Guía del usuario

AWS Amplify Hospedaje

AWS Amplify Hospedaje: Guía del usuario

Table of Contents

Le damos la bienvenida a AWS Amplify Hosting

AWS Amplify es un conjunto de herramientas y funciones diseñadas específicamente que permiten a los desarrolladores web y móviles frontend crear, rápida y fácilmente, aplicaciones de pila completa en AWS. Amplify proporciona dos servicios: Amplify Hosting y Amplify Studio. Amplify Hosting proporciona un flujo de trabajo basado en Git para alojar aplicaciones web sin servidor de pila completa con implementación continua. Esta guía del usuario proporciona toda la información necesaria para empezar a usar Amplify Hosting.

Características de Amplify Hosting

- Amplify Hosting es compatible con los marcos SPA comunes, por ejemplo, React, Angular, Vue.js, Ionic y Ember, así como con generadores de sitios estáticos como Gatsby, Eleventy, Hugo y Jekyll. VuePress
- Administrar entornos de producción y ensayo para su frontend y backend conectando nuevas ramificaciones. Consulte implementaciones de ramificaciones de características.
- Conecte su aplicación a un dominio personalizado. Consulte configurar dominios personalizados.
- Implemente y aloje aplicaciones web de SSR. Amplify Hosting detecta automáticamente las aplicaciones creadas con el marco Next.js.

  Amplify también es compatible con cualquier marco de 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. Hay un adaptador disponible para implementar una aplicación de Nuxt en Amplify.
- Previsualice los cambios en las revisiones de código configurando vistas previas de las solicitudes de extracción.
- Mejore la calidad de su aplicación con pruebas integrales. Consulte, end-to-end probando.
- Proteja su aplicación web mediante contraseña para poder trabajar en nuevas características sin hacer que estén accesibles públicamente. Consulte restringir el acceso.
- Configure reescrituras y redireccionamientos para mantener las clasificaciones de SEO y dirigir el tráfico en función de las necesidades de su aplicación cliente. Consulte usar redireccionamientos.
- 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 funciones de alojamiento de Amplify, consulte el tutorial Introducción al código existente. Tras completar el tutorial, podrás conectar tu repositorio de git (GitHub GitLab, BitBucket Cloud yAWS CodeCommit) para configurar el despliegue continuo. Como alternativa, puede comenzar con uno de los ejemplos de implementación continua de pila completa.

Amplify Studio

Puede acceder a Amplify Studio desde la consola de AWS Amplify en la AWS Management Console. Amplify Studio es un entorno de desarrollo visual que simplifica la creación de aplicaciones web y móviles escalables de pila completa. 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, crea el backend de la aplicación y, a continuación, conecta los dos componentes. Consulte la guía del usuario de Amplify Studio en la documentación de Amplify.

Características de Amplify Studio

- El modelado visual de datos le permite centrarse en los objetos específicos de su dominio en lugar de en la infraestructura en la nube.
- Configure la autenticación de su aplicación.
- Autorización potente y fácilmente comprensible.
- Configura todas nfrastructure-as-code las funciones de backend con. AWS CloudFormation
- Funciona con la interfaz de línea de comandos (CLI) de Amplify. Todas las actualizaciones que realice en Studio se pueden incorporar a la CLI.
- Envíe invitaciones por correo electrónico a los usuarios para configurar y gestionar el backend. Estos usuarios también podrán iniciar sesión en la CLI de Amplify con su correo electrónico.
- Administración de contenido con soporte para Markdown.
- Gestione los usuarios y grupos de su aplicación.
- Use el diseñador visual de Studio para crear componentes de interfaz de usuario de frontend. Elija entre decenas de diseños en la biblioteca de componentes prediseñados de interfaz de usuario.
- Importe a Studio prototipos de Figma creados por diseñadores como código de React.
- Personalice la interfaz de usuario de frontend con temas para aplicar estilos globales a los componentes de su aplicación.

- Configure y pruebe los componentes de la interfaz de usuario directamente en Studio para comprobar cómo se actualizan y muestran los datos.
- Vincule el backend conectado a la nube con la interfaz de usuario de frontend en unos sencillos pasos.

Introducción a Amplify Studio

No necesita una cuenta de AWS para empezar a crear un backend usando Studio. Puede empezar a modelar datos para su backend de forma local sin necesidad de tener una cuenta de AWS.

Si tiene una cuenta de AWS, podrá acceder a un conjunto ampliado de funciones de Studio para gestionar su entorno de backend. También podrá usar el diseñador visual para crear componentes de interfaz de usuario de frontend y conectarlos al backend de su aplicación. Para obtener más información, consulte Introducción en la documentación de Amplify.

Aplicaciones web modernas SPA

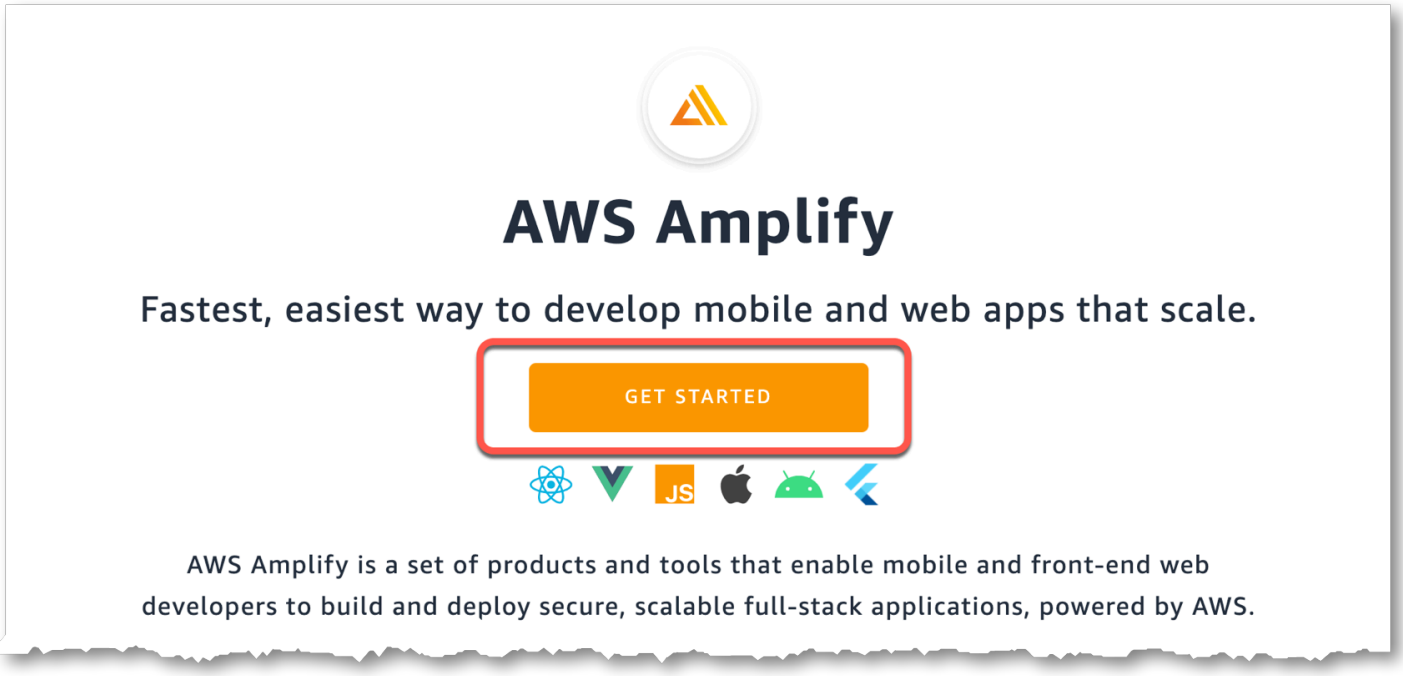
Esta guía del usuario está destinada a clientes con conocimientos básicos sobre las aplicaciones web modernas de página única (SPA). Las aplicaciones web modernas se construyen como SPA con un paquete para todos los componentes de la aplicación en archivos estáticos. Las arquitecturas web de servidores de cliente tradicionales conducen a experiencias pobres; cada clic o búsqueda necesita un trayecto de recorrido al servidor, volver a representar la aplicación completa. Las aplicaciones web modernas ofrecen una experiencia de usuario similar a la de una aplicación nativa, ya que ofrecen la interfaz de usuario o interfaz de usuario de la aplicación de manera eficiente a los navegadores como JavaScript archivos HTML/HTML prediseñados que, a su vez, pueden invocar la funcionalidad de backend sin tener que volver a cargar la página.

La funcionalidad de una aplicación web moderna se suele distribuir entre varios lugares, tales como base de datos, servicios de autenticación, código de frontend que se ejecuta en el navegador y lógica de negocio de backend o funciones AWS Lambda, que se ejecutan en la nube. Esto hace que la implementación de aplicaciones sea compleja y que consuma mucho tiempo, ya que los desarrolladores tienen que coordinar con cuidado implementaciones en el frontend y en el backend para evitar implementaciones parciales o fallidas. Amplify simplifica la implementación del frontend y backend en un único flujo de trabajo.

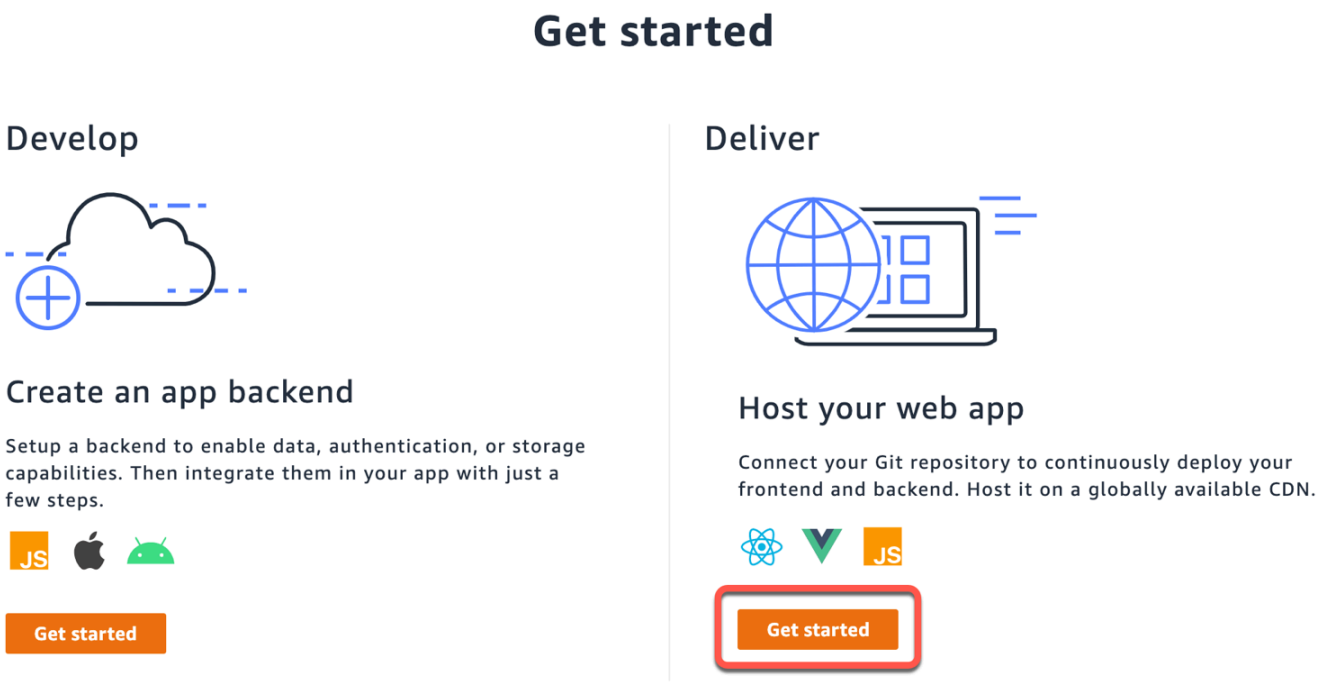
Introducción al código existente

En este tutorial, aprenderá a compilar, implementar y alojar de forma continua una aplicación web moderna. Las aplicaciones web modernas incluyen marcos de trabajo single-page application (SPA) (por ejemplo, React, Angular o Vue) y generadores de sitios estáticos (SSG) (por ejemplo, Hugo, Jekyll o Gatsby). Amplify Hosting también admite aplicaciones web que utilizan renderización de servidor (SSR) y que se crean con Next.js.

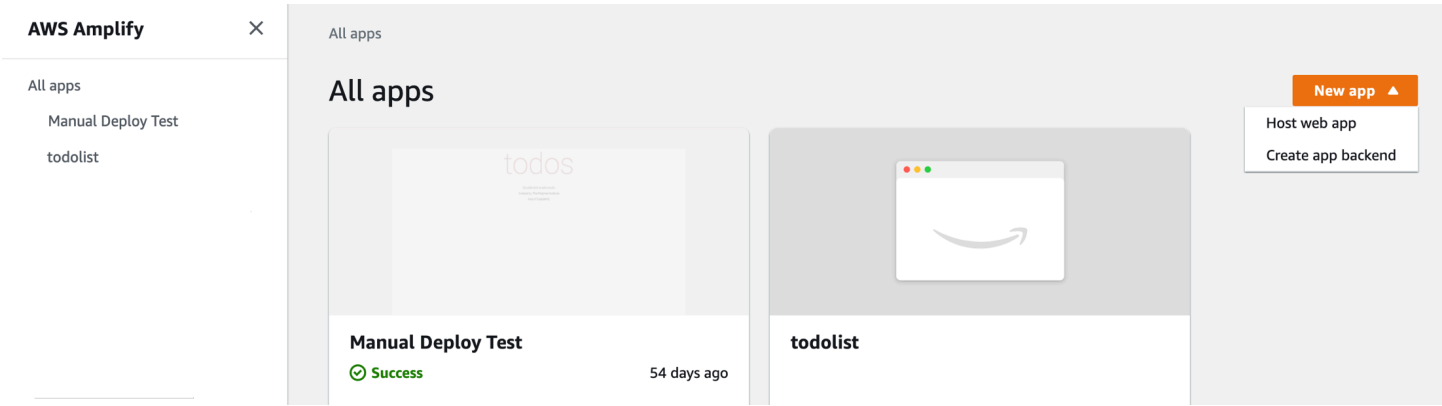
Para empezar, inicie sesión en la consola de Amplify. Si comienza desde la página de inicio de AWS Amplify, elija Primeros pasos en la parte superior de la página.



A continuación, elija Primeros pasos en Entrega.

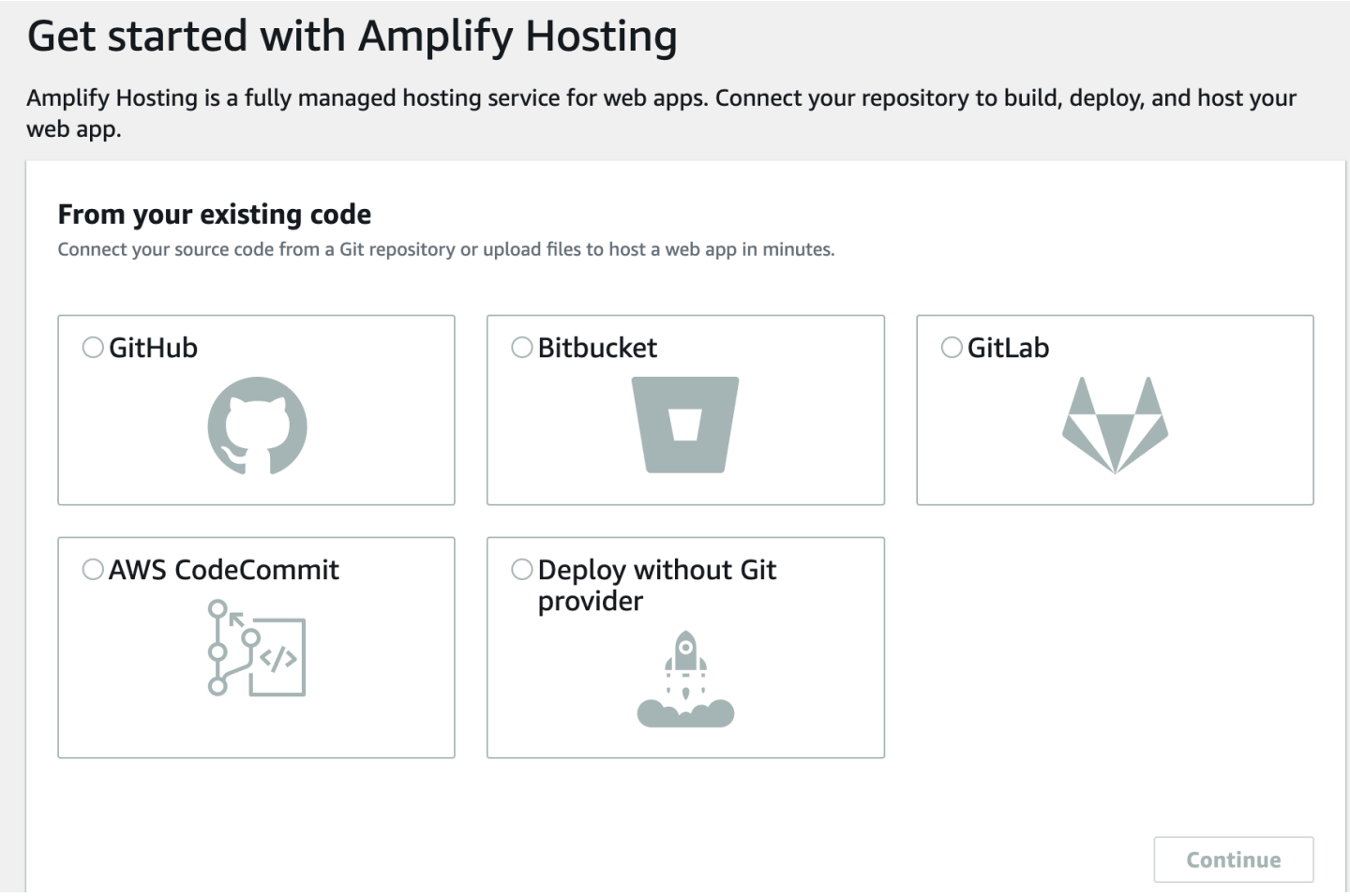


Si comienza desde la página Todas las aplicaciones, elija Nueva aplicación y, a continuación, Alojar aplicación web en la esquina superior derecha.



Paso 1: Conectar un repositorio

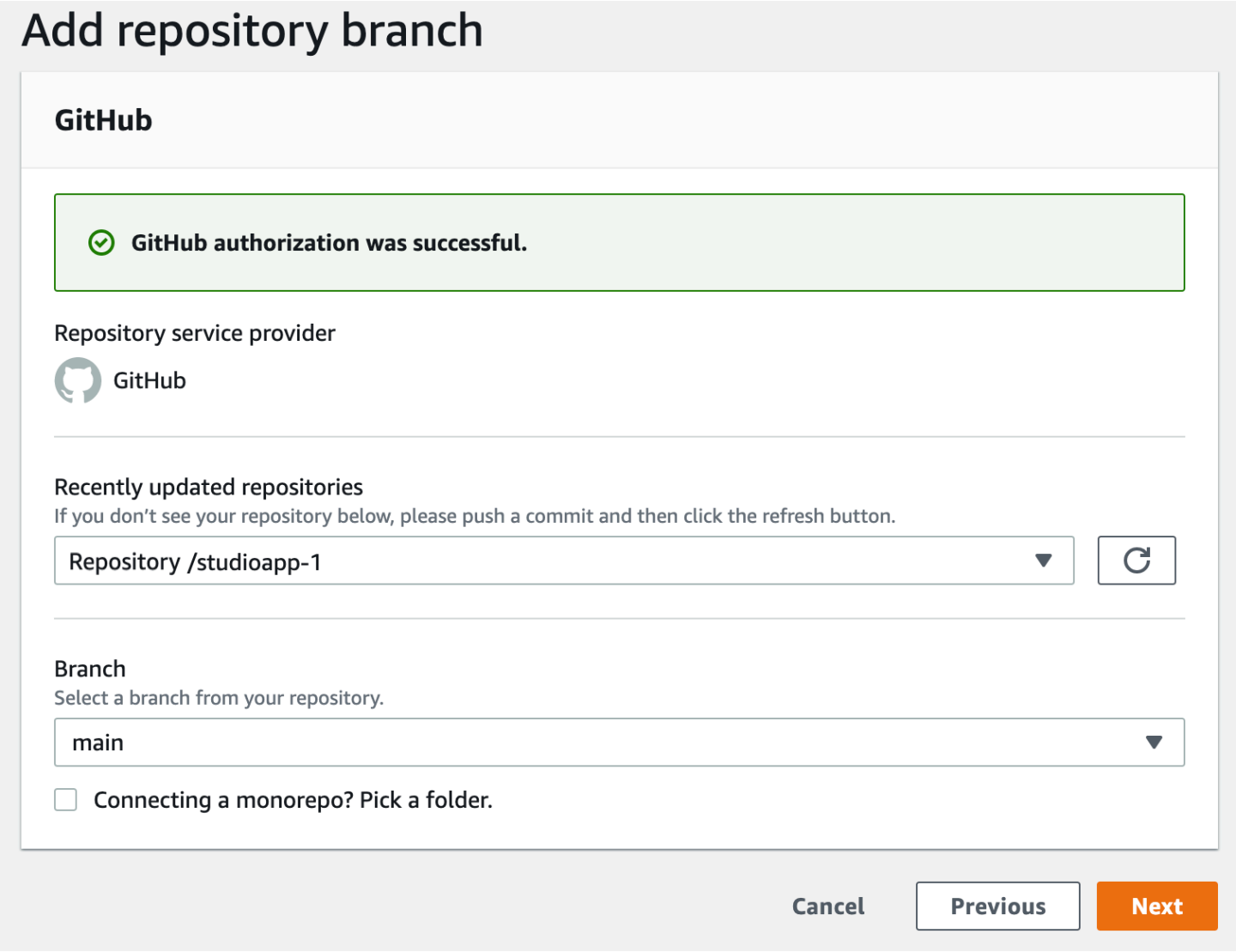
Conecte los repositorios de GitHub, Bitbucket, GitLab o AWS CodeCommit. También tiene la opción de cargar manualmente sus artefactos de compilación sin necesidad de conectar un repositorio de Git. Para obtener más información, consulte las implementaciones manuales.



Después de 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 de AWS. Amplify obtiene acceso a su repositorio utilizando claves de implementación instaladas solo en un repositorio específico.

Ahora, Amplify cuenta con la característica de aplicaciones de GitHub para autorizar el acceso de Amplify a repositorios de GitHub. Con la aplicación Amplify GitHub, los permisos están más ajustados, lo que te permite conceder acceso a Amplify solo a los repositorios que especifiques. Para obtener más información sobre la instalación y autorización de la aplicación de GitHub, consulte Configurar el acceso de Amplify a repositorios de GitHub.

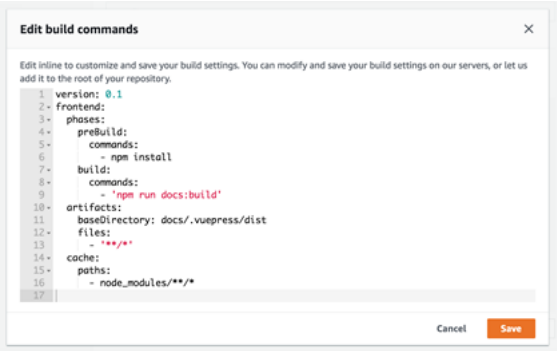
Después de conectarse al proveedor de servicios de repositorio, elija un repositorio y, a continuación, elija la ramificación correspondiente para compilar e implementar.



Paso 2: Confirmar la configuración de compilaciones para el frontend

Para la ramificación seleccionada, Amplify inspecciona el repositorio con el fin de detectar automáticamente la secuencia de comandos de compilación que se va a ejecutar.

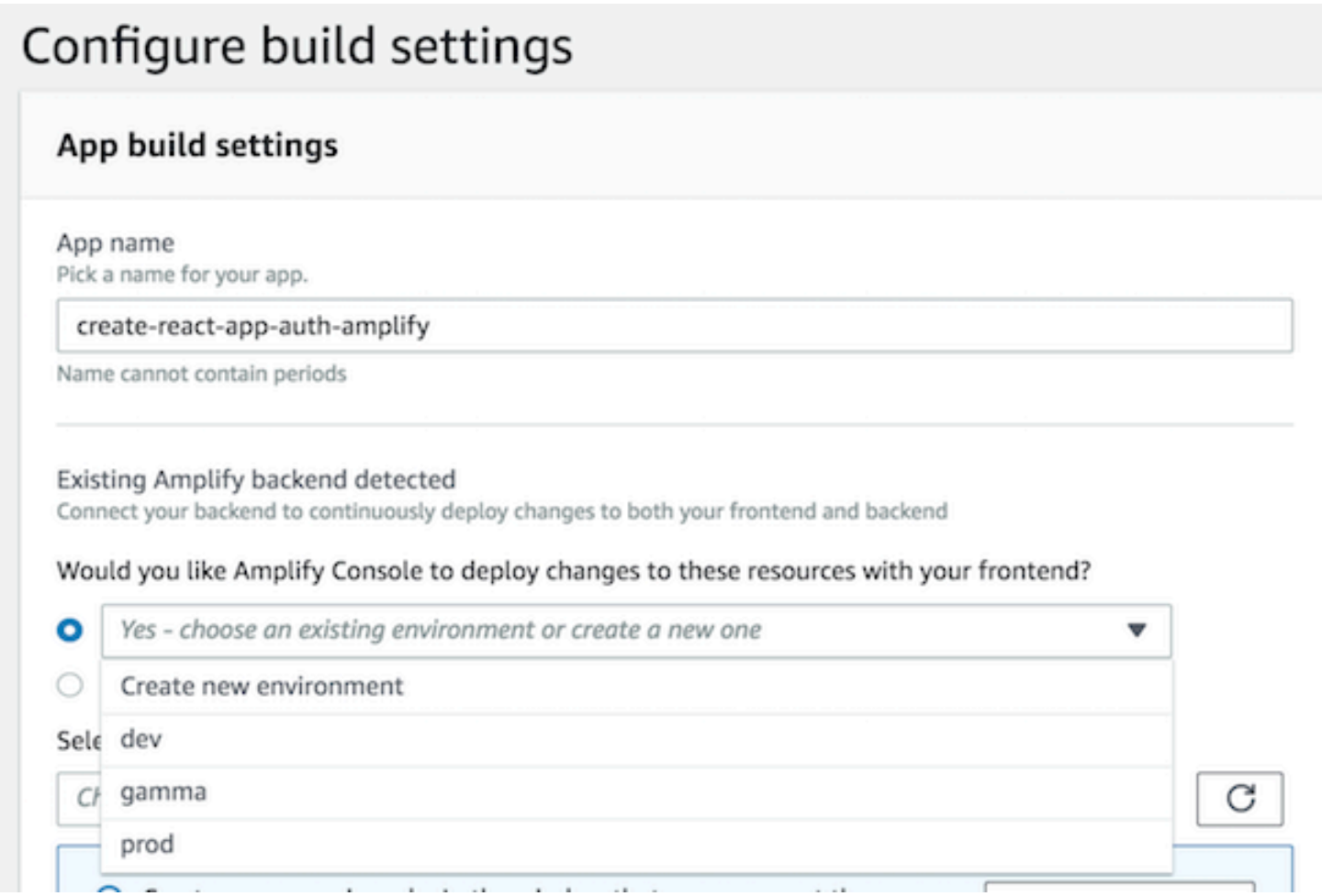
Importante: compruebe que el directorio de salida de compilación y comandos de compilación (es decir, artifacts > baseDirectory [artefactos > baseDirectory]) es correcto. Si tiene que modificar esta información, elija Editar para abrir el editor YML. Puede guardar la configuración de compilación en nuestros servidores, o bien puede descargar el archivo YML y añadirlo a la raíz de su repositorio (para monorepos, almacene el archivo YML en el directorio raíz de la aplicación).



