repositorios GitHub tema para describir la nueva GitHub aplicación para autorizar el acceso de Amplify a GitHub tu repositorio.	5 de abril de 2022
Lanzamiento de la nueva característica Amplify Studio	Se ha actualizado el tema Bienvenido a AWS Amplify Hosting para describir las actualizaciones de Amplify Studio, con un diseñador visual para la creación de componentes de interfaz de usuario que puede conectar a sus datos de backend.	2 de diciembre de 2021
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas gestionadas de AWS que facilitan la compatibilidad de Amplify con Amplify Studio.	2 de diciembre de 2021
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	8 de noviembre de 2021

Cambio	Descripción	Fecha
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas administradas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	27 de septiembre de 2021
Nuevo tema de políticas administradas	Se agregó el AWS políticas administradas para AWS Amplify tema para describir las políticas AWS administradas de Amplify y los cambios recientes en esas políticas.	28 de julio de 2021
Se ha actualizado el capítulo de representación en el lado del servidor	Se ha actualizado el capítulo Implementación de aplicaciones renderizadas del servidor con Amplify Hosting para describir la nueva compatibilidad con las versiones 10.x.x y 11 de Next.js.	22 de julio de 2021
Se ha actualizado el capítulo sobre configuración de ajustes de compilación	Se ha agregado el tema Modificar la configuración de compilación de monorepo para describir cómo configurar los ajustes de compilación y la nueva variable de entorno `AMPLIFY_MONOREPO_APP_ROOT` al implementar una aplicación monorepo con Amplify.	20 de julio de 2021

Cambio	Descripción	Fecha
Se ha actualizado el capítulo sobre implementaciones de ramificación de características	Se ha agregado el tema Generación automática de configuración de Amplify en tiempo de compilación (solo para aplicaciones de Gen 1) para describir cómo generar automáticamente el archivo `aws-exports.js` en tiempo de compilación. Se ha agregado el tema Compilaciones de backend condicionales (solo para aplicaciones de Gen 1) para describir cómo habilitar las compilaciones de backend condicionales. Se ha agregado el tema Use los backends de Amplify en todas las aplicaciones (solo aplicaciones de Gen 1) para describir cómo reutilizar los backends existentes para crear una nueva aplicación, conectar una ramificación nueva a una aplicación existente o actualizar un frontend existente para que apunte a un entorno de backend distinto.	30 de junio de 2021

Cambio	Descripción	Fecha
Capítulo de seguridad actualizado	Se ha agregado el tema Protección de datos en Amplify para describir cómo aplicar el modelo de responsabilidad compartida y cómo Amplify usa el cifrado para proteger sus datos en reposo y en tránsito.	3 de junio de 2021
Nuevo lanzamiento de compatibilidad con la caracterí stica SSR	Se ha agregado el capítulo Implementación de aplicacio nes renderizadas del servidor con Amplify Hosting para describir la compatibilidad de Amplify con aplicaciones web creadas con Next.js, que emplean representación en el lado del servidor (SSR).	18 de mayo de 2021
Nuevo capítulo de seguridad	Se ha agregado el capítulo Seguridad en Amplify para describir cómo aplicar el modelo de responsabilidad compartida al usar Amplify y cómo configurar Amplify para cumplir sus objetivos de seguridad, así como de conformidad.	26 de marzo de 2021

Cambio	Descripción	Fecha
Se ha actualizado el tema de compilaciones personalizadas	Se ha actualizado el tema Imágenes de compilación personalizadas y actualizaciones de paquetes en tiempo real para describir cómo configurar una imagen de compilación personalizada alojada en Amazon Elastic Container Registry Public.	12 de marzo de 2021
Se ha actualizado el tema de supervisión	Se ha actualizado el tema Monitorización para describir cómo acceder a los datos de CloudWatch las métricas de Amazon y configurar las alarmas.	2 de febrero de 2021
Nuevo tema de CloudTrail registro	Se agregó el AWS CloudTrail tema Cómo registrar las llamadas a la API Amplify utilizando para describir cómo AWS CloudTrail captura y registra todas las acciones de la API para la referencia de la API de la AWS Amplify consola y la referencia de la API de la interfaz de usuario del AWS Amplify administrador.	2 de febrero de 2021

Cambio	Descripción	Fecha
Lanzamiento de nueva característica de interfaz de usuario de administración	Se ha actualizado el tema Bienvenido a AWS Amplify Hosting para describir la nueva interfaz de usuario de administración, una interfaz visual que permite a los desarrolladores de frontend web y móvil crear, además de gestionar backends de aplicaciones fuera de AWS Management Console.	1 de diciembre de 2020
Lanzamiento de la nueva característica de modo de rendimiento	Se ha actualizado el tema Gestionar el rendimiento de las aplicaciones para describir cómo activar el modo de rendimiento y optimizarlo con el fin de mejorar el rendimien to del alojamiento.	4 de noviembre de 2020
Se ha actualizado el tema de encabezados personalizados	Se ha actualizado el tema Encabezados personali zados para describir cómo definir encabezados personali zados en una aplicación de Amplify mediante la consola o editando un archivo YML.	28 de octubre de 2020

Cambio	Descripción	Fecha
Lanzamiento de la nueva característica de subdominios automáticos	Se ha agregado el tema Configurar subdominios automáticos para un dominio personalizado de Route 53, que describe el uso de la característica de implement aciones de ramificación basadas en patrones para aplicaciones conectadas a dominios personalizados de Amazon Route 53. Se ha agregado el tema Acceso a la vista previa web con subdominios para describir cómo configurar las vistas previas web de solicitudes de extracción de modo que sean accesibles desde subdomini os.	20 de junio de 2020
Nuevo tema de notificaciones	Se ha agregado el tema Notificaciones para describir cómo configurar las notificac iones por correo electrónico de una aplicación de Amplify con el fin de alertar a las partes interesadas o a los miembros del equipo cuando se realicen compilaciones correctas o fallidas.	20 de junio de 2020

Cambio	Descripción	Fecha
Se ha actualizado el tema de dominios personalizados	Se ha actualizado el Configuración de dominios personalizados tema para mejorar los procedimientos de adición de dominios personalizados en Amazon Route 53 y Google Domains. GoDaddy Esta actualización también incluye nueva información de solución de problemas a la hora de configurar dominios personalizados.	12 de mayo de 2020
AWS Amplify lanzamiento	Esta versión presenta Amplify.	26 de noviembre de 2018

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.