Para obtener más información, consulte la sintaxis de las especificaciones de compilación de YAML.

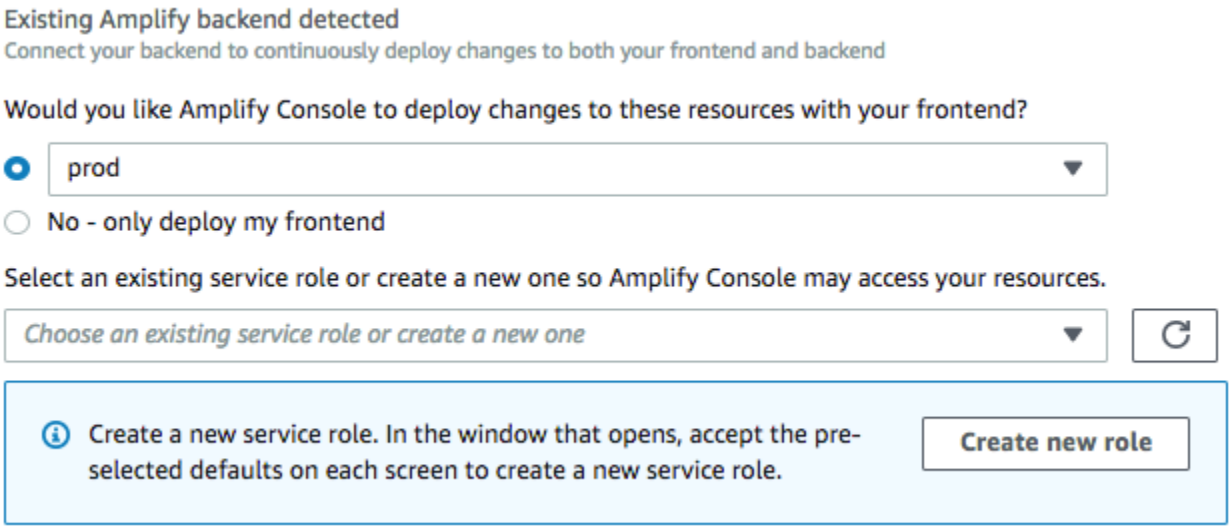
Paso 2b: Confirmar la configuración de compilaciones para el backend

Si ha conectado un repositorio suministrado por la versión 1.0+ de Amplify CLI (ejecute amplify -v para buscar la versión de CLI), Amplify Hosting implementará o actualizará automáticamente recursos de backend (cualquier recurso suministrado por Amplify CLI) en un solo flujo de trabajo con la compilación de frontend. Puede elegir entre apuntar un entorno de backend existente a su ramificación, o bien crear un entorno completamente nuevo. Para ver un tutorial detallado, consulte la introducción a las implementaciones de Fullstack.



Para implementar la funcionalidad de backend mediante Amplify CLI durante la compilación, cree o reutilice un rol de servicio de AWS Identity and Access Management (IAM). Los roles de IAM son una manera segura de conceder permisos a Amplify para actuar en los recursos de su cuenta. Para obtener instrucciones detalladas, consulte Adición de un rol de servicio.

Nota: Amplify CLI no se ejecutará sin un rol de servicio de IAM habilitado.



Paso 2c: Añadir variables de entorno (opcional)

Casi todas las aplicaciones necesitan obtener información de configuración en tiempo de ejecución. Estas configuraciones pueden ser los detalles de conexión de la base de datos, las claves de API u otros parámetros. Las variables de entorno proporcionan un medio para exponer estas configuraciones en el momento de la compilación.

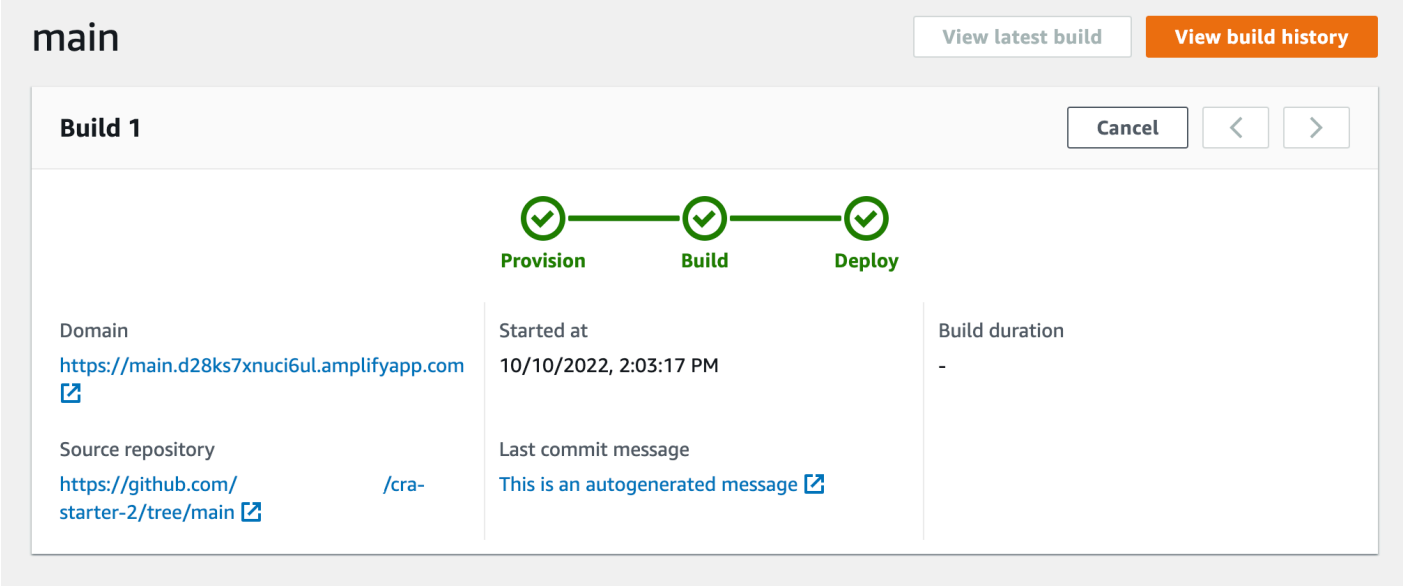
Paso 3: Guardar e implementar

Revise toda la configuración para garantizar que todo se haya configurado correctamente. Elija Guardar e implementar para implementar su aplicación web en una red de entrega de contenido (CDN) de AWS global. La compilación de frontend suele tardar de 1 a 2 minutos, pero puede variar en función del tamaño de la aplicación.

Acceda a la pantalla de registros de compilación seleccionando un indicador de progreso en la sección de ramificación. Una compilación tiene las siguientes etapas:

1. Aprovisionamiento: el entorno de compilación se configura mediante una imagen de Docker en un host con 4 vCPU, 7 GB de memoria. Cada compilación obtiene su propia instancia de host, lo que garantiza que todos los recursos se aíslen de forma segura. Se muestra el contenido del archivo de Docker para garantizar que la imagen predeterminada sea compatible con sus requisitos.
2. Compilación: la fase de compilación consta de tres etapas: configuración (clona el repositorio en contenedor), implementar backend (ejecuta la CLI de Amplify para implementar recursos de backend) y compilar front-end (compila los artefactos del front-end).

3. Implementación: cuando se completa el proceso de compilación, todos los artefactos se implementan en un entorno de alojamiento administrado por Amplify Hosting. Puedes ver tu aplicación en el dominio `amplifyapp.com`. Cada implementación es atómica: las implementaciones atómicas eliminan los períodos de mantenimiento asegurándose de que la aplicación web solo se actualice después de haber completado toda la implementación.



> **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, por sus siglas en inglés). Para obtener más información, consulte la página de configuración de cookies en la red de desarrolladores de Mozilla.

Siguientes pasos

- Añadir un dominio personalizado a su aplicación
- Administrar varios entornos
- Obtener una vista previa de las solicitudes de extracción antes de fusionarlas

Introducción a las implementaciones continuas de pila completa

Amplify Hosting permite a los desarrolladores crear aplicaciones con Amplify Framework para implementar de forma continua actualizaciones en su backend y frontend en cada confirmación de código. Con Amplify Hosting puede implementar backends sin servidor con API REST/GraphQL, autenticación, análisis y almacenamiento, creados con Amplify Studio, en la misma confirmación que su código de frontend.

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.

Temas

Requisitos previos

Antes de empezar este tutorial, tendrá que hacer lo siguiente:

- Registro en una Cuenta de AWS. Para comenzar, acceda a https://portal.aws.amazon.com/billing/signup#/start/email.
- Crea una cuenta con un proveedor de repositorios de git GitHub, como Bitbucket o AWS CodeCommit. GitLab
- 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 necesitas crear una nueva aplicación de interfaz para utilizarla en este ejemplo, puedes seguir las instrucciones para crear una aplicación de React que se encuentran en la documentación de creación de una aplicación de React.

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. En el GitHub caso de los repositorios, Amplify ahora usa 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. Configurar el acceso de Amplify a repositorios de 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.

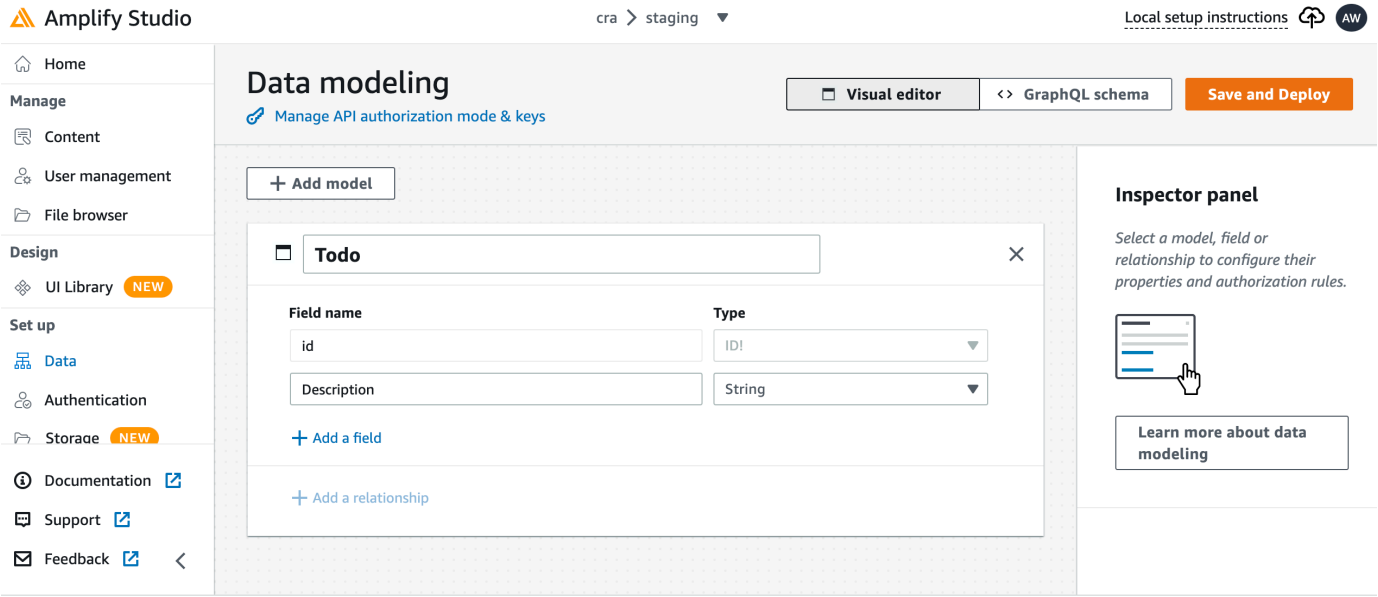
Siga estas instrucciones para crear una base de datos simple usando la interfaz del compilador visual de backend de Amplify Studio.

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.

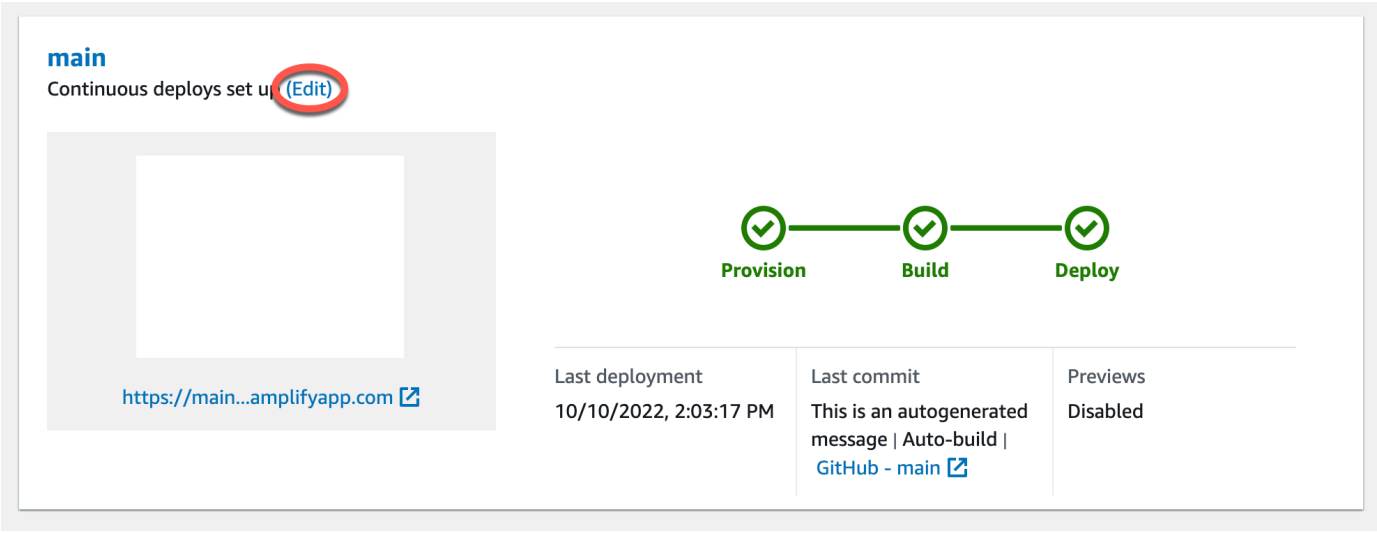
```
amplify pull --appId abcd1234 --envName staging
```

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.

   El CI/CD de pila completa está habilitado de forma predeterminada. Desmarque esta opción para desactivar el CI/CD de pila completa en este backend. Al desactivar el CI/CD de pila completa, la aplicación se ejecuta 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 ver instrucciones, consulte Adición de un rol de servicio.
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 acciones siguientes:

   - 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.

Siguientes pasos

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 backend de la aplicación. Para obtener más información y ver tutoriales, consulte la guía del usuario de Amplify Studio en la Documentación de Amplify Framework.

Implemente aplicaciones renderizadas del servidor con Amplify Hosting

Se puede utilizar 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 de 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.

Para obtener más información sobre cómo Amplify es compatible con SSR, revise los siguientes temas.

Temas

¿Qué es la renderización del servidor?

Amplify admite la implementación y el alojamiento de aplicaciones web estáticas creadas con marcos de aplicaciones de una sola página (SPA), como React, y aplicaciones creadas con un generador de sitios estáticos (SSG), como Gatsby. Las aplicaciones web estáticas consisten en una combinación de archivos, como HTML, CSS y JavaScript archivos, que se almacenan en una red de entrega de contenido (CDN). Cuando el navegador de un cliente realiza una solicitud al sitio web, el servidor devuelve una página al cliente con una respuesta HTTP y el navegador del cliente interpreta el contenido y se lo muestra al usuario.

Amplify también es compatible con aplicaciones web con representación del servidor (SSR). Cuando un cliente envía una solicitud a una página SSR, la HTML de la página se crea en el servidor en cada

solicitud. SSR permite a un desarrollador personalizar un sitio web por solicitud y usuario. Además, SSR puede mejorar el rendimiento y la optimización de motores de búsqueda (SEO) de un sitio web.

Compatibilidad de Amplify con marcos de SSR

Amplify Hosting es compatible con cualquier marco SSR JavaScript basado en un paquete de implementación que se ajusta al resultado de compilación que Amplify espera. Amplify Hosting proporciona una especificación de implementación que estandariza la estructura de archivos y directorios para la salida de la compilación de una aplicación para cualquier marco de SSR.

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 o un adaptador de marcos, puede desarrollar su propia solución para generar un paquete de implementación que se ajuste a la estructura de directorios prevista de Amplify Hosting.

Amplify Hosting admite los siguientes elementos primitivos: activos estáticos, computación, optimización de imágenes y reglas de enrutamiento. Puede utilizar estos elementos primitivos para implementar aplicaciones con una funcionalidad más completa. Para obtener información detallada sobre cada elemento primitivo, consulte Compatibilidad de los elementos primitivos de SSR con Amplify.

Puede elegir entre las siguientes situaciones para empezar a implementar una aplicación de SSR en Amplify.

Implementación de una aplicación de Next.js

> Amplify admite aplicaciones creadas con Next.js sin necesidad de un adaptador o configuración manual en la consola. Para obtener más información, consulte Amplificación de la compatibilidad con SSR de Next.js.

Implementación de una aplicación que utilice un adaptador de marcos

> Puede hacer referencia a cualquier adaptador de marcos de código abierto disponible para implementar la aplicación de SSR en Amplify Hosting. Para obtener más información, consulte Uso de un adaptador de marcos.

Hay un adaptador disponible para el marco Nuxt. Para obtener más información sobre cómo utilizar este adaptador, consulte la documentación de Nuxt.

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 Implementación de un servidor Express mediante el manifiesto de implementación.

Configuración de un script 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 Uso de la especificación de implementación de Amplify Hosting para configurar la salida de la compilación. Para ver un ejemplo, consulte Implementación de un servidor Express mediante el manifiesto de implementación.

Implementación de una aplicación de SSR en Amplify

Puede usar las instrucciones de este tema para implementar una aplicación creada con cualquier marco con un paquete 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 ver instrucciones, consulte Uso de un adaptador de marcos.

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, elija Nueva aplicación y, a continuación, Alojar aplicación web.
3. Selecciona tu proveedor GitHub, Bitbucket o AWS CodeCommit repositorio y GitLab, a continuación, selecciona Continuar.
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. En la página Configuración de compilación, Amplify detecta automáticamente las aplicaciones de 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. En la sección Implementación de la representación del servidor (SSR), seleccione Habilitar registros de aplicaciones de SSR.
6. La aplicación requiere un rol de servicio de IAM que Amplify asume para entregar los registros en su Cuenta de AWS. Puede permitir que Amplify Hosting cree automáticamente un rol de servicio o puede especificar un rol que haya creado usted.
   - Para permitir que Amplify cree automáticamente un rol y lo asocie a su aplicación
     - En la sección Rol de IAM, elija Crear y utilizar un nuevo rol de servicio.
   - Para adjuntar un rol de servicio que haya creado anteriormente
     a. En la sección Rol de IAM, elija Utilizar un rol de servicio existente.
     b. Elija el rol que desea utilizar de la lista.
7. Elija Siguiente.
8. En la página Revisar, elija Guardar e implementar.

Uso de un adaptador de marcos

Puede instalar y usar cualquier adaptador de compilación de marcos de 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 de SSR en Amplify.

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

Utilice la especificación de implementación de Amplify para configurar la salida de la compilación para un marco de SSR que quiera integrar con Amplify Hosting. Si es el autor de un marco, puede usar la especificación de implementación para comprender cómo estructurar la salida de compilación que Amplify espera. Si no utiliza un marco, puede desarrollar su propia solución para generar la salida de compilación que Amplify espera.

Especificación de implementación de Amplify Hosting

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.

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.

Cálculo

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.

El directorio .amplify-hosting/static

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`.

El directorio .amplify-hosting/compute

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.

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

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 Uso de redireccionamientos.

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
ruta	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 el estático y el

Clave	Tipo	Obligatorio	Descripción
			ImageOptimization primitivo. Los encabezados personalizados anulan el valor especificado. Para obtener más información sobre los encabezados de cliente de Amplify Hosting, consulte Encabezados personalizados. **Note** Este encabezado de 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.

Cálculo

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 (nombre)	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 ejecutará el código para el recurso de computación especific ado. El archivo debe

Clave	Tipo	Obligatorio	Descripción
			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).
formats	ImageFormat[]	Sí	Matriz de formatos de imagen de salida permitidos.
minimumCacheTTL	Número	Sí	Duración del almacenamiento en caché en segundos

Clave	Tipo	Obligatorio	Descripción
			para las imágenes optimizadas.
dangerouslyAllowSVG	Booleano	Sí	Permite URL de imágenes de entrada en formato SVG. De forma predeterm inada, está deshabili tada 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
protocol	Cadena	No	Protocolo del patrón remoto permitido. Los valores válidos son `http` o `https`.
hostname	Cadena	Sí	Nombre de host del patrón remoto permitido. Puede especificar un literal o un comodín. Un carácter único

Clave	Tipo	Obligatorio	Descripción
			“*” coincide con un único subdominio. Un carácter doble “**” coincide con cualquier cantidad de subdominios. Amplify no permite caracteres comodín generales cuando solo se especifica “**”.
port	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 (nombre)	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.

En el caso de los marcos de SSR como Nuxt.js, el destino de la regla de método catch-all tiene que ser el elemento primitivo de computación. 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 una aplicación Nuxt.js tiene una página en `/blog/[slug]` donde `[slug]` es 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 distribuye activos estáticos desde la ruta `/_nuxt`. 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

En el caso de los marcos de SSR como Nuxt.js, donde se especifica una ruta de método catch-all para el destino del elemento primitivo de computación, los autores de los marcos podrían considerar la posibilidad de especificar una reserva estática para la ruta de método catch-all a fin de resolver el problema del 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.
- framework: ajústela para especificar la configuración del servidor Express.
- imageSettings: esta sección es opcional para un servidor Express, 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.
- computeResources: utilice esta sección para especificar el tiempo de ejecución y el punto de entrada del servidor Express.

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.

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 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.

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 Especificación de implementación de Amplify Hosting.

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 con direcciones URL remotas.

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`
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`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
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`
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`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
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`
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`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
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`
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`

Parámetros de consulta	Tipo	Obligatorio	Descripción	Ejemplo
blur	Número	No	Aplica un desenfoque gaussiano del sigma especific ado. Los valores válidos se encuentran entre 0,3 y 1000.	`?blur=20`
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.

Almacenamiento en caché

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.

El valor de `minimumCacheTTL` especificado en la configuración de la optimización de imágenes define el límite inferior de la directiva cache-control max-age. 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.

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

Cuando Amplify compila e implementa una aplicación de computación de Next.js, utiliza la versión de tiempo de ejecución de Node.js que coincide con la versión principal de Node.js que se utilizó para compilar la aplicación.

Puede especificar la versión de Node.js que se utilizará en la característica de anulación de paquetes en directo de la consola de Amplify. Para obtener más información sobre cómo configurar las actualizaciones de paquetes en directo, consulte Actualizaciones de paquetes en directo. También puede especificar la versión de Node.js mediante otros mecanismos, como comandos nvm. Si no especifica ninguna versión, Amplify usará de forma predeterminada la versión actual que utiliza el contenedor de compilación de Amplify.

Resolución de problemas de las implementaciones de SSR

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

- Usa un adaptador de marcos
- Las rutas de la API de Edge permiten que la compilación de Next.js falle
- La regeneración estática incremental bajo demanda no funciona para su aplicación
- El resultado de compilación de tu aplicación supera el tamaño máximo permitido
- La compilación falla debido a un error de memoria insuficiente
- El tamaño de respuesta HTTP es demasiado grande

Usa un adaptador de marcos

Si tiene problemas para implementar una aplicación de SSR que usa un adaptador de marcos, consulte Compatibilidad de Amplify con marcos de SSR.

Las rutas de la API de Edge permiten que la compilación de Next.js falle

Actualmente, Amplify no admite las rutas de la API de Edge de Next.js. Debe utilizar API y middleware que no sean periféricos cuando aloje su aplicación con Amplify.

La regeneración estática incremental bajo demanda no funciona para su 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 tu aplicación supera el tamaño máximo permitido

Actualmente, el tamaño máximo de salida de compilación que Amplify admite para aplicaciones SSR es de 220 MB. Si recibes un mensaje de error que indica que el tamaño del resultado de compilación de tu aplicación supera el tamaño máximo permitido, debes tomar medidas para reducirlo.

Para reducir el tamaño del resultado de compilación de una aplicación, puedes inspeccionar los artefactos de compilación de la aplicación e identificar cualquier dependencia importante que desees actualizar o eliminar. En primer lugar, descarga los artefactos de compilación a tu ordenador local. A continuación, compruebe el tamaño de los directorios. Por ejemplo, el `node_modules` directorio puede contener archivos binarios como, por ejemplo, los archivos de tiempo de ejecución del servidor Next.js `@swc` y a los `@esbuild` que hacen referencia. Como estos binarios no son necesarios en tiempo de ejecución, puedes 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).

Para descargar e inspeccionar el resultado de la compilación de una aplicación de Next.js

1. Abre una ventana de terminal y ejecuta el siguiente comando. Cambia el identificador de la aplicación, el nombre de la sucursal y el identificador del trabajo por tu propia información. Para la identificación del trabajo, usa el número de compilación de la compilación fallida que estás investigando.

```
aws amplify get-job --app-id abcd1234 --branch-name main --job-id 2
```

2. En la salida del terminal, localiza la URL del artefacto prefirmado en la `stepName: "BUILD"` sección`job`,`steps`,. La URL aparece resaltada en rojo en el siguiente resultado de ejemplo.

```
"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. Copia y pega la URL en una ventana del navegador. Se descarga un `artifacts.zip` archivo en el ordenador local. Este es el resultado de la compilación.

4. Ejecuta el comando du disk usage para inspeccionar el tamaño de los directorios. El siguiente comando de ejemplo devuelve el tamaño de los `static` directorios `compute` y.

```
du -csh compute static
```

A continuación se muestra un ejemplo del resultado con información sobre el tamaño de los `static` directorios `compute` y.

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

5. Abra el `compute` directorio y localice la `node_modules` carpeta. Revise sus dependencias para ver si hay archivos que pueda actualizar o eliminar para reducir el tamaño de la carpeta.
6. Si tu aplicación incluye archivos binarios que no son necesarios en tiempo de ejecución, elimínalos después de la compilación añadiendo los siguientes comandos a la sección de compilación del archivo de tu aplicación. `amplify.yml`

```
- 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 `amplify.yml` archivo 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
```

La 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 las 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 es demasiado grande

Actualmente, el tamaño máximo de respuesta que Amplify admite para las aplicaciones Next.js 12 y 13 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.

Amplificación de la compatibilidad con SSR de Next.js

Amplify admite la implementación y el alojamiento de aplicaciones web renderizadas en el servidor (SSR) creadas únicamente con Next.js. Next.js es un marco de React para desarrollar SPA con JavaScript. Puede implementar aplicaciones creadas con Next.js 13 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

La computación de Amplify Hosting administra completamente la representación del servidor (SSR) para las aplicaciones creadas con Next.js 12 y 13. Si ha implementado una aplicación de

Next.js en Amplify antes del lanzamiento del procesamiento de Amplify Hosting, su aplicación utiliza el anterior proveedor clásico de SSR de Amplify (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)
- 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
- Detección automática de configuración regional internacionalizada (i18n)
- Streaming de Next.js
- Ejecución de middleware en activos estáticos e imágenes optimizadas

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.

Precios de las aplicaciones SSR de Next.js

Al implementar su aplicación SSR Next.js 12 o posterior, 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 deAWS Amplify.

Implementación de una aplicación SSR de Next.js con Amplify

De forma predeterminada, Amplify implementa nuevas aplicaciones de SSR mediante el servicio de computación de Amplify Hosting compatible con Next.js 12 y 13. 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 ver 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.

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, elija Nueva aplicación y, a continuación, Alojar aplicación web.
3. Selecciona tu proveedor GitHub, Bitbucket o AWS CodeCommit repositorio y GitLab, a continuación, selecciona Continuar.
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

        - En la sección Rol de IAM, elija Crear y utilizar un nuevo rol de servicio.
    - Para adjuntar un rol de servicio que haya creado anteriormente

        a. En la sección Rol de IAM, elija Utilizar un rol de servicio existente.

        b. Elija el rol que desea utilizar de la lista.
6. Elija 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 detectar si la aplicación es SSR o SSG.

A continuación, se muestra un ejemplo del script de compilación de una aplicación SSR de Next.js. El script de compilación `"next build"` indica que la aplicación es compatible con las páginas SSG y SSR.

```
"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. 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

Después de inspeccionar el archivo `package.json` de su aplicación para determinar si está implementando una aplicación SSG o SSR, 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 Configuración de ajustes de compilación.

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/**/*
```

Si Amplify detecta que está implementando una aplicación SSG, 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.

A continuación se muestra un ejemplo de la configuración de compilación de una aplicación donde `baseDirectory` se configura en `out`. Esto indica que los artefactos de compilación son para una aplicación de Next.js 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/**/*
```

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 procesamiento de Amplify Hosting es compatible con la versión 13 de Next.js.

La consola de Amplify detecta las aplicaciones de su cuenta que se han implementado antes del lanzamiento del servicio de computación de Amplify Hosting con total compatibilidad con Next.js 12 y 13. 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.

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 13.

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 en la consola de Amplify, primero debe actualizar el archivo package.json de la aplicación para utilizar la versión 12 o 13 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 de SSR de Next.js 12 o 13 se establece 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 de SSG a SSR, actualice la aplicación para que utilice la versión 12 o 13 de Next.js y agregue la funcionalidad de 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.

Actualice 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 o 13 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
```

Añada 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. Elija 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 Configuración de ajustes de compilación.

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/**/*
```

Actualice el 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 va a implementar una aplicación SSR en un monorepo y quiere que Next.js pueda acceder a variables de entorno específicas, debe anteponer la raíz de la aplicación al archivo. `.env.production` El siguiente ejemplo de especificación de compilación para una aplicación Next.js dentro de un monorepo de Nx demuestra cómo agregar 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 muestran cómo tomar variables específicas del entorno de compilación y agregarlas al `.env.production` archivo de una aplicación en un monorepo con la raíz de la 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. Tenga en cuenta que las aplicaciones pnpm y Turborepo requieren una configuración adicional. Para obtener más información, consulte 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.

Amazon CloudWatch Logs para aplicaciones SSR

Amplify envía información sobre su tiempo de ejecución de Next.js a Amazon CloudWatch Logs en su. Cuenta de AWSAl 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 Adición de un rol de servicio.

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. Sin los permisos mínimos requeridos, aparecerá un error al intentar 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 Adición de un rol de servicio. 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:ListCertificates
acm:RequestCertificate
cloudfront:CreateCloudFrontOriginAccessIdentity
cloudfront:CreateDistribution
cloudfront:CreateInvalidation
cloudfront:GetDistribution
cloudfront:GetDistributionConfig
cloudfront:ListCloudFrontOriginAccessIdentities
cloudfront:ListDistributions
cloudfront:ListDistributionsByLambdaFunction
cloudfront:ListDistributionsByWebACLId
cloudfront:ListFieldLevelEncryptionConfigs
cloudfront:ListFieldLevelEncryptionProfiles
cloudfront:ListInvalidations
cloudfront:ListPublicKeys
cloudfront:ListStreamingDistributions
cloudfront:UpdateDistribution
cloudfront:TagResource
cloudfront:UntagResource
cloudfront:ListTagsForResource
cloudfront:DeleteDistribution
iam:AttachRolePolicy
iam:CreateRole
iam:CreateServiceLinkedRole
iam:GetRole
iam:PutRolePolicy
iam:PassRole
iam:UpdateAssumeRolePolicy
iam:DeleteRolePolicy
lambda:CreateFunction
lambda:EnableReplication
lambda:DeleteFunction
lambda:GetFunction
lambda:GetFunctionConfiguration
lambda:PublishVersion
lambda:UpdateFunctionCode
lambda:UpdateFunctionConfiguration
lambda:ListTags
lambda:TagResource
lambda:UntagResource
lambda:ListEventSourceMappings
lambda:CreateEventSourceMapping
```

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

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 se ha anulado
- Recibe un error 404 después de implementar su sitio SSR
- A tu aplicación le falta la regla de reescritura para las distribuciones de SSR CloudFront
- La aplicación es demasiado grande para implementarla
- La compilación falla debido a un error de memoria insuficiente
- Su aplicación tiene las ramificaciones SSR y SSG
- Su aplicación almacena los archivos estáticos en una carpeta con una ruta reservada
- Su aplicación ha alcanzado un CloudFront límite
- Las variables de entorno no se transfieren a las funciones de Lambda
- Las funciones de Lambda@Edge se crean en la región este de EE. UU. (Norte de Virginia)
- Su aplicación Next.js utiliza características no compatibles

- Las imágenes de su aplicación Next.js no se cargan
- Regiones no admitidas

El directorio de salida se ha anulado

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 Configuración de ajustes de compilación.

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/**/*
```

Recibe un error 404 después de implementar su sitio SSR

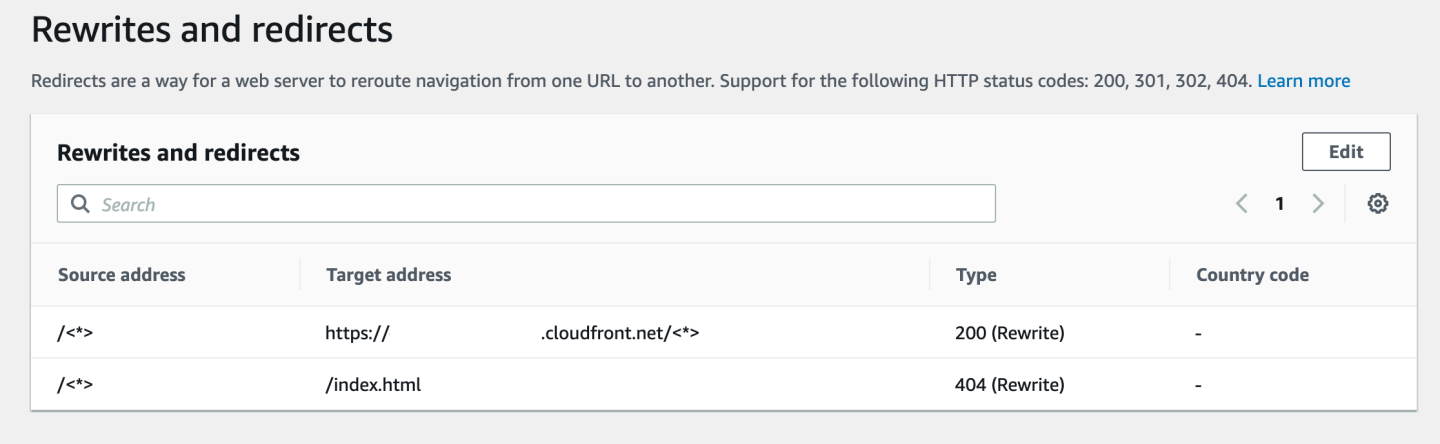
Si recibe un error 404 después de implementar su sitio, el problema podría deberse a que se ha anulado el directorio de salida. Para comprobar el archivo `next.config.js` y comprobar

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 se ha anulado.

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

Cuando implementas una aplicación de SSR, Amplify crea una regla de reescritura para CloudFront tus distribuciones de 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.



La 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
      },
}
```

La 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 las 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.

Su 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.

Su 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.

Su 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.

Su 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 ver instrucciones, consulte Migración de una aplicación SSR de Next.js 11 al procesamiento de Amplify Hosting.

Las imágenes de su 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.

Configuración de dominios personalizados

Puede conectar una aplicación que haya implementado con Amplify Hosting a un dominio personalizado. Cuando utilizas Amplify para implementar tu aplicación web, Amplify la aloja en el `amplifyapp.com` dominio 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`.

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 dominios acreditados por terceros, consulte el Directorio de registradores acreditados en el sitio web de la ICANN.

Cuando configuras tu dominio personalizado, puedes usar el certificado administrado predeterminado que Amplify te proporciona o puedes usar tu propio certificado personalizado. Puedes cambiar el certificado en uso para el dominio en cualquier momento. Para obtener información detallada sobre la administración de certificados, consulteUso de certificados SSL/TLS.

Antes de continuar con la configuración de un dominio personalizado, compruebe que cumple los siguientes requisitos previos.

- Tienes un nombre de dominio registrado.
- Tienes un certificado emitido o importado a AWS Certificate Manager.
- Has implementado tu aplicación en Amplify Hosting.

  Para obtener más información sobre cómo llevar a cabo este paso, consulte Introducción al código existente.
- Tiene un conocimiento básico de los dominios y la terminología de 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 SSL/TLS
- Añadir un dominio personalizado administrado en Amazon Route 53
- Añadir un dominio personalizado administrado por un proveedor de DNS externo
- Agregue un dominio personalizado administrado por GoDaddy
- Añada un dominio personalizado gestionado por Google Domains
- Actualiza el certificado SSL/TLS de un dominio
- Administración de subdominios
- Subdominios comodín
- Configure 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 en Amplify Hosting

Cuando agrega un dominio personalizado con Amplify Hosting, debe llevar a cabo una serie de pasos para ver su aplicación en su dominio personalizado. La siguiente lista describe cada paso del proceso de configuración del dominio.

Creación de SSL/TLS

Si utiliza un certificado administrado, AWS Amplify emita un certificado SSL/TLS para configurar un dominio personalizado seguro.

Configuración y verificación de SSL/TLS

Antes de emitir un certificado gestionado, Amplify comprueba que eres 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, debe agregar manualmente el registro de verificación de DNS proporcionado en la consola de Amplify a su dominio con un proveedor de DNS externo.

Si utilizas un certificado personalizado, eres responsable de validar la propiedad del dominio.

Activación del dominio

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

Uso de certificados SSL/TLS

Un certificado SSL/TLS es un documento digital que permite a los navegadores web identificar y establecer conexiones de red cifradas con sitios web mediante el protocolo seguro SSL/TLS. Cuando configuras tu dominio personalizado, puedes usar el certificado administrado predeterminado que Amplify te proporciona o puedes usar tu propio certificado personalizado.

Con un certificado administrado, Amplify emite un certificado SSL/TLS para todos los dominios conectados a tu aplicación, de modo que todo el tráfico esté protegido 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. Amplify no puede renovar el certificado si el registro de verificación de CNAME se ha modificado o eliminado en la configuración de DNS con tu proveedor de dominio. Deberá eliminar y volver a añadir el dominio en la consola de Amplify.

Para usar un certificado personalizado, debes obtener un certificado de la autoridad certificadora externa que elijas. A continuación, importe el certificado 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 sus recursos internos conectados Servicios de AWS y sus recursos internos conectados. Asegúrese de solicitar o importar el certificado en la región EE.UU. Este (Norte de Virginia) (us-east-1).

Asegúrese de que su certificado personalizado cubra todos los subdominios que planea agregar. Puedes usar un comodín al principio del nombre de dominio para incluir varios subdominios. Por ejemplo, si tu dominio lo es`example.com`, puedes 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 cómo importar certificados a AWS Certificate Manager, consulte Importación de certificados a AWS Certificate Manager en la Guía del AWS Certificate Manager usuario.

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 gestiona las renovaciones automáticamente. Usted es responsable de renovar sus certificados personalizados y volver a importarlos.

Puede cambiar el certificado en uso para un dominio en cualquier momento. Por ejemplo, puede cambiar del certificado administrado predeterminado a un certificado personalizado o cambiar de un certificado personalizado a un certificado administrado. Además, puede cambiar el certificado personalizado en uso por un certificado personalizado diferente. Para obtener instrucciones sobre cómo actualizar los certificados, consulte Actualizar el certificado SSL/TLS de un dominio.

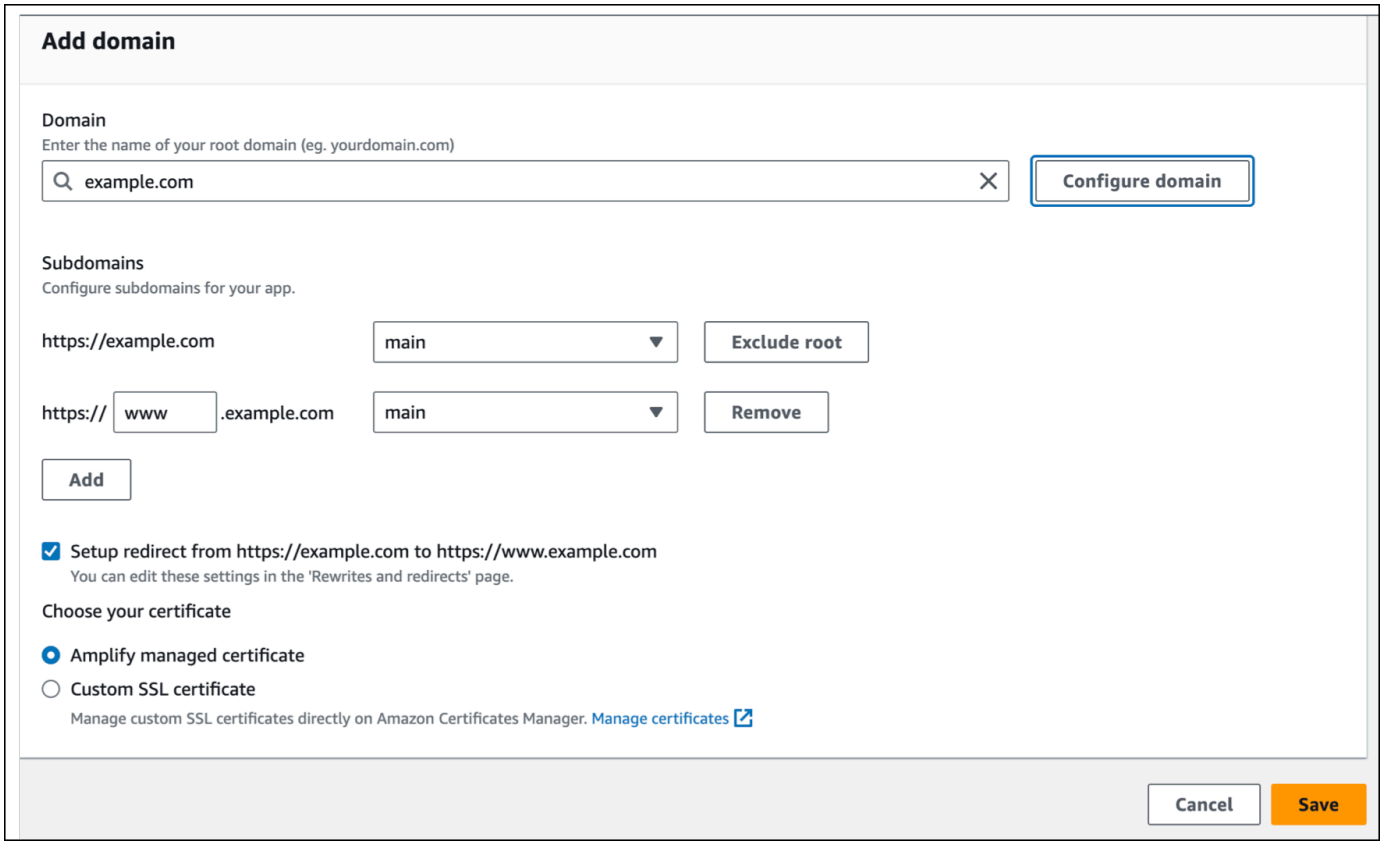
Añadir un dominio personalizado administrado en Amazon Route 53

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 Configuración de la aplicación, Administración de dominio.
4. En la página Gestión de dominio, elija Agregar dominio.
5. En Dominio, introduce tu dominio raíz. Por ejemplo, si ha registrado el nombre de dominio https://ejemplo.com, indique ejemplo.com como dominio.

   A medida que comience a escribir, aparecen en la lista los dominios raíz que ya administra en Route 53. Puedes elegir el dominio que quieres usar de la lista. 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, selecciona Configurar dominio.
7. Amplify crea automáticamente dos entradas de subdominio para su dominio de forma predeterminada. Por ejemplo, si su nombre de dominio es ejemplo.com, verá los subdominios https://www.ejemplo.com y https://ejemplo.com con una redirección configurada desde el dominio raíz al subdominio www.

(Opcional) Puede modificar la configuración predeterminada si desea añadir solo subdominios. Para cambiar la configuración predeterminada, selecciona Reescrituras y redirecciones en el panel de navegación y, a continuación, configura tu dominio.



8. Elige el certificado SSL/TLS que deseas usar. 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 gestionado por Amplify predeterminado.
     - Elija Amplify managed certificate.
   - Utilice un certificado personalizado de terceros.
     a. Elige un certificado SSL personalizado.
     b. Seleccione el certificado que desee utilizar de la lista.
9. Seleccione Guardar.

> **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.

Añadir un dominio personalizado administrado por un proveedor de DNS externo

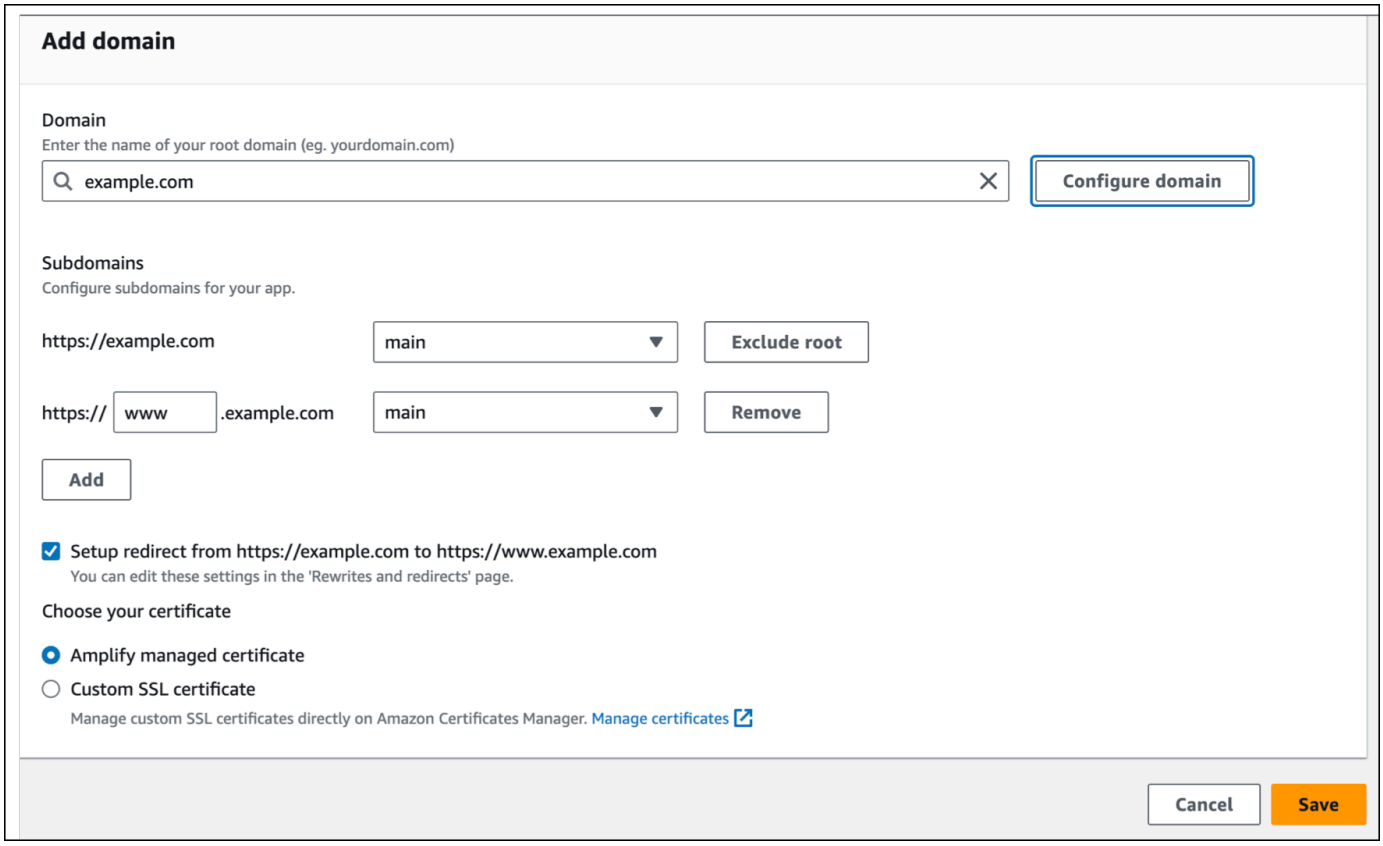
Si no usa Amazon Route 53 para administrar su dominio, puede añadir un dominio personalizado gestionado por un proveedor de DNS externo a su aplicación implementada con Amplify.

Si utilizas GoDaddy Google Domains, consulta the section called “Agregue un dominio personalizado administrado por GoDaddy” o the section called “Añada un dominio personalizado gestionado por Google Domains” para conocer los procedimientos específicos de estos proveedores.

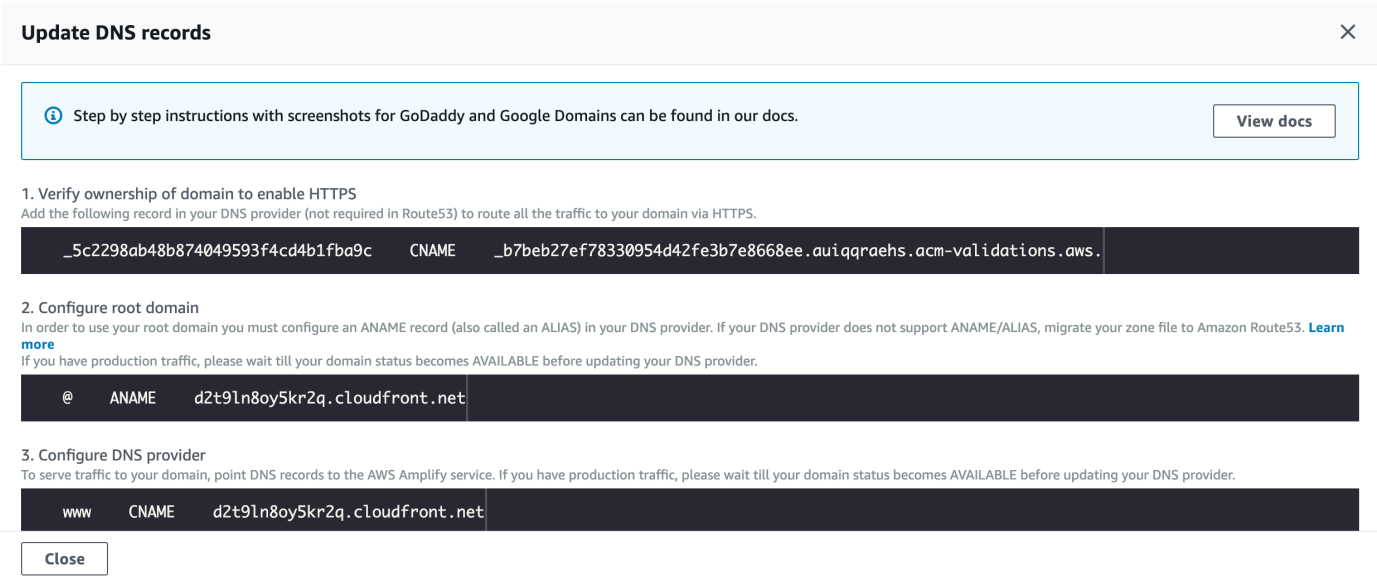
Para añadir un dominio personalizado administrado por un proveedor de DNS externo

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 Configuración de la aplicación, Administración de dominio.
4. En la página Gestión de dominio, elija Agregar dominio.
5. En Dominio, introduzca el nombre de su dominio raíz y, a continuación, elija Configurar dominio. Por ejemplo, si el nombre de su dominio es https://ejemplo.com, introduzca **`example.com`**.
6. Amplify crea automáticamente dos entradas de subdominio para su dominio de forma predeterminada. Por ejemplo, si su nombre de dominio es ejemplo.com, verá los subdominios https://www.ejemplo.com y https://ejemplo.com con una redirección configurada desde el dominio raíz al subdominio www.

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



7. Elija 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 gestionado por Amplify predeterminado.
     - Elija Amplify managed certificate.
   - Utilice un certificado personalizado de terceros.
     a. Elige un certificado SSL personalizado.
     b. Seleccione el certificado que desee utilizar de la lista.
8. Seleccione Guardar.
9. En el menú Acciones, elija Ver registros de DNS. En el siguiente paso, utilizarás estos registros DNS para actualizar tus registros DNS con tu proveedor de dominios externo.



10. Realice una de las acciones siguientes:

    - Si lo estás usando GoDaddy, ve aAgregue un dominio personalizado administrado por GoDaddy.
    - Si usa Google Domains, acceda a Añada un dominio personalizado gestionado por Google Domains.
    - Si usa otro proveedor de DNS externo, continúe con el siguiente paso de este procedimiento.

11. 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.
12. Configura un CNAME para que apunte al servidor AWS de validación. Por ejemplo, si el servidor de validación es _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, introduzca _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws. 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.

13. Configure el registro de CNAME (por ejemplo, https://*.ejemplo.com) para apuntar sus subdominios al dominio 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.
14. Configure el registro ANAME/ALIAS para que apunte al dominio raíz de su **`amplifyapp`** dominio (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.

Agregue un dominio personalizado administrado por GoDaddy

Para añadir un dominio personalizado gestionado por GoDaddy

1. Siga los pasos del uno al nueve del procedimientothe section called “Añadir un dominio personalizado administrado por un proveedor de DNS externo”.
2. Inicie sesión en su GoDaddy cuenta.

3. En tu lista de dominios, busca el dominio que deseas añadir y selecciona 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 Registros de DNS, selecciona Añadir 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 xxxxxxxxxxxxx.cloudfront.net como dominio de su aplicación, introduzca xxxxxxxxxxxxx.cloudfront.net en Valor.

   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, introduzca solo _c3e2d7eaf1e656b73f46cd6980fdc0e en Nombre.

   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.

   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 registros ANAME/ALIAS. Para los proveedores de DNS que no admiten ANAME/ALIAS, le recomendamos encarecidamente migrar 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, busca el menú en la parte superior de la página y selecciona Reenvío.
   b. En la sección Dominio, selecciona Añadir reenvío.
   c. Selecciona http://y, a continuación, introduce el nombre del subdominio al que quieres redirigirte (por ejemplo, www.example.com) como URL de destino.
   d. En Tipo de redirección, elija Temporal (302).
   e. Elija Guardar.

Añada un dominio personalizado gestionado por Google Domains

Para añadir un dominio personalizado gestionado por Google Domains

1. Siga los pasos del uno al nueve del procedimiento para agregar un dominio personalizado administrado por un proveedor de DNS externo.
2. Inicie sesión en su cuenta de https://domains.google.com y, en el panel de navegación izquierdo, elija Mis dominios.
3. En la lista de dominios, busque el dominio que desea añadir y eligir Gestionar.
4. En el panel de navegación izquierdo, elija DNS. Google mostrará los Registros de recursos de su dominio. Deberá añadir dos nuevos registros CNAME.
5. Cree el primer registro CNAME para apuntar todos los subdominios al dominio de Amplify como se indica a continuación:

   a. En Nombre de host, introduzca solo el nombre del subdominio. Por ejemplo, si su subdominio es www.ejemplo.com, introduzca www en Nombre de host.
   b. En Tipo, elija CNAME.

c. En Datos, introduzca el valor que encontrará en la consola de Amplify.

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

6. Cree 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 Nombre de host, 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, introduzca solo _c3e2d7eaf1e656b73f46cd6980fdc0e en Nombre de host.

   b. En Tipo, elija CNAME.

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

      Por ejemplo, si el servidor de validación es _cf1z2npwt9vzexample93c1j4xzc92wl.2te3iym6kzr.acm-validations.aws., introduzca _cf1z2npwt9vzexample93c1j4xzc92wl.2te3iym6kzr.acm-validations.aws. en Datos.

7. Elija 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.

8. La compatibilidad de Google Domains con los registros ANAME/ALIAS está en versión preliminar. Para proveedores de DNS que no admiten ANAME/ALIAS, le recomendamos encarecidamente migrar su DNS a Amazon Route 53. Para obtener más información, consulte Configuración de Amazon Route 53 como servicio DNS. Si desea mantener Google Domains como su proveedor y actualizar el dominio raíz, configure una redirección de subdominio. Busque la página del Sitio web de su dominio de Google. A continuación, elija Redireccionar dominio y configure la redirección en la página de Redirección web.

> **Note**
>
> Las actualizaciones de la configuración de DNS de un dominio de Google pueden tardar hasta 48 horas en surtir efecto. Si necesita ayuda para resolver los errores que puedan surgir, consulte Solución de problemas de dominios personalizados.

Actualiza el certificado SSL/TLS de un dominio

Puede cambiar el certificado SSL/TLS que se utiliza para un dominio en cualquier momento. Por ejemplo, puede pasar de usar un certificado administrado a usar un certificado personalizado. También puede cambiar el certificado personalizado que se utiliza para el dominio. Para obtener más información sobre los certificados, consulte Uso de certificados SSL/TLS.

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

Para actualizar el certificado de un dominio

1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elige la aplicación que deseas actualizar.
3. En el panel de navegación, elija Configuración de la aplicación, luego Administración de dominio.
4. En la página de administración de dominios, selecciona Administrar dominio.
5. En la página de detalles de tu dominio, busca la sección Elige tu certificado. El procedimiento para actualizar el certificado varía en función del tipo de cambio que desee realizar.
   - Para cambiar de un certificado personalizado al certificado gestionado por Amplify predeterminado
     - Elija Amplify managed certificate.
   - Para cambiar de un certificado gestionado a un certificado personalizado
     a. Elija un certificado SSL personalizado.
     b. Seleccione el certificado que desee utilizar de la lista.
   - Para cambiar un certificado personalizado por un certificado personalizado diferente
     - En el caso del certificado SSL personalizado, seleccione de la lista el nuevo certificado que desee utilizar.

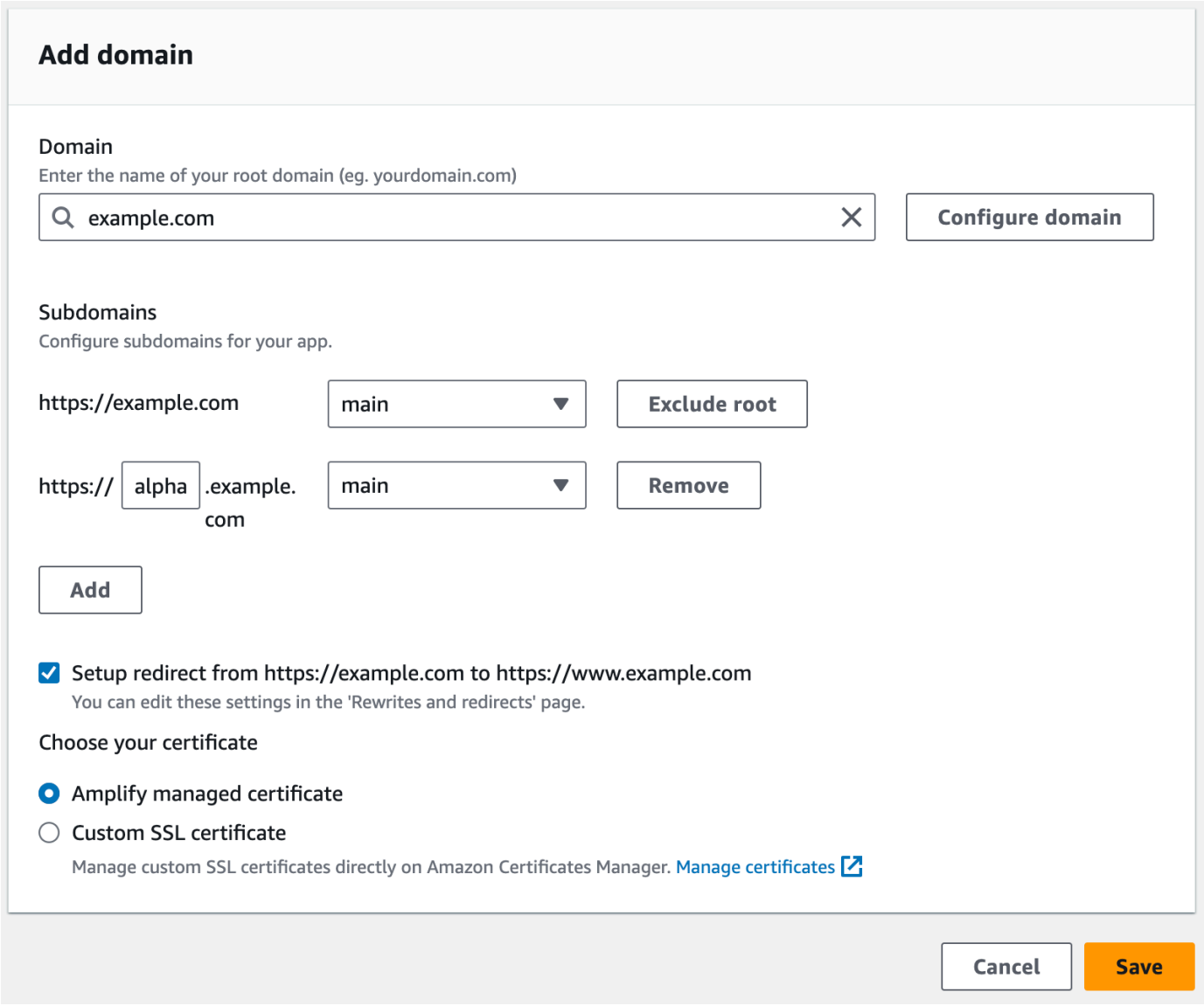
6. Seleccione Actualizar. Los detalles del estado del dominio indicarán que Amplify ha iniciado el proceso de creación de SSL para un certificado gestionado o el proceso de configuración de un certificado personalizado.

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

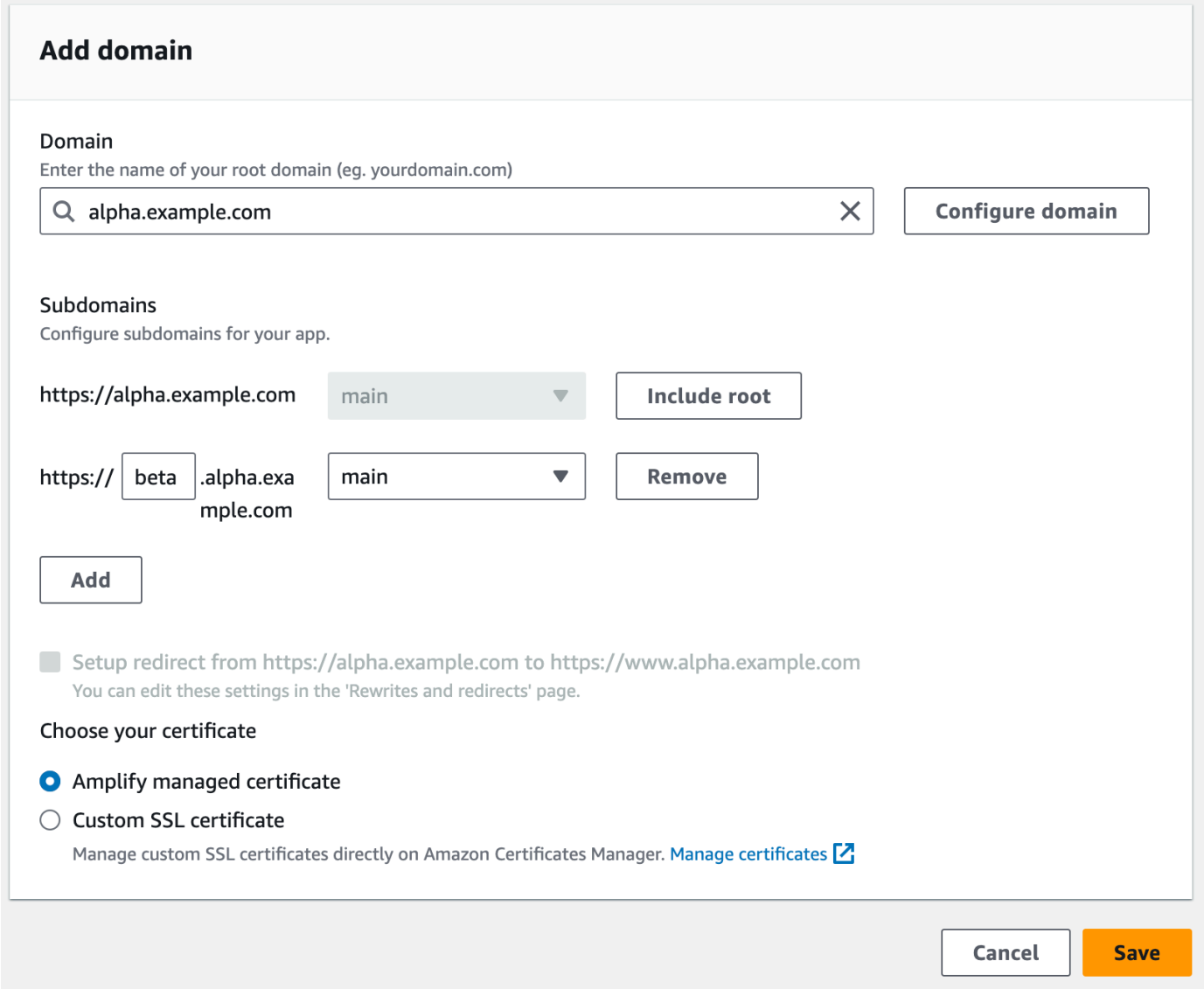
1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación a la que desea añadir un subdominio.
3. En el panel de navegación, elija Configuración de la aplicación, luego Administración de dominio.
4. En la página Gestión de dominio, elija Agregar dominio.
5. En Dominio, introduzca el nombre de su dominio raíz y, a continuación, elija Configurar dominio. Por ejemplo, si ha registrado el nombre de dominio https://ejemplo.com, indique ejemplo.com como dominio.
6. Elija Excluir raíz y modifique el nombre del subdominio. Por ejemplo, si el dominio es example.com, puedes modificarlo para que solo añada el subdominio alfa, como se muestra en la siguiente captura de pantalla.



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 Configuración de la aplicación, luego Administración de dominio.
4. En la página Gestión de dominio, elija Agregar dominio.
5. En Dominio, introduzca el nombre de un dominio con un subdominio, elija Excluir raíz y modifique el subdominio para añadir 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, tal como se muestra en la siguiente captura de pantalla.



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 Configuración de la aplicación, luego Administración de dominio.
4. En la página de administración de dominios, selecciona Administrar dominio.
5. En Editar dominio, puede editar los subdominios existentes.
6. (Opcional) Para agregar un nuevo subdominio, elija Agregar.
7. Elija Actualizar para guardar los cambios.

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 Configuración de la aplicación, luego Administración de dominio.
4. En la página de administración de dominios, selecciona Administrar dominio.
5. En Editar dominio, puede añadir o eliminar subdominios comodín.
   - Para agregar un nuevo subdominio comodín
     a. Elija Añadir.
     b. En el subdominio, introduzca un *****.
     c. En la ramificación de la aplicación, seleccione un nombre de ramificación de la lista.

        En la siguiente captura de pantalla de ejemplo, se ha creado el subdominio comodín `*.example.com` para la ramificación `dev` de la aplicación.

     d. Elija Actualizar para guardar los cambios.
   - 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. Elija Actualizar para guardar los cambios.

Configure 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 Configuración de la aplicación, luego Administración de dominio.
4. En la página de administración de dominios, selecciona Administrar dominio.
5. En la parte inferior izquierda, seleccione la casilla de verificación Detección automática de subdominios.

> **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 detección 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

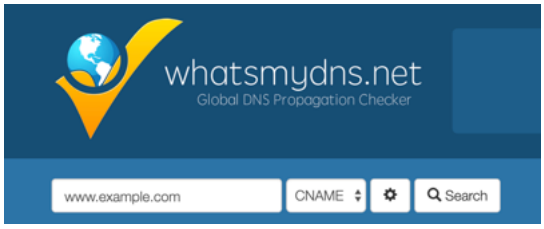
Si tiene algún problema al añadir un dominio personalizado a una aplicación en la consola de AWS Amplify , consulte los siguientes temas de esta sección para resolverlo.

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

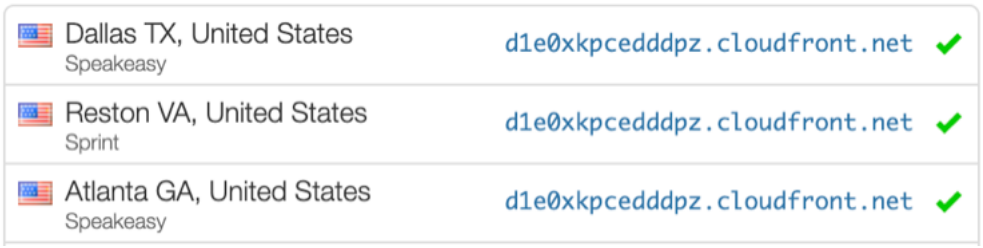
Temas

¿Cómo verifico 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 se ha quedado bloqueado en el estado Pendiente de verificación, compruebe que sus registros CNAME se resuelven correctamente. Consulte el anterior tema de la guía de solución de problemas, Cómo verifico que mi CNAME se resuelve, para obtener instrucciones sobre cómo llevar a cabo esta tarea.
2. Si sus registros CNAME no se resuelven, confirme con su proveedor de dominios que su configuración de DNS incluye la entrada CNAME.

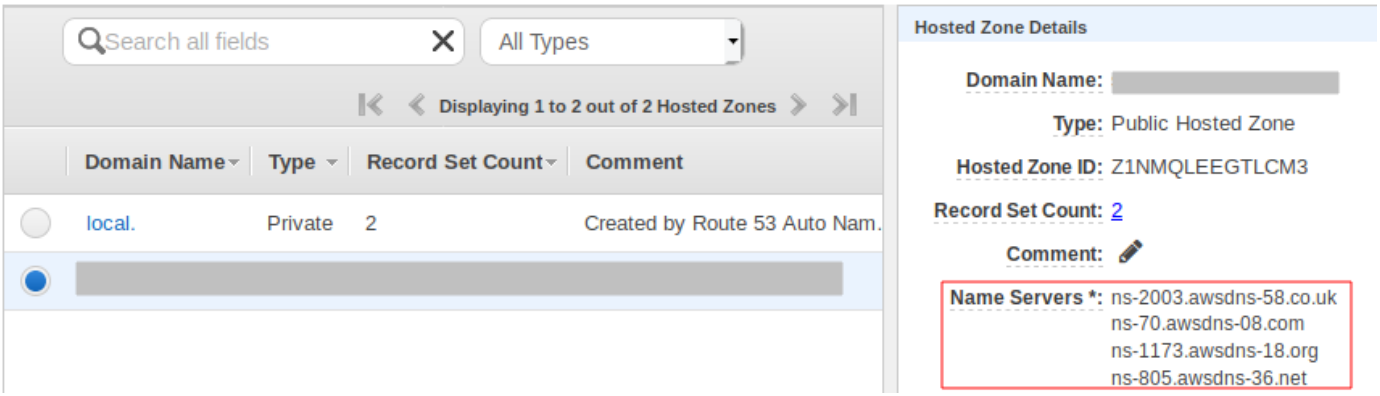
   > **Important**
   >
   > Es importante que actualice los registros CNAME en cuanto cree su dominio personalizado. Una vez que su aplicación se crea en la consola de Amplify, su registro CNAME se comprueba cada pocos minutos para determinar si llega a una resolución. Si no llega a una resolución transcurrida una hora, la comprobación se realizará cada pocas horas, lo que puede provocar un retraso en que su dominio esté listo para usar. Si ha agregado o actualizado sus registros CNAME horas después de crear la aplicación, es probable que la aplicación se quede bloqueada en estado Pendiente de verificación.

3. Si ha comprobado la existencia del registro CNAME, puede que haya algún problema con su proveedor de DNS. Puede contactar con el proveedor de DNS para diagnosticar el motivo por el cual el CNAME de verificación de DNS no llega a una resolución o migrar su DNS a Route 53. Para obtener más información, consulte Establecimiento de Amazon Route 53 como el servicio DNS de un dominio existente.

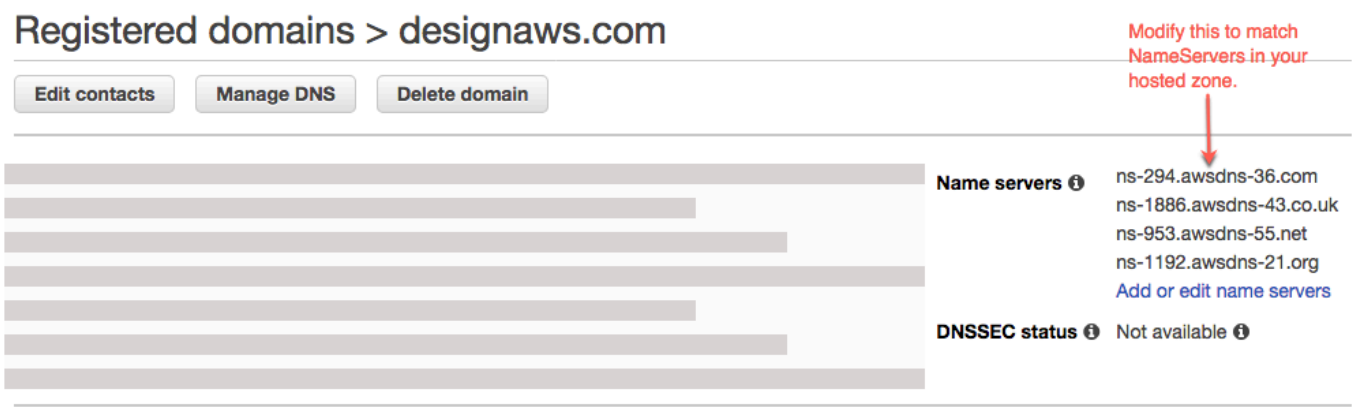
Mi dominio alojado con Amazon Route 53 está bloqueado en estado Verificación pendiente

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 AWS Support. Para obtener más información, consulte Creación de un caso de soporte en la Guía del usuario de AWS Support .

Aparece un error de CNAME AlreadyExistsException

Si recibes un AlreadyExistsException error de CNAME, 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 Lleve a cabo los siguientes pasos para diagnosticar la causa del error.

1. Inicia sesión en la CloudFrontconsola de Amazon y comprueba que no tienes este dominio implementado en ninguna otra distribución. Se puede adjuntar un único CNAME registro 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. 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 usa www.ejemplo.com para otra aplicación, no podrá usar www.ejemplo.com con la aplicación que está intentando conectar. Puede usar otros subdominios, como blog.ejemplo.com.
4. 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 sigue viendo esta excepción después de 6 horas, póngase en contacto con AWS Support. Para obtener más información, consulte Creación de un caso de soporte en la Guía del usuario de AWS Support .

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 AWS 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.

        - Añadir un dominio personalizado administrado en Amazon Route 53.
        - Añadir un dominio personalizado administrado por un proveedor de DNS externo.
        - Agregue un dominio personalizado administrado por GoDaddy.
        - Añada un dominio personalizado gestionado por Google Domains.
   - Póngase en contacto con nosotros AWS Support para que se vuelva a emitir su certificado SSL.

Configuración de ajustes de compilación

Al implementar una aplicación con Amplify Hosting, esta detecta automáticamente el marco de frontend y la configuración de compilación asociada al inspeccionar el archivo `package.json` en el repositorio. 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.

Puede editar la configuración de compilación de la aplicación en la consola de Amplify seleccionando Configuración de la aplicación y Configuración de compilación. 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.

> **Note**
>
> La configuración de compilación está visible 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.

Compilación de comandos y ajustes de especificación

El archivo YAML de especificación de compilación contiene un conjunto de comandos de compilación y ajustes 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. Aprenda a implementar un backend con su frontend.

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

El campo de memoria caché de buildspec se utiliza para almacenar en la memoria caché las dependencias en tiempo de compilación, como la carpeta node_modules, y se sugiere automáticamente en función del administrador de paquetes y el marco en el que se integra la aplicación del cliente. Durante la primera compilación, todas las rutas se almacenan en la memoria caché y, en las compilaciones posteriores, volvemos a inflar la caché y utilizamos esas dependencias almacenadas en la memoria caché siempre que es posible para acelerar el tiempo de compilación.

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

Sintaxis de YAML de especificación de compilación

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

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

Configuración de compilación específica de ramificación

Puede utilizar scripts del intérprete de comandos Bash para establecer la configuración de compilación específica de ramificación. Por ejemplo, el siguiente script utiliza la variable de entorno del sistema $AWS_BRANCH para ejecutar un conjunto de comandos si el nombre de ramificación es principal y otro conjunto de comandos si el nombre de ramificación es desarrollo.

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

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

Acceso 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

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 AL2023 de Amplify ejecuta su código con un nombre de usuario sin privilegios. `amplify` Amplify otorga a este usuario privilegios para ejecutar comandos del sistema operativo mediante el comando de Linux. `sudo` Si desea instalar paquetes de sistema operativo para las dependencias que faltan, puede utilizar comandos como `yum` y `rpm` with. `sudo`

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

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

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.

Deshabilitación de las compilaciones automáticas

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

Habilitar o deshabilitar la compilación e implementación de frontend basadas 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 Configuración de la aplicación 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`.

Habilite o deshabilite las compilaciones de backend basadas en diferencias

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 Configuración de la aplicación 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. Elija Guardar.

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.

Sintaxis de YAML de especificación de compilación de 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, consulte[Configuración 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.

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*
```

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.

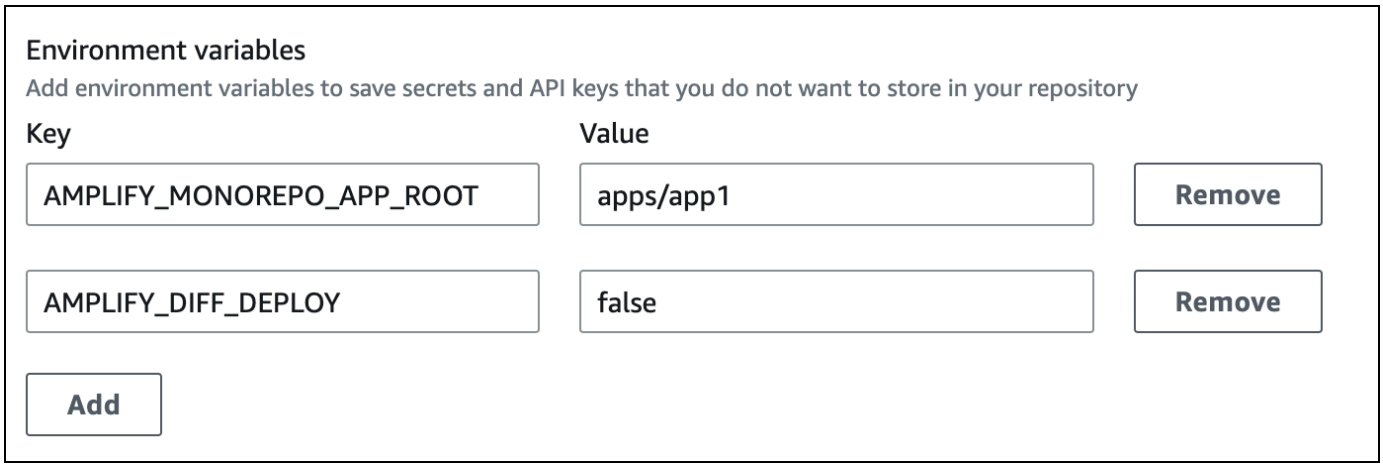
Configuración automática de la variable de entorno AMPLIFY_MONOREPO_APP_ROOT durante la implementación

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 Nueva aplicación, Alojar aplicación web en la esquina superior derecha.
3. En la página Aloje su aplicación web, elija su proveedor de Git y, a continuación, seleccione Continuar.
4. En la página Añadir ramificación de repositorio, haga lo siguiente:
    a. Elija el nombre de su repositorio de la lista de repositorios actualizados recientemente.
    b. Para Ramificación, elija la ramificación que desea usar.
    c. Seleccione ¿Desea conectar un monorepo? Seleccione una carpeta.
    d. Introduzca la ruta a su aplicación en su monorepo, por ejemplo, **`apps/app1`**.
    e. Elija Siguiente.

5. En la página de configuración de compilación, puede utilizar la configuración predeterminada o personalizar la configuración de compilación de su aplicación. En la siguiente captura de pantalla de ejemplo, en la sección Variables de entorno, Amplify configura `AMPLIFY_MONOREPO_APP_ROOT` en `apps/app1` utilizando la ruta que ha especificado en el paso 4d.



6. Elija 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 Configuración de la aplicación 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 Añadir variable.

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.

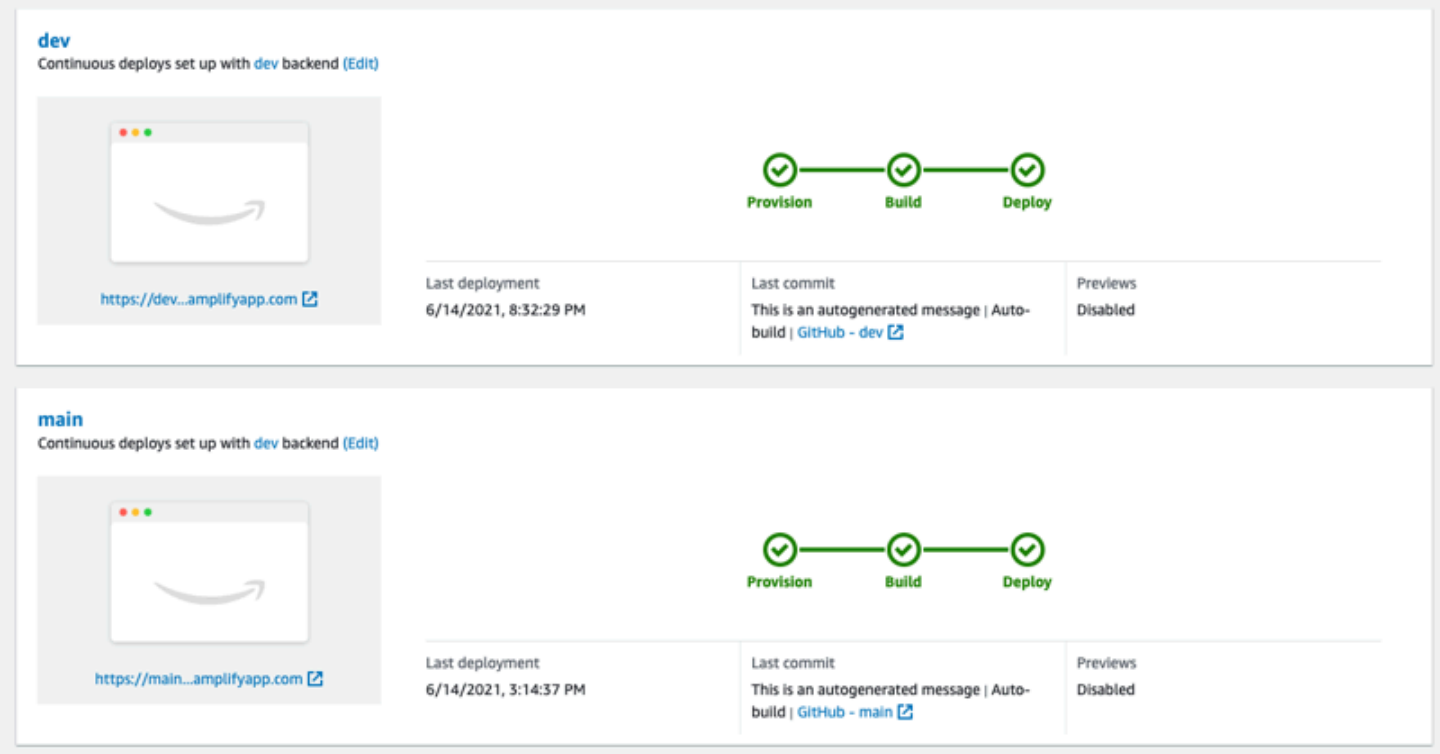
```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
          commands:
            - npm install -g pnpm
```

Implementaciones de ramificaciones de características y flujos de trabajo de equipo

Amplify Hosting se ha diseñado para trabajar con la ramificación de características y los flujos de trabajo de GitFlow. Amplify utiliza ramificaciones de Git para crear nuevas implementaciones cada vez que un desarrollador conecta una nueva ramificación en su repositorio. Después de conectar su primera ramificación, puede crear una nueva implementación de ramificaciones de características añadiendo una ramificación tal y como se indica a continuación:

1. En la página de lista de ramificación, elija Conectar ramificación.
2. Elija una ramificación de su repositorio.
3. Guarda y, a continuación, implemente su aplicación.

La aplicación ahora tiene dos implementaciones disponibles en https://main.appid.amplifyapp.com y https://dev.appid.amplifyapp.com. Esto puede variar de un equipo a otro, pero normalmente la ramificación principal rastrea el código de versión y es su ramificación 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 de equipo con entornos de backend de Amplify
- Implementaciones de ramificaciones de características basadas en patrones
- Generación automática de configuración de Amplify en tiempo de compilación
- Compilaciones de backend condicionales
- Use los backends de Amplify en todas las aplicaciones

Flujos de trabajo de equipo con entornos de backend de Amplify

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 más información acerca de esta implementación, consulte Introducción a las implementaciones continuas de pila completa.

> **Note**
>
> Puede reutilizar fácilmente los entornos de backend de Amplify en sus aplicaciones de Amplify. Para obtener más información, consulte Use los backends de Amplify en todas las aplicaciones.

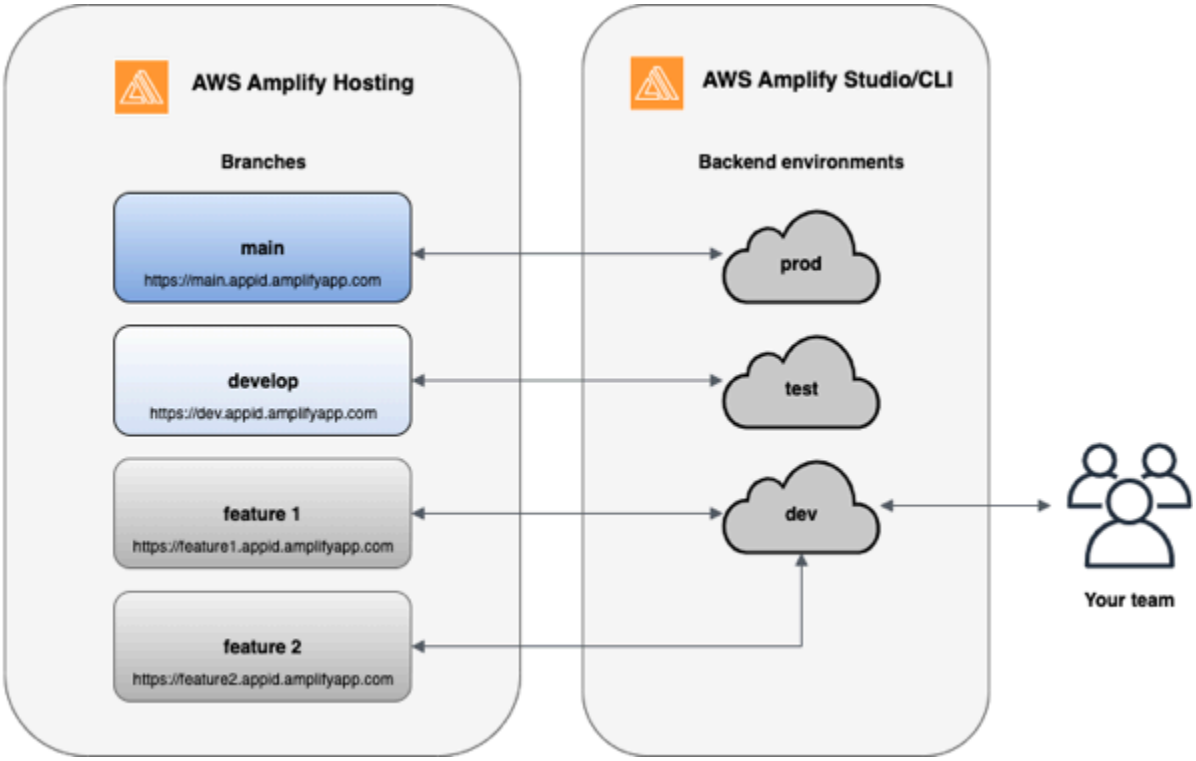
Puede usar Amplify Hosting para implementar de forma continua recursos de backend como las API de GraphQL y las funciones de Lambda con su implementación de ramificaciones de características. Puede usar los siguientes modelos de ramificación para implementar su backend y frontend con Amplify Hosting.

Temas

- Flujo de trabajo de ramificación de característica
- Flujo de trabajo de GitFlow
- Entorno de pruebas por desarrollador

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 tiene ningún proyecto, cree uno con herramientas de arranque como create-react-app o Gatsby.

```
create-react-app next-unicorn
cd next-unicorn
amplify init
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: prod
...
amplify push
```

3. Añada entornos de backend de pruebas y desarrollo.

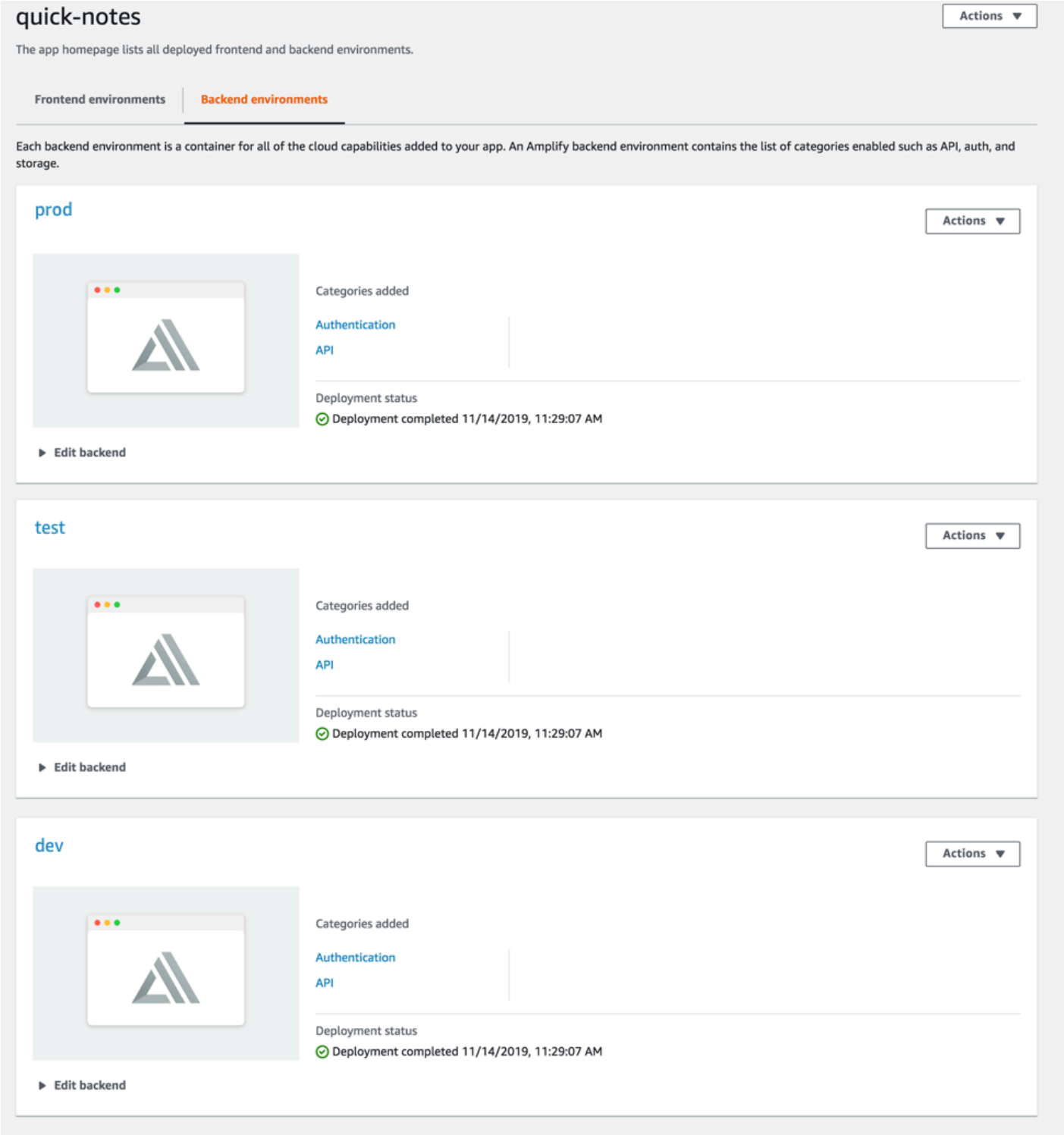
```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: test
...
amplify push
```

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

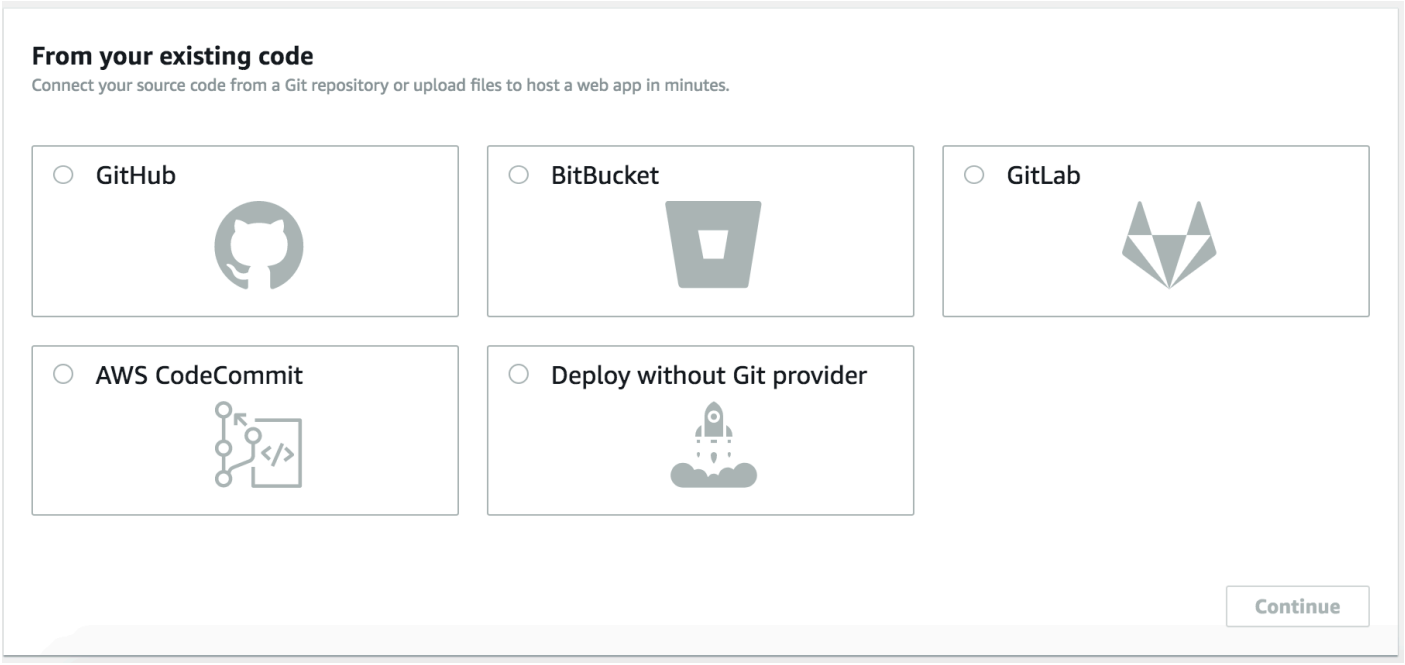
4. Inserte código en el repositorio de Git que elija (en este ejemplo supondremos que lo ha insertado en el principal).

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

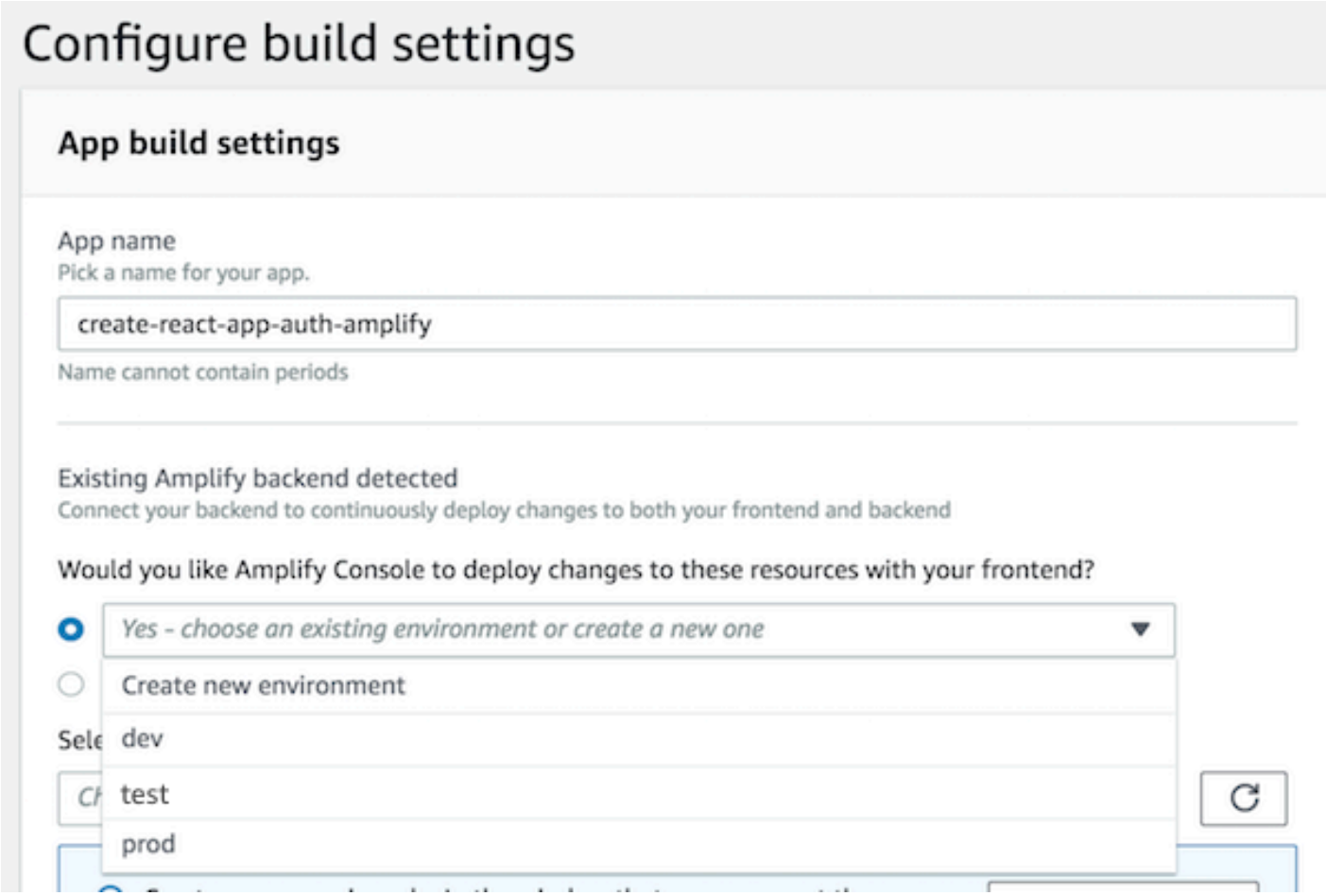
5. Visite Amplify en AWS Management Console para ver su entorno de backend actual. Desplácese un nivel hacia arriba desde la ruta de navegación para ver una lista de todos los entornos de backend creados en la pestaña de Entornos de backend.



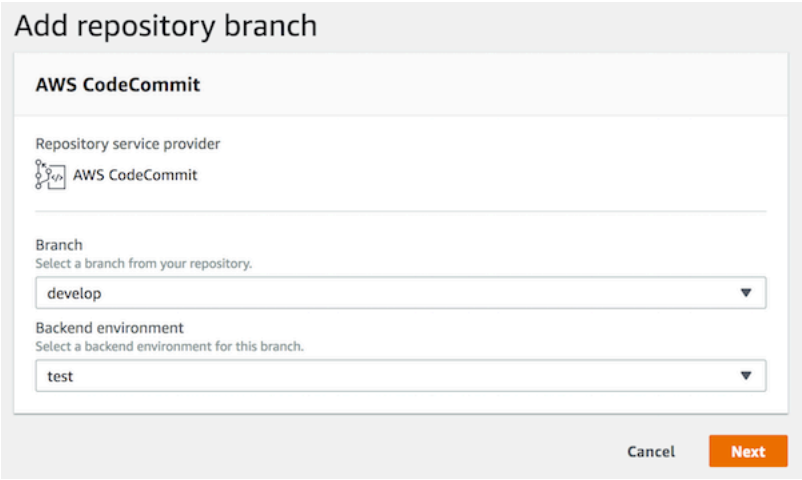
6. Pase a la pestaña de Entornos de frontend y conecte su proveedor de repositorios con la ramificación principal.



7. En la pantalla de configuración de compilación, seleccione un entorno de backend existente para configurar la implementación continua con la ramificación principal. Elija producción en el menú desplegable y conceda el rol de servicio a Amplify. Elija Guardar e implementar. Una vez que se complete la compilación, obtendrá una implementación de ramificaciones principales 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 tiene la AWS CLI instalada en el sistema (no la CLI de Amplify), podrá conectar una ramificación 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. Se podrá acceder a su característica en https://newinternet.appid.amplifyapp.com para compartirla con sus 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. De este modo se pondrá en marcha una compilación que actualizará el backend y el frontend en Amplify con una implementación de ramificaciones 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”).

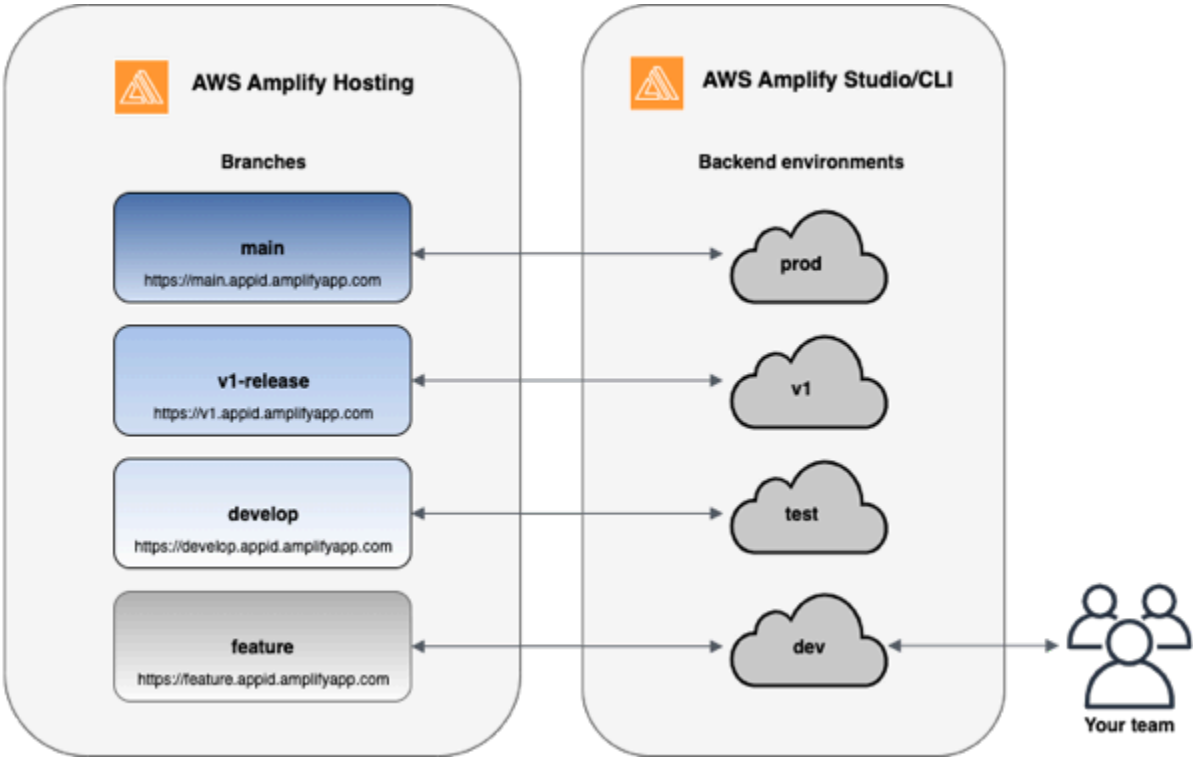
```
git push origin --delete newinternet
aws amplify delete-branch --app-id <appid> --branch-name <branchname>
```

```
amplify env remove dev
```

Flujo de trabajo de GitFlow

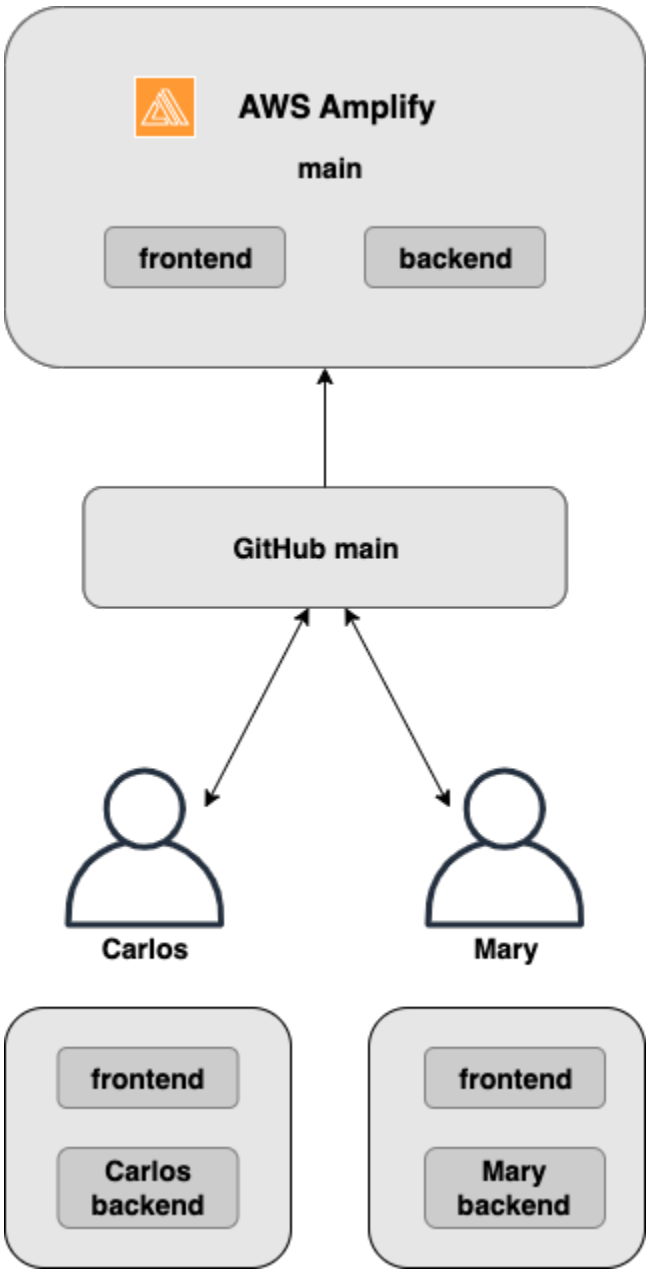
GitFlow utiliza dos ramificaciones para registrar el historial del proyecto. La ramificación principal realiza un seguimiento del código de la versión solo y la ramificación de desarrollo se utiliza como una ramificación de integración de nuevas características. GitFlow simplifica el desarrollo paralelo aislando el nuevo desarrollo del trabajo realizado. 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).

En el siguiente diagrama se 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 tiene ningún proyecto, cree uno con herramientas de arranque como create-react-app o 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).

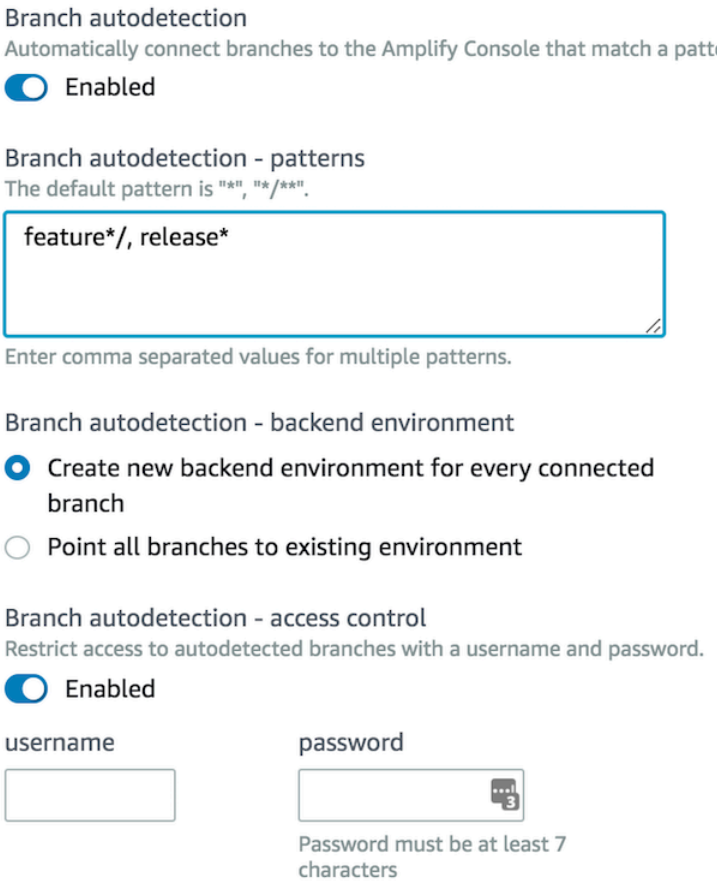
```
git commit -am 'Added mary sandbox'
git push origin main
```

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 que se complete la compilación, obtendrá una implementación de ramificaciones principales disponible en https://main.appid.amplifyapp.com con un nuevo entorno de backend que se enlaza a la ramificación.
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 nuevo entorno. Una vez que se complete la compilación, obtendrá una implementación de ramificaciones de desarrollo disponible en https://develop.appid.amplifyapp.com con un nuevo entorno de backend que se enlaza a la ramificación.

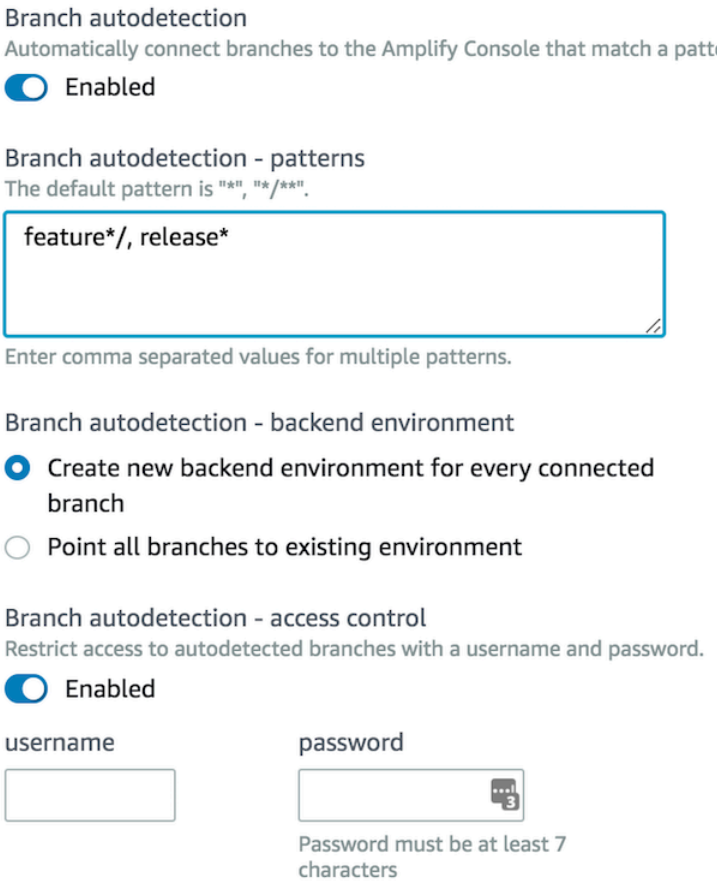
Implementaciones de ramificaciones de características basadas en patrones

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 usan la ramificación de características o los flujos de trabajo de GitFlow para sus versiones ya pueden definir patrones como “release**” para implementar automáticamente las ramificaciones de Git que empiezan por “release” en una URL compartible. En esta publicación de blog se describe el uso de esta característica con otros flujos de trabajo de equipo.

1. Elija App settings > General > Edit (Configuración de la aplicación > General > Editar).
2. Invierta el conmutador de detección automática de ramificaciones en Enabled (Habilitado).



1. Defina patrones para implementar ramificaciones de forma automática.
   - **`*`**: 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*`.
2. Configure la protección de contraseñas automática para todas las ramificaciones creadas automáticamente estableciendo Branch autodetection - access control (Detección automática de ramificaciones: control de acceso) en Enabled (Habilitado).
3. En el caso de las aplicaciones compiladas con un backend de Amplify, puede elegir crear un nuevo entorno o apuntar todas las ramificaciones a un backend existente.



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.

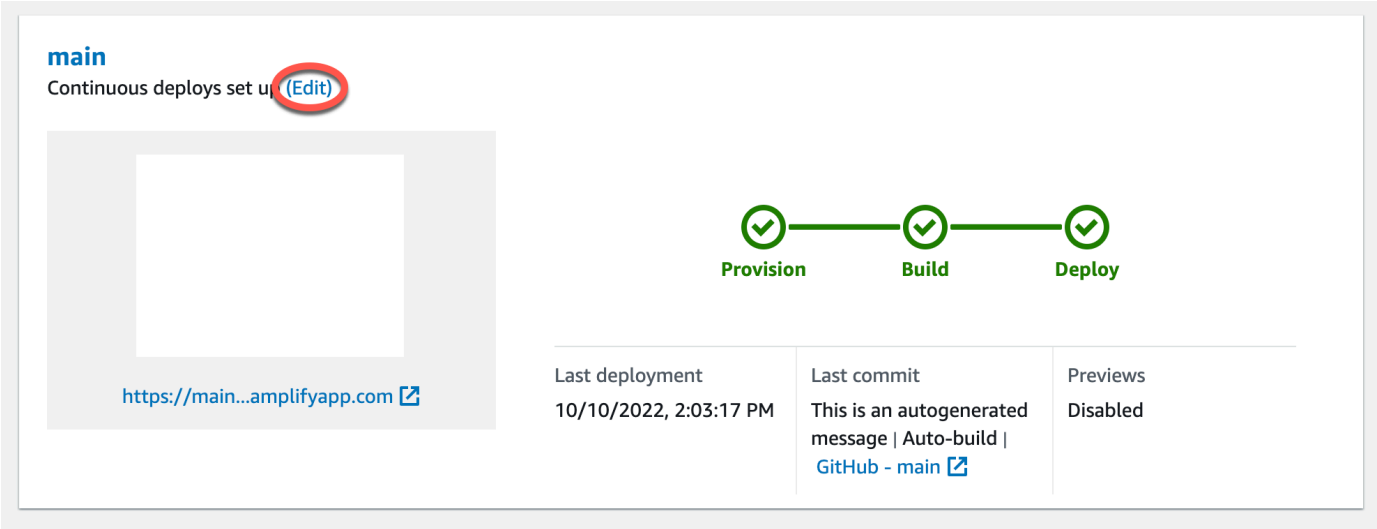
- Para obtener más información sobre la configuración de implementaciones de ramificaciones de características basadas en patrones, consulte Configure subdominios automáticos para un dominio personalizado de Amazon Route 53
- Para obtener más información sobre cómo conectar una aplicación de Amplify a un dominio personalizado gestionado en Route 53, consulte Añadir 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

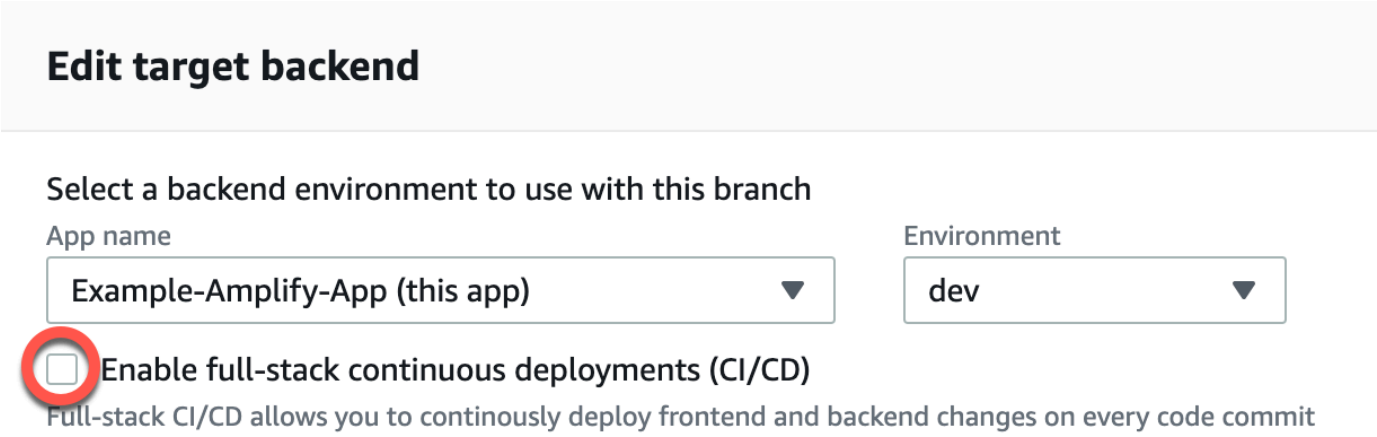
Amplify permite generar automáticamente el archivo `aws-exports.js` de configuración de Amplify en tiempo de compilación. Al desactivar las implementaciones de CI/CD de pila completa, permite que su aplicación genere automáticamente el archivo `aws-exports.js` y garantiza que no se lleven a cabo actualizaciones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

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

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
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 Adición de un rol de servicio.
7. Elija Save. Amplify aplicará estos cambios la próxima vez que compile la aplicación.

Compilaciones de backend condicionales

Amplify admite compilaciones de backend condicionales en todas las ramificaciones de una aplicación. 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 Habilite o deshabilite las compilaciones de backend basadas en diferencias.

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 Compilación de comandos y ajustes de especificación.

Use los backends de Amplify en todas las aplicaciones

Amplify le permite reutilizar fácilmente los entornos de backend existentes en todas sus aplicaciones 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. Inicie sesión en AWS Management Console y abra la consola de Amplify.
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. El CI/CD de pila completa está desactivado de forma predeterminada. Al desactivar el CI/CD de pila completa, la aplicación se ejecuta 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 Adición de un rol de servicio.

e. Elija Siguiente.

7. Elija Guardar e implementar.

Reutilice los backends al conectar una ramificación a una aplicación existente

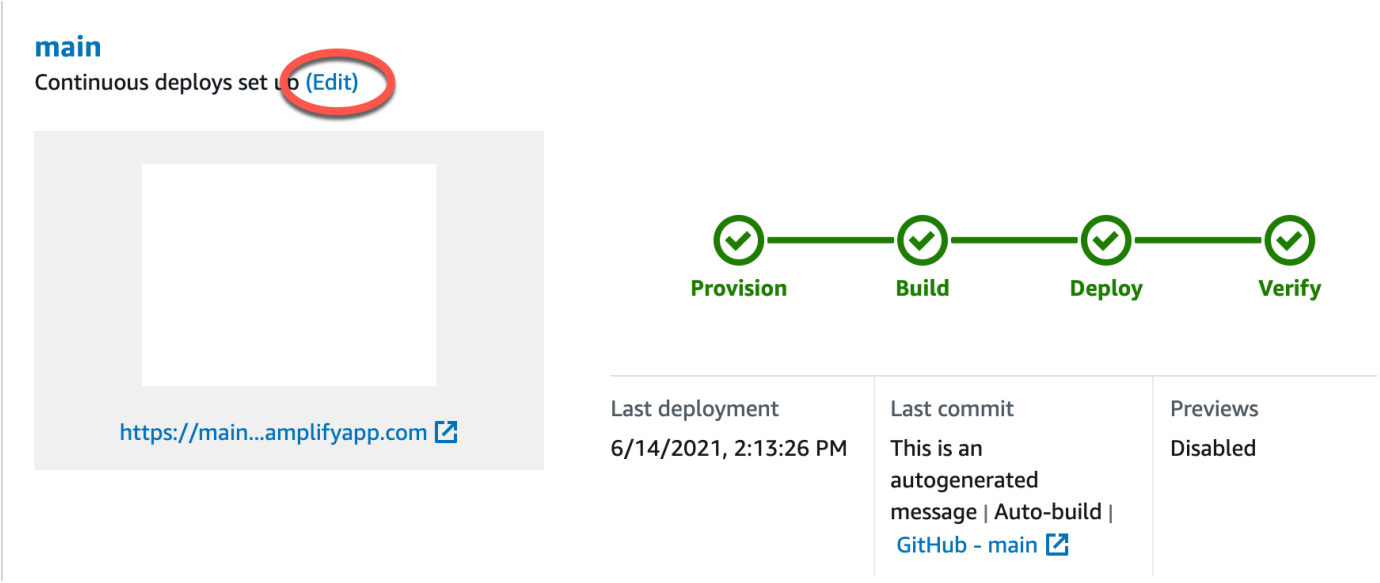
Para reutilizar un backend al conectar una ramificación a una aplicación Amplify existente

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación a la que desea conectar una nueva ramificación.
3. En el panel de navegación, elija Configuración de la aplicación, 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 Adición de un rol de servicio.
9. El CI/CD de pila completa está desactivado de forma predeterminada. Al desactivar el CI/CD de pila completa, la aplicación se ejecuta 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. Inicie sesión en AWS Management Console y abra la consola de Amplify.
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. El CI/CD de pila completa está habilitado de forma predeterminada. Desmarque esta opción para desactivar el CI/CD de pila completa en este backend. Al desactivar el CI/CD de pila completa, la aplicación se ejecuta 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 Save. Amplify aplicará estos cambios la próxima vez que compile la aplicación.

Implementaciones manuales

Las implementaciones manuales le permiten publicar su aplicación web con Amplify Hosting sin necesidad de conectar un proveedor de Git. Arrastre y suelte una carpeta desde su escritorio para alojar su sitio en 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.

En Amazon S3, también puede configurar activadores de AWS Lambda 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, por sus siglas en inglés). Para obtener más información, consulte Implemente aplicaciones renderizadas del servidor con Amplify Hosting.

Implementación manual mediante la función de arrastrar y soltar

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

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. El modo en que se desplaza a la página de alojamiento de su aplicación web depende de si empieza desde la página de inicio de Amplify o desde la página de todas las aplicaciones.
   - Desde la página de inicio de Amplify
     a. Elija Primeros pasos.
     b. En la sección Entrega elija Primeros pasos.
   - En la página de todas las aplicaciones
     - En la esquina superior derecha, elija Nueva aplicación, Alojar aplicación web
3. En la página Aloja tu aplicación web, elija Implementar sin el proveedor de Git. A continuación, elija Continuar.
4. En la sección Iniciar una implementación manual de Nombre de la aplicación, especifique el nombre de la aplicación.

5. En Nombre de entorno, especifique un nombre significativo para el entorno, como **`development`** o**`production`**.
6. En Método, elija Arrastrar y soltar.
7. Arrastre y suelte los archivos del escritorio en la zona de colocación o utilice Eligir archivos para seleccionar los archivos del ordenador. Los archivos que arrastra y suelta o selecciona pueden ser una carpeta o un archivo zip que contenga el nodo raíz de su sitio.
8. Elija Guardar e implementar.

Implementación manual de Amazon S3 o de la dirección URL

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

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. En la parte superior de la página, elija Primeros pasos.
3. En la sección Entrega elija Primeros pasos.
4. En la página Aloja tu aplicación web, elija Implementar sin el proveedor de Git. A continuación, elija Continuar.
5. En la sección Iniciar una implementación manual de Nombre de la aplicación, especifique el nombre de la aplicación.
6. En Nombre de entorno, especifique un nombre significativo para el entorno, como **`development`** o**`production`**.
7. En Método, elija Amazon S3 o Cualquier dirección URL.
8. El procedimiento para cargar los archivos depende del método de carga.
   - Amazon S3
     a. En Bucket, seleccione el nombre del bucket de Amazon S3 en la lista. Las listas de control de acceso (ACL, por sus siglas en inglés) deben estar habilitadas para el bucket que seleccione. Para obtener más información, consulte Resolución de problemas de acceso al bucket de Amazon S3.
     b. En el caso del archivo zip, seleccione el nombre del archivo zip que desee implementar.
   - Cualquier dirección URL
     - En el caso de la dirección URL del recurso, introduzca la dirección URL del archivo comprimido que desee implementar.

9. 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.

Resolución de problemas de acceso al bucket de Amazon S3

Al crear un bucket de Amazon S3, utilice su configuración de propiedad de objetos de Amazon S3 para controlar si se han habilitado o deshabilitado las listas de control de acceso (ACL) del bucket. Para implementar manualmente una aplicación en Amplify desde un bucket de Amazon S3, las ACL deben estar habilitadas en el bucket.

Si recibe un error de `AccessControlList` al realizar la implementación desde un bucket de Amazon S3, es porque el bucket se ha creado con las ACL deshabilitadas y debe habilitarlas 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.

Botón Deploy to Amplify (Implementar en Amplify)

El botón Implementar en la consola de Amplify le permite compartir proyectos de GitHub públicamente o en su equipo. La siguiente es una imagen del botón:

Incorporación del botón Deploy to Amplify Hosting (Implementar en Amplify Hosting) en un repositorio o blog

Añada el botón a su archivo README.md de GitHub, publicación de blog o cualquier otra página de marcado que represente 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 dirección URL de la consola de Amplify con un enlace a su repositorio GitHub. 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.

Utilice el siguiente ejemplo para añadir el botón a un archivo de marcado, como el archivo README.md de GitHub. 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>
```

Configurar el acceso de Amplify a repositorios de GitHub

Amplify ahora usa la característica de GitHub Apps para autorizar el acceso de solo lectura de Amplify a los repositorios de GitHub. Con la aplicación Amplify GitHub, 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 Apps, consulte Acerca de GitHub Apps en el sitio web de GitHub.

Cuando conecta una nueva aplicación almacenada en un repositorio de GitHub, Amplify usa GitHub App de forma predeterminada para acceder al repositorio. Sin embargo, las aplicaciones de Amplify existentes que ya haya conectado a repositorios de GitHub accederán mediante OAuth. La CI/CD seguirá funcionando en estas aplicaciones, pero le recomendamos encarecidamente que las migre para usar la nueva aplicación de Amplify GitHub.

Cuando implementa una nueva aplicación o migra una aplicación existente mediante la consola de Amplify, se le redirigirá automáticamente a la ubicación de instalación de la aplicación de 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. Use el formato `https://github.com/apps/aws-amplify-REGION` y sustituya *`REGIÓN`* por la región en la que implementará su aplicación de Amplify. Por ejemplo, para instalar la aplicación de Amplify GitHub en la región Oeste de EE. UU. (Oregón), acceda a https://github.com/apps/aws-amplify-us-west-2.

Temas

- Instalar y autorizar la aplicación Amplify GitHub en una nueva implementación
- Migrar una aplicación de OAuth existente a Amplify GitHub App
- Configurar la aplicación de Amplify GitHub para implementaciones AWS CloudFormation, CLI y SDK
- Configurar vistas previas web con la aplicación de Amplify GitHub

Instalar y autorizar la aplicación Amplify GitHub en una nueva implementación

Cuando implemente una nueva aplicación en Amplify a partir de código existente en un repositorio de GitHub, siga estas instrucciones para instalar y autorizar GitHub App.

Para instalar y autorizar la aplicación de Amplify GitHub

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. En la página Todas las aplicaciones, elija Nueva aplicación y, a continuación, Alojar aplicación web.
3. En la página Comenzar con Amplify Hosting, elija GitHub y, a continuación, elija Continuar.
4. Si es la primera vez que conecta un repositorio de GitHub, se abrirá una nueva página de GitHub.com en su navegador solicitando permiso para autorizar el acceso de AWS Amplify a su cuenta de GitHub. Elija Autorizar.
5. A continuación, deberá instalar la aplicación de Amplify GitHub en su cuenta de GitHub. Se abrirá una página de GitHub.com solicitando permiso para instalar y autorizar AWS Amplify en su cuenta de GitHub.
6. Seleccione la cuenta de GitHub donde desea instalar la aplicación de Amplify GitHub.
7. Haga 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.

Migrar una aplicación de OAuth existente a Amplify GitHub App

Las aplicaciones de Amplify existentes conectadas previamente a repositorios de GitHub emplean OAuth para acceder a los repositorios. Le recomendamos firmemente que migre estas aplicaciones para usar la aplicación de Amplify GitHub.

Siga estas instrucciones para migrar una aplicación y eliminar el correspondiente webhook de OAuth de su cuenta de GitHub. Tenga en cuenta que el procedimiento de migración varía en función de si la aplicación de Amplify GitHub está ya instalada. Después de migrar su primera aplicación e instalar y autorizar GitHub App, solo necesitará actualizar los permisos del repositorio para las siguientes migraciones de aplicaciones.

Para migrar una aplicación de OAuth a GitHub App

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación que desea migrar.
3. En la página de información de la aplicación, busque el mensaje azul Migrar a GitHub App y elija Iniciar migración.
4. En la página Instalar y autorizar GitHub App, elija Configurar GitHub App.
5. Se abrirá una nueva página de GitHub.com en su navegador solicitando permiso para autorizar AWS Amplify en su cuenta de GitHub. Elija Autorizar.
6. Seleccione la cuenta de GitHub donde desea instalar la aplicación de Amplify GitHub.
7. Haga 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 App de su aplicación en la consola de Amplify. Si la autorización de GitHub 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.

Configurar la aplicación de Amplify GitHub para implementaciones AWS CloudFormation, CLI y SDK

Las aplicaciones de Amplify existentes conectadas previamente a repositorios de GitHub emplean OAuth para acceder a los repositorios. Pueden ser aplicaciones que haya implementado mediante la interfaz de línea de comandos de Amplify (CLI), AWS CloudFormation o SDK. Le recomendamos firmemente que migre estas aplicaciones para usar la nueva aplicación de Amplify GitHub. La migración debe llevarse a cabo en la consola Amplify de la AWS Management Console. Para obtener instrucciones, consulte Migrar una aplicación de OAuth existente a Amplify GitHub App.

Puede usar AWS CloudFormation, la CLI de Amplify y los SDK para implementar una nueva aplicación de Amplify que acceda a los repositorios mediante GitHub App. Para llevar a cabo este proceso, deberá instalar previamente la aplicación de Amplify GitHub en su cuenta de GitHub. Después, tendrá que generar un token de acceso personal en su cuenta de GitHub. Por último, deberá implementar la aplicación y especificar el token de acceso personal.

Instale la aplicación de Amplify GitHub en su cuenta

1. Abra un navegador web y acceda a la ubicación de instalación de la aplicación de Amplify GitHub para la región de AWS en la que desea implementar su aplicación.

   Use el formato `https://github.com/apps/aws-amplify-REGION/installations/new`, sustituyendo *REGIÓN* por la región seleccionada. 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. Seleccione la cuenta de GitHub donde desea instalar la aplicación de Amplify GitHub.
3. Haga 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.

Genere un token de acceso personal en su cuenta de GitHub

1. Inicie sesión en su cuenta de GitHub.

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 GitHub Apps, en el menú de navegación de la izquierda, elija 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 proporcionarlo cuando implemente una aplicación de Amplify con CLI, AWS CloudFormation o SDK.

Tras instalar la aplicación de Amplify GitHub en su cuenta de GitHub y generar un token de acceso personal, podrá implementar una nueva aplicación con la CLI de Amplify, AWS CloudFormation o SDK. En el campo `accessToken`, introduzca el token de acceso personal que creó en el procedimiento anterior. Para obtener más información, consulte Crear aplicación en la referencia de la API de Amplify y AWS::Amplify::App en la Guía del usuario de AWS CloudFormation.

El siguiente comando de CLI implementa una nueva aplicación de Amplify que emplea GitHub App para acceder al repositorio. Sustituya *myapp-using-githubapp*, *https://github.com/Myaccount/react-app* y *MY_TOKEN* con su información.

```
aws amplify create-app --name myapp-using-githubapp --repository https://github.com/
Myaccount/react-app --access-token MY_TOKEN
```

Configurar vistas previas web con la aplicación de Amplify GitHub

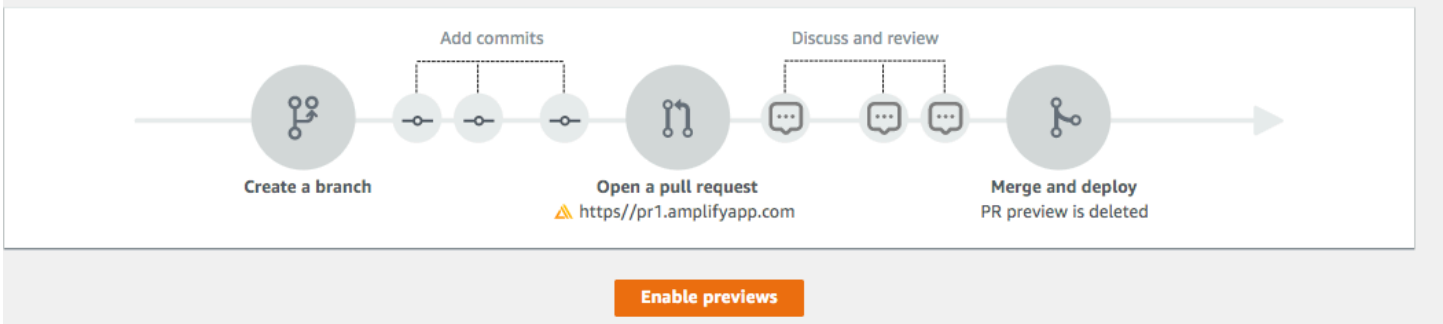
Una vista previa web implementa todas las solicitudes de extracción (PR) realizadas en su repositorio de GitHub en una URL de vista previa única. Ahora, las vistas previas usan la aplicación de Amplify GitHub para acceder a su repositorio de GitHub. Para obtener más instrucciones sobre cómo instalar y autorizar GitHub App para las vistas previas web, consulte Habilite las vistas previas web.

Vistas previas web para solicitudes de extracción

Las vistas previas web ofrecen a los equipos de desarrollo y control de calidad (QA) una manera de previsualizar los cambios de las solicitudes de extracción (PR) antes de fusionar el código en una ramificación 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.

> **Note**
>
> Actualmente, la ramificación de vista previa de Amplify es compatible con GitLab y BitBucket. AWS CodeCommit no tiene una paridad total de características con GitHub. La variable de entorno `AWS_PULL_REQUEST_ID` solo se encuentra disponible al usar GitHub como proveedor de repositorios.

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) genera un backend efímero que se elimina al cerrar la PR.



> **Important**
>
> Por motivos de seguridad, puede 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.

Habilite las vistas previas web

En el caso de aplicaciones almacenadas en un repositorio de GitHub, las vistas previas emplean la aplicación Amplify GitHub para acceder al repositorio. Si habilita las vistas previas web en una aplicación de Amplify existente que haya implementado previamente desde un repositorio de GitHub con acceso mediante OAuth, deberá migrar la aplicación para poder usar la aplicación Amplify GitHub. Para obtener información sobre cómo realizar la migración, consulte Migrar una aplicación de OAuth existente a Amplify GitHub App.

Para habilitar las vistas previas web de solicitudes de extracción

1. Acceda a Configuración de aplicación, Vistas previas y, a continuación, elija Habilitar 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 más información sobre este tipo de implementación, consulte Introducción al código existente.

2. En repositorios de GitHub, siga estos pasos para instalar y autorizar la aplicación Amplify GitHub en su cuenta:

   a. En la ventana Instalar la aplicación GitHub para habilitar vistas previas, elija Instalar aplicación GitHub.

   b. Seleccione la cuenta de GitHub en la que desea configurar la aplicación Amplify GitHub.

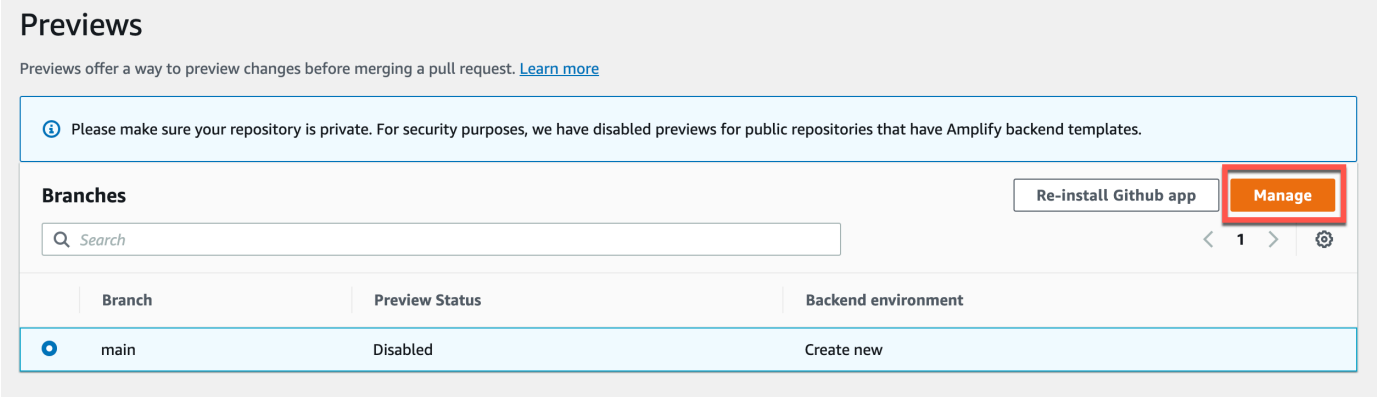
   c. Se abrirá una página en GitHub.com para configurar los permisos de repositorio de su cuenta.

   d. Haga 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.

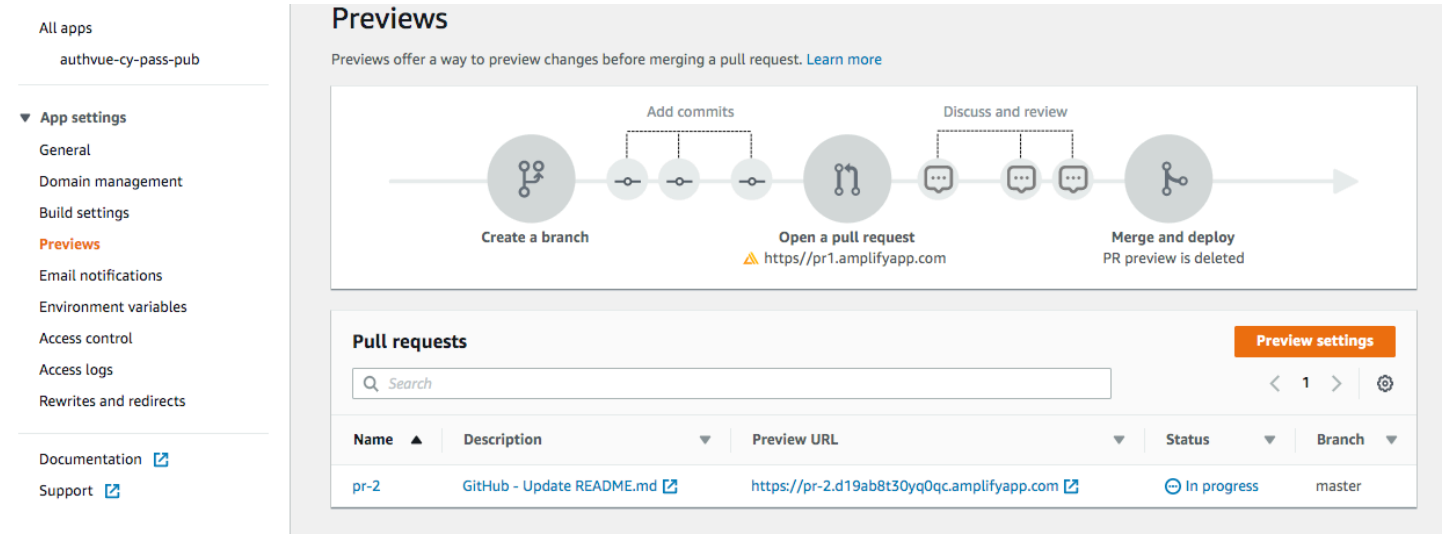
e. Elija Guardar

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 Gestionar.

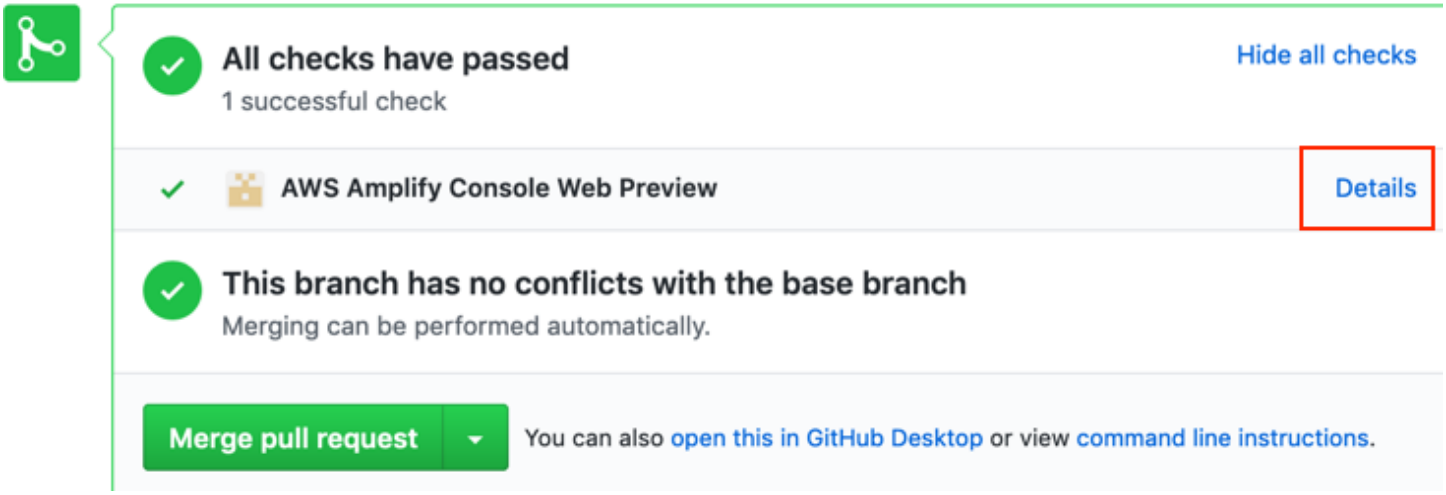


4. En la ventana Gestionar configuración de vista previa de ramificación, active Vistas previas de solicitudes de extracción.
5. Para aplicaciones de pila completa, siga uno de estos pasos:
   - Elija Crear nuevo entorno de backend para cada solicitud de extracción. Esta opción le permitirá probar los 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.



En repositorios de GitHub, puede acceder a una vista previa de su URL directamente desde la solicitud de extracción en su cuenta de GitHub.



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.

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 Configure subdominios automáticos para un dominio personalizado de Amazon Route 53.

Añada pruebas integrales de Cypress a su aplicación de Amplify

Puede ejecutar pruebas integrales (E2E) en la fase de pruebas de su aplicación de 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.

Tutorial: Configurar pruebas integrales con Cypress

Cypress es un marco de pruebas basado en JavaScript que le permite ejecutar pruebas E2E en un navegador. Para ver un tutorial de configuración de pruebas E2E, consulte la entrada del blog Ejecución de pruebas integrales de Cypress para su implementación CI/CD de pila completa con Amplify.

Agregue pruebas a su 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 (prueba previa)

> 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.

test

> Ejecuta los comandos de Cypress para realizar pruebas con mochawesome.

postTest (prueba posterior)

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.

artifacts>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.

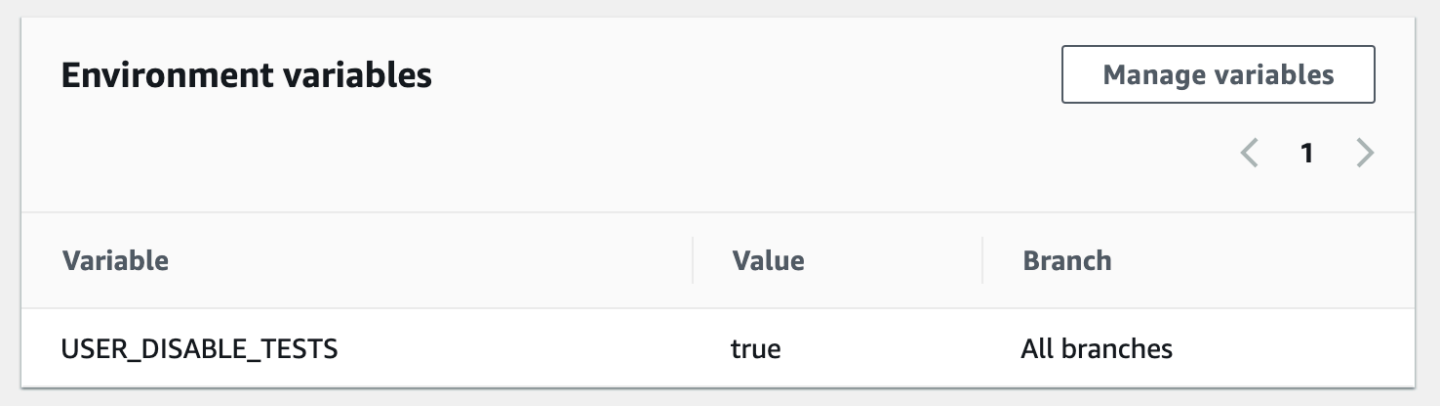
```
test:
  phases:
    preTest:
      commands:
        - npm ci
        - npm install -g pm2
        - npm install -g wait-on
        - npm install mocha mochawesome mochawesome-merge mochawesome-report-generator
        - pm2 start npm -- start
        - wait-on http://localhost:3000
    test:
  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

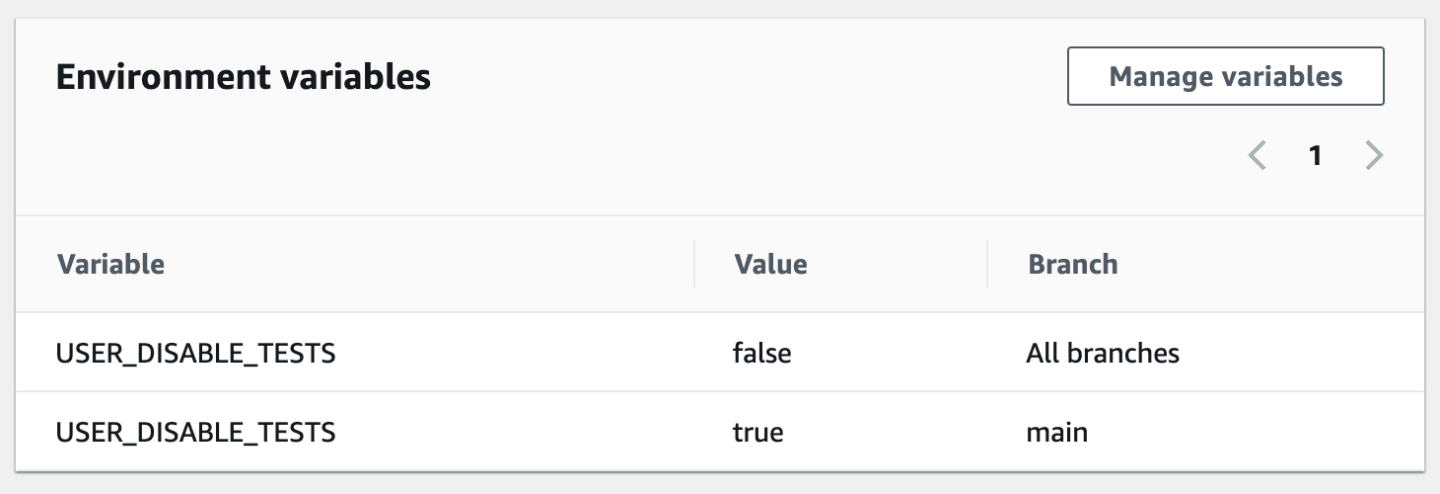
Desactivar las pruebas

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 desea deshabilitar globalmente la ejecución de las pruebas o ejecutarlas solo en ramificaciones específicas, puedes usar la variable de entorno USER_DISABLE_TESTS sin tener que modificar la configuración de compilación.

Para deshabilitar globalmente las pruebas en todas las ramificaciones, agregue la variable de entorno USER_DISABLE_TESTS con un valor de `true` para todas las ramificaciones. 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 en una ramificación específica, añada la variable de entorno USER_DISABLE_TESTS con un valor de `false` para todas las ramificaciones y, a continuación, añada una anulación para cada ramificación 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.

Uso de redireccionamientos

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.

Tipos de redireccionamiento

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.

Creación y edición de redireccionamientos

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 Configuración de la aplicación y, a continuación, elija Reescrituras y redireccionamientos.
4. En la sección Reescrituras y redireccionamientos, elija Editar.
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 Añadir 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 Añadir masivamente reescrituras y redireccionamientos.

6. Elija Guardar.

Orden de redireccionamientos

Los redireccionamientos se ejecutan 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
```

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.

Redireccionamientos y reescrituras sencillos

En esta sección se incluye código de ejemplo de situaciones de redireccionamiento comunes.

> **Note**
>
> La coincidencia de dominios de direcciones originales no distingue entre mayúsculas y minúsculas.

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 de SPA admiten HTML5 history.pushState() para cambiar la ubicación del navegador sin desencadenar una solicitud del 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
</^[^.]+$\|\.(?!(css\|gif\|ico\|jpg\|js\|png\|txt\|svg\|woff\|woff2\|ttf\|map\|json\|webp)$)([^.]+$)/>	/index.html	200	

JSON [{"source": "</^[^.]+$|\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|webp)$)([^.]+$)/>", "status": "200", "target": "/index.html", "condition": null}]

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}]

Barras finales y direcciones URL limpias

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 direcciones URL limpias añadiendo una barra diagonal siempre que sea 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.htm l 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>"}]

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.

El siguiente es un ejemplo de una redirección válida 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 ramificaciones

Si está trabajando en características que aún no se han lanzado, puede proteger con contraseña aquellas ramificaciones de características que aún no disponen de acceso público. 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.

Para establecer contraseñas en las ramificaciones de características

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación cuyas ramificaciones de características desea proteger con contraseña.
3. En el panel de navegación, elija Configuración de la aplicación y luego elija Control de acceso.
4. En la sección Configuración de control de acceso, elija Gestionar acceso.
5. Siga uno de estos pasos en la Configuración de control de acceso:
   - Para configurar un nombre de usuario y una contraseña en todas las ramificaciones conectadas, active Aplicar una contraseña global. Por ejemplo, si tiene las ramificaciones principales, de desarrollo y de características conectadas, puede utilizar una contraseña global para establecer el mismo nombre de usuario y contraseña para todas las ramificaciones.
   - Para configurar un nombre de usuario y una contraseña en una ramificación concreta, desactive Aplicar una contraseña global. En la ramificación para la que desee establecer un nombre de usuario y contraseña únicos, elija Restringido con contraseña en Configuración de acceso e introduzca un nombre de usuario, así como una contraseña.
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.

Variables de entorno

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 no permite crear variables de entorno con un `AWS` prefijo. Este prefijo está reservado únicamente para el uso interno de Amplify.

> **Important**
>
> No utilice variables de entorno para almacenar claves secretas. Guarde los secretos en un secreto de entorno creado con el almacén de AWS Systems Manager parámetros. Para obtener más información, consulte Secretos del entorno.

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 la compilación en minutos	`30`
_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	`true`

Nombre de variable	Descripción	Ejemplo de valor
	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 Desactivar las pruebas.	
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/...`
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`

Nombre de variable	Descripción	Ejemplo de valor
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. Esta variable de entorno solo está disponible cuando se utiliza GitHub como proveedor de repositorios.	1
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`

Nombre de variable	Descripción	Ejemplo de valor
AMPLIFY_DIFF_DEPLOY	Habilite o deshabilite la implementación de frontend basada en diferencias. Para obtener más información, consulte Habilitar o deshabilitar la compilación e implementación de frontend basadas en diferencias.	`true`
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. Para obtener más información, consulte Habilite o deshabilite las compilaciones de backend basadas en diferencias	`true`
AMPLIFY_BACKEND_PULL_ONLY	Amplify gestiona esta variable de entorno. 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. 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`.	`true`
AMPLIFY_ENABLE_DEBUG_OUTPUT	Establezca esta variable en `true` para imprimir un rastreo de pila 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. Para obtener más información, consulte el límite de permisos de IAM para roles generados 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. Para obtener más información, consulte la actualización de esquemas.	`true`

> **Note**
>
> Las variables de `AMPLIFY_AMAZON_CLIENT_SECRET` entorno `AMPLIFY_AMAZON_CLIENT_ID` y las variables de entorno son símbolos de OAuth, no una clave de AWS acceso ni una clave secreta.

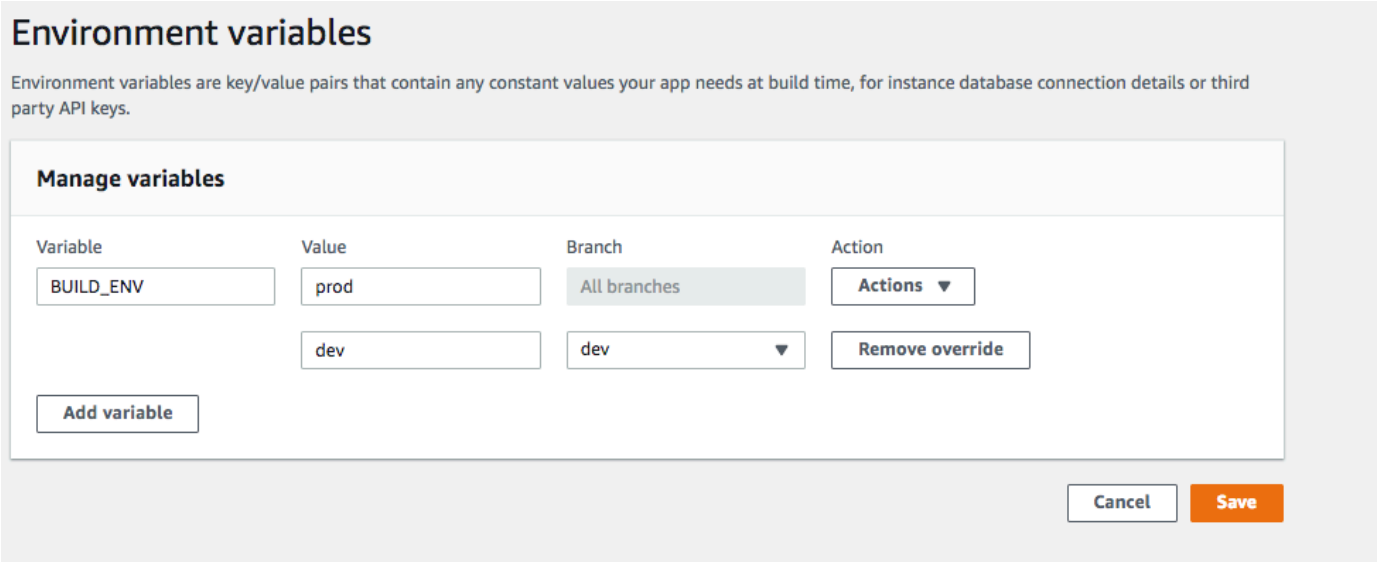
Configuración de las variables de entorno

Utilice las siguientes instrucciones para configurar las variables de entorno de una aplicación en la consola 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

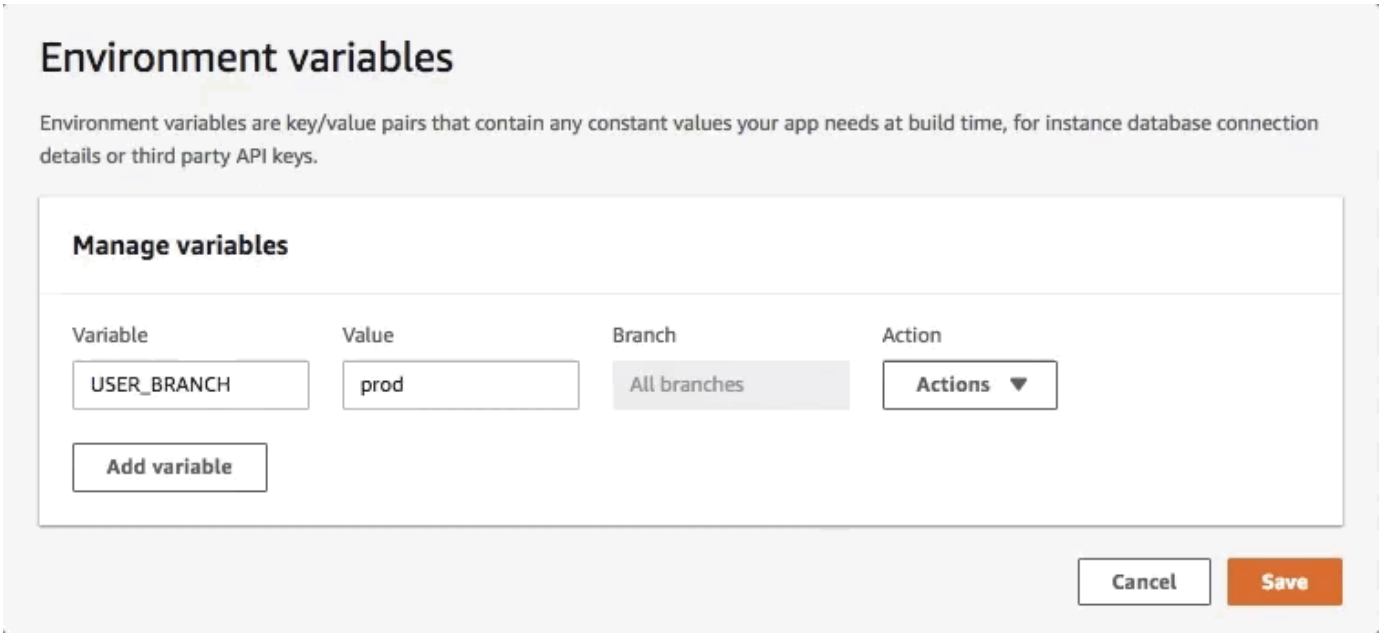
1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. En la consola de Amplify, elija Configuración de la aplicación y, a continuación, elija Variables de entorno.
3. En la sección Variables de entorno, elija Administrar variables.
4. En la sección Administrar variables en Variable, especifique 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. Elija Guardar.

Acceda 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.

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 Configuración de la aplicación 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. En este ejemplo, se cambia el comportamiento del npm (BUILD_ENV) y se añade 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
```

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.

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

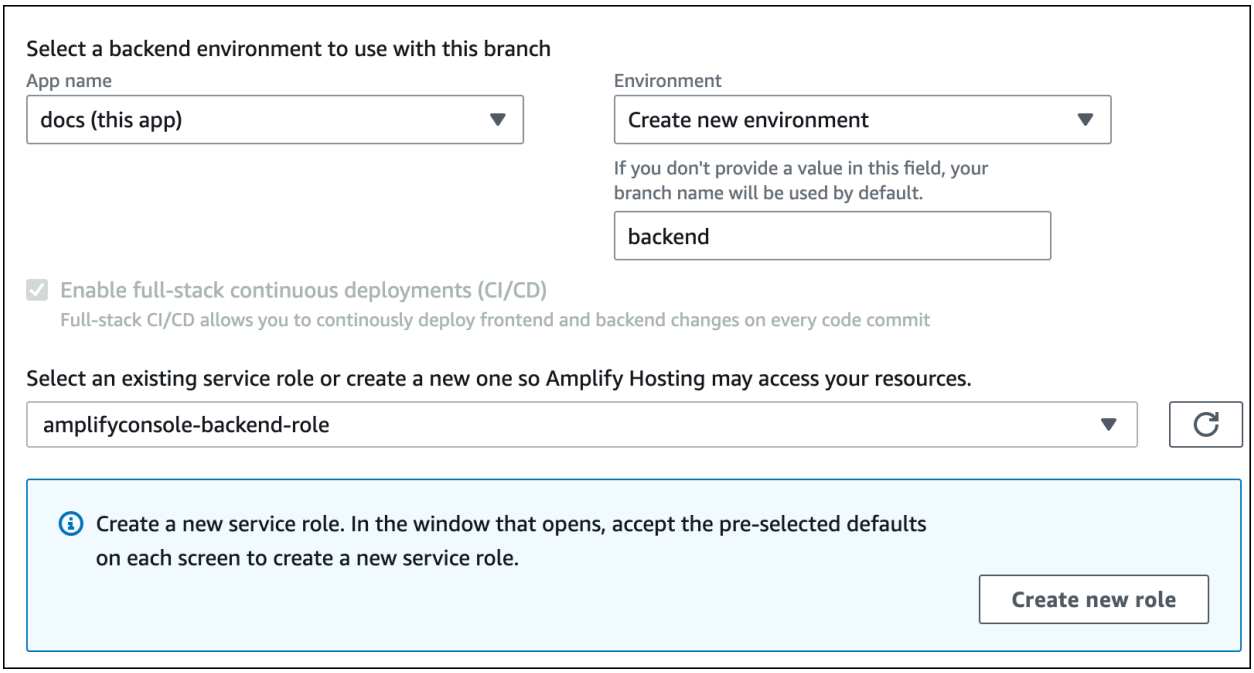
compilación, consulte los ejemplos de cómo añadir variables de entorno en la sección de comandos de compilación.

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 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.

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 Acceda a las variables de entorno en el momento de la compilación.

Secretos del entorno

Los secretos de entorno son similares a las variables de entorno, pero son pares clave-valor del almacén de parámetros AWS Systems Manager (SSM) que se pueden cifrar. Algunos valores deben estar cifrados, como la clave privada del inicio de sesión con Apple de Amplify.

Establecer secretos del entorno

Usa las siguientes instrucciones para establecer un secreto de entorno para una aplicación de Amplify 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

El acceso a un secreto de entorno 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`.

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-----

Encabezados personalizados

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 AWS Management Console, o bien descargando y editando el archivo `customHttp.yml` de una aplicación y guardándolo 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 editando las especificaciones de compilación (buildspec) en AWS Management Console o descargando y actualizando el archivo `amplify.yml`, así como guardándolo en el directorio raíz del proyecto. Los encabezados personalizados especificados de tal manera deberían migrarse fuera del buildspec y del archivo `amplify.yml`. Para obtener instrucciones, consulte Migración de encabezados personalizados.

Encabezado personalizado en formato 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.

key

El nombre del encabezado personalizado.

value

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 AWS Amplify. Puede especificar los encabezados en la AWS Management Console o bien descargar y editar el archivo `customHttp.yml` de la aplicación, que deberá guardar en el directorio raíz del proyecto.

Para configurar encabezados personalizados de una aplicación en la AWS Management Console

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.

2. Elija la aplicación para la que desea configurar encabezados personalizados.
3. En el panel de navegación, elija Configuración de la aplicación y Encabezados personalizados.
4. En la sección Especificación de encabezados personalizados, elija Editar.
5. En la ventana Editar, 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.

Para configurar encabezados personalizados mediante el archivo customHttp.yml

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación para la que desea configurar encabezados personalizados.
3. En el panel de navegación, elija Configuración de la aplicación y Encabezados personalizados.
4. En la sección Especificación de encabezados personalizados, elija Descargar.
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 AWS Management Console.

Migración de encabezados personalizados

Anteriormente, los encabezados HTTP personalizados de una aplicación se especificaban editando el buildspec en la AWS Management Console o descargando y actualizando el archivo `amplify.yml`, así como guardándolo en el directorio raíz del proyecto. Se recomienda encarecidamente migrar los encabezados personalizados fuera del buildspec y del archivo `amplify.yml`.

Especifique sus encabezados personalizados en la sección Encabezados personalizados de la AWS Management Console o descargando y editando el archivo `customHttp.yml`.

Para migrar los encabezados personalizados almacenados en la consola de Amplify

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación cuyos encabezados personalizados desee migrar.
3. En el panel de navegación, elija Configuración de la aplicación, y luego 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 Configuración de la aplicación y Encabezados personalizados.
8. En la sección Especificación de encabezados personalizados, elija Editar.
9. En la ventana Editar, introduzca la información de los encabezados personalizados que eliminó en el paso 6.
10. Elija 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. Inicie sesión en AWS Management Console y abra la consola de Amplify.
5. Elija la aplicación para la que desea configurar encabezados personalizados.
6. En el panel de navegación, elija Configuración de la aplicación y Encabezados personalizados.
7. En la sección Especificación de encabezados personalizados, elija 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 AWS Management Console.

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 Encabezado personalizado en formato YAML.
- Puede especificar encabezados personalizados para una aplicación en monorepo en la sección Encabezados personalizados de la AWS Management Console. Tenga en cuenta que 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 AWS Management Console.

Ejemplo de encabezados 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'"
```

Ejemplo de encabezado de control de caché

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 (1 hora).

```
customHeaders:
  - pattern: '/img/*'
    headers:
      - key: 'Cache-Control'
        value: 's-maxage=3600'
```

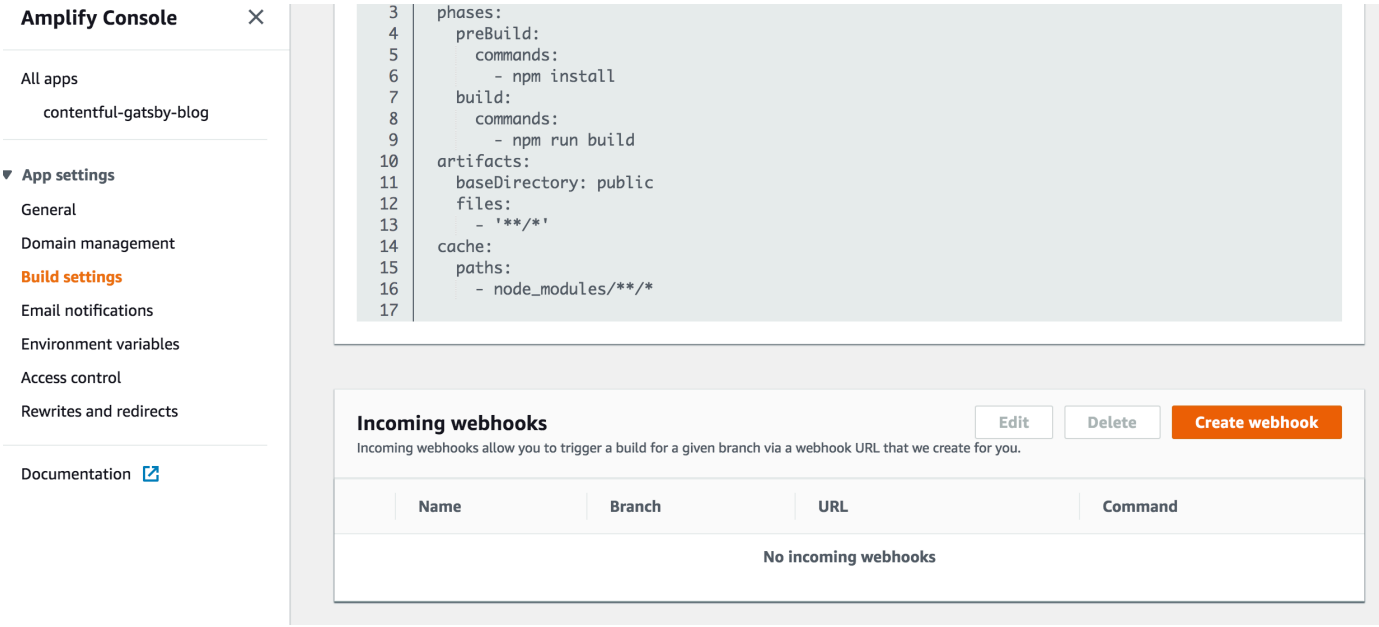
Para obtener más información sobre cómo controlar el rendimiento de las aplicaciones con encabezados, consulte Uso de los encabezados para controlar la duración del almacenamiento en caché.

Webhooks entrantes

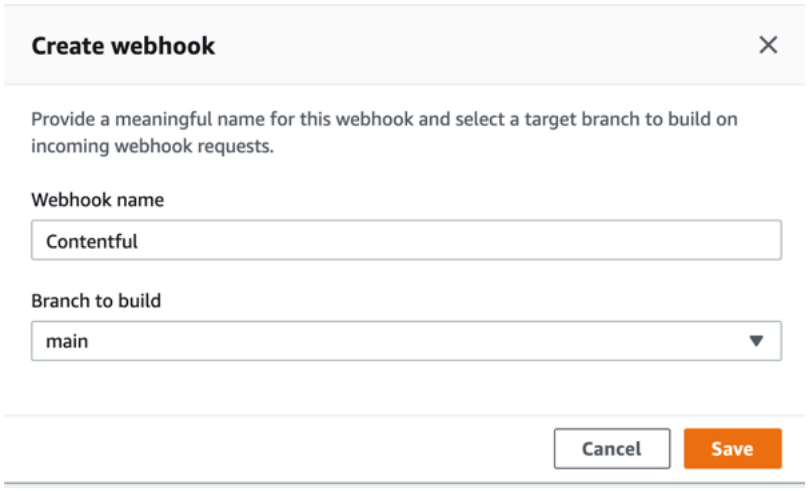
Configura un webhook entrante en la consola de Amplify para activar una compilación sin enviar código a tu repositorio de Git. Puede usar los desencadenadores de webhooks con herramientas CMS sin pantalla (como Contentful o GraphCMS) para iniciar compilaciones cuando haya cambios o compilaciones diarias mediante servicios como Zapier.

Para crear un webhook entrante

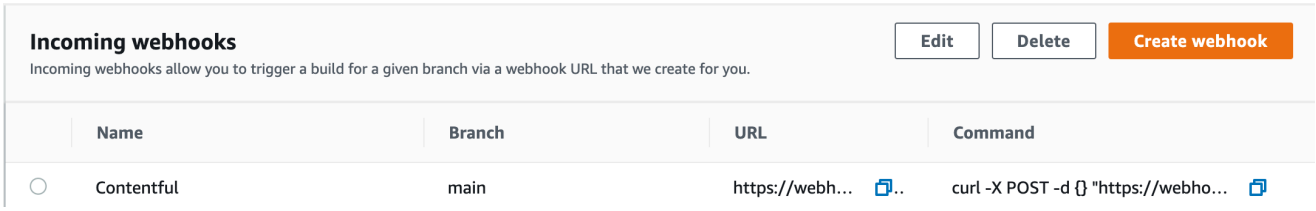
1. Inicia sesión en la consola de Amplify AWS Management Console y ábrela.
2. Elija la aplicación para la que desea crear un webhook.
3. En el panel de navegación, elija Configuración de compilación.
4. En la página Configuración de compilación, desplácese hasta la sección Webhooks entrantes y elija 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 Guardar.



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 avanzada u otro servicio para activar las compilaciones.
   - Ejecute el comando curl en una ventana de terminal para activar una nueva compilación.



Supervisión

AWS Amplify emite métricas a través de Amazon CloudWatch y proporciona registros de acceso con información detallada sobre las solicitudes realizadas a tu aplicación. Utilice los temas de esta sección para aprender a utilizar estas métricas y registros para supervisar su aplicación.

Temas

Monitorización con 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.

Métricas

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.

Métricas	Descripción
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 utilizando las solicitudes `POST` y `PUT`. 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. La estadística más relevante es `Sum`. Utilice la estadística `Sum` para obtener el número total de apariciones de estos errores.

Métricas	Descripción
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.

Puede acceder a CloudWatch las métricas AWS Management Console en https://console.aws.amazon.com/cloudwatch/. De forma alternativa, puede acceder a las métricas en la consola de Amplify mediante el siguiente procedimiento.

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.

Alarmas

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 las CloudWatch API. Por ejemplo, puede crear una alarma que le avise cuando el porcentaje de 4XXErrors supere el 15 % durante tres periodos 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 que la alarma se active.
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.

Amazon CloudWatch Logs para aplicaciones SSR

Amplify envía información sobre su tiempo de ejecución de Next.js a Amazon CloudWatch Logs en su. Cuenta de AWSAl 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 Adición de un rol de servicio. Para obtener más información acerca cómo implementar aplicaciones renderizadas en el servidor, consulte Implemente aplicaciones renderizadas del servidor con Amplify Hosting.

Registros de acceso

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 conserva todos los registros de acceso de una aplicación hasta que la elimines. Todos los registros de acceso de una aplicación están disponibles en la consola de Amplify. Sin embargo, cada solicitud individual de registros de acceso está limitada a un período de dos semanas que usted 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.

Utilice el siguiente procedimiento para recuperar los registros de acceso de una aplicación.

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, elija Configuración de la aplicación y Supervisión.
4. En la página Supervisión, elija Registros de acceso.
5. Elija Editar rango de tiempo.
6. En la ventana Editar rango de tiempo de Fecha de inicio, especifique el primer día del intervalo de dos semanas para recuperar los registros. En Fecha de inicio, elija la hora del primer día para iniciar la recuperación de los registros.
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;
```

Notificaciones

Puede configurar notificaciones para que una aplicación de AWS Amplify alerte a las partes interesadas o a los miembros del equipo de compilaciones correctas o fallidas. 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.

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. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación para la que desee configurar las notificaciones por correo electrónico.
3. En el panel de navegación, elija Configuración de aplicación, Notificaciones y, a continuación, en la sección Notificaciones por correo electrónico, elija Añadir notificación.
4. Realice una de las siguientes en la sección de Administración de notificaciones:
   - 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.
5. Cuando termine, elija Save (Guardar).

Imágenes de compilación personalizadas y actualizaciones de paquetes en directo

Temas

Imágenes de compilación personalizadas

Puedes 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.

> **Note**
>
> La Configuración de compilación es visible en el menú Configuración de aplicación de la consola Amplify solo cuando la aplicación está configurada para implementación continua y conectada a un repositorio de git. Para obtener más información sobre este tipo de implementación, consulte Introducción al código existente.

Requisitos de las imágenes de compilación personalizadas

Para que una imagen de compilación personalizada sirva como imagen de compilación de Amplify, debe cumplir los siguientes requisitos:

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 establece una anulación, la característica de anulación de paquetes de Amplify utiliza NVM para cambiar las versiones de Node.js antes de cada compilación.
2. Wget: Amplify puede usar la utilidad Wget para descargar archivos durante el proceso de compilación. Le recomendamos que la instale en su imagen personalizada.
3. Tar: Amplify puede usar la utilidad Tar para descomprimir archivos descargados durante el proceso de compilación. Le recomendamos que la instale en su imagen personalizada.

Configuración de una imagen de compilación personalizada

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. Inicie sesión en AWS Management Console y abra la consola de Amplify.
3. Elija la aplicación para la que desea configurar una imagen de compilación personalizada.

4. En el panel de navegación, elija Configuración de la aplicación, y luego Configuración de compilación.
5. En la página Configuración de compilación, en la sección Configuración de imagen de compilación, elija Editar.
6. En el cuadro de diálogo Editar imagen de compilación, abra el menú Imagen de compilación y elija Imagen de compilación.
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. Elija Save.

Actualizaciones de paquetes en directo

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.

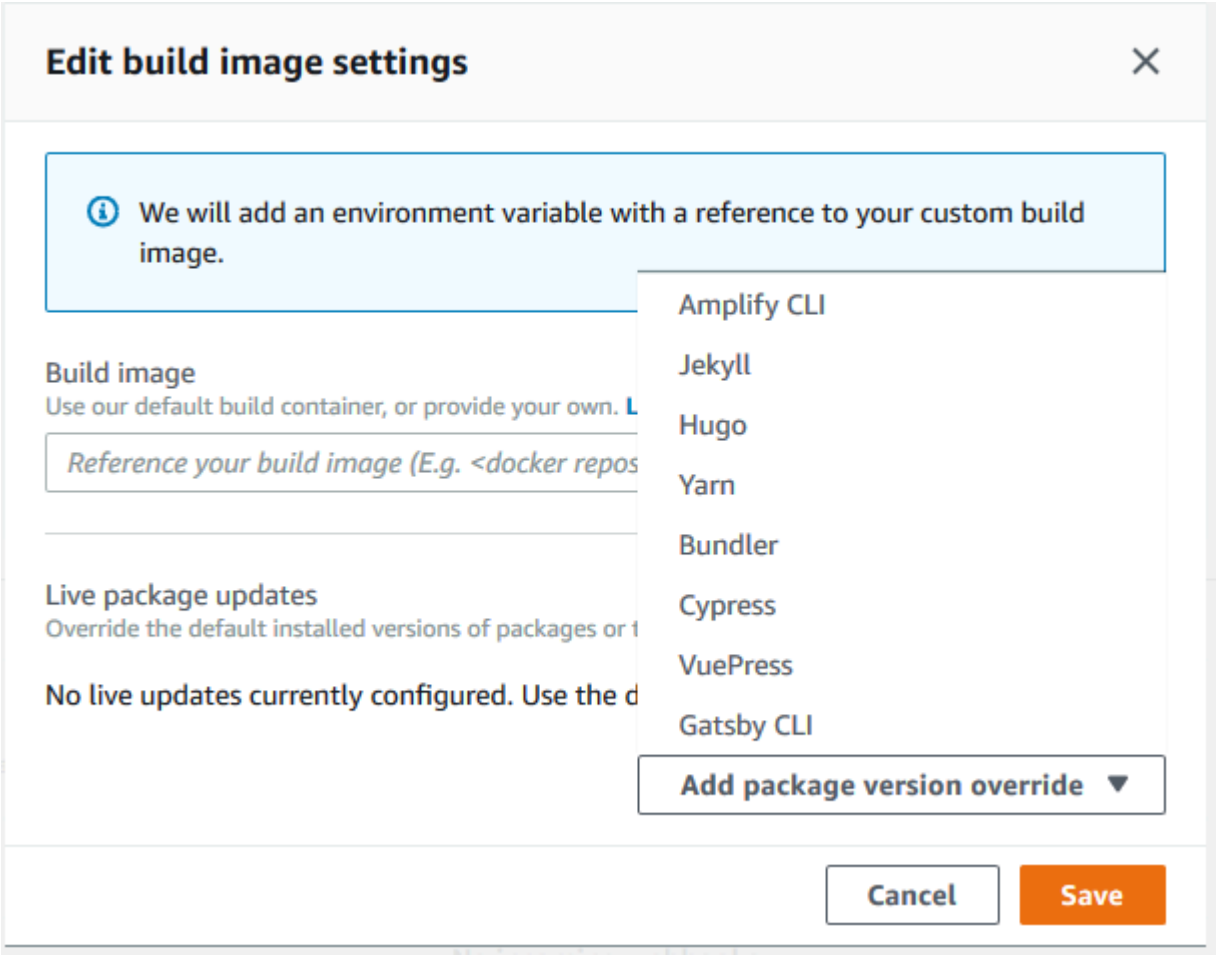
> **Warning**
>
> Si se establece la versión de Node.js en la más reciente, se producen errores en las compilaciones. En su lugar, debe especificar una versión exacta de Node.js, como `18`, `21.5` o `v0.1.2`.

Configuración de las actualizaciones de paquetes en directo

Para configurar las actualizaciones de paquetes en directo

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación para la que desea configurar las actualizaciones de paquetes en directo.
3. En el panel de navegación, elija Configuración de la aplicación, y luego Configuración de compilación.
4. En la página Configuración de compilación, en la sección Configuración de imagen de compilación, elija Editar.
5. En el cuadro de diálogo Editar configuración de imagen de compilación, abra la lista Agregar anulación de versión de paquete y elija el paquete que desea cambiar.

   La siguiente captura de pantalla muestra el cuadro de diálogo Editar configuración de imagen de compilación con la lista Agregar anulación de versión de paquete ampliada.



6. 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.
7. Elija Save.

Adición de un rol de servicio

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 de AWS Identity and Access Management (IAM) que Amplify asume cuando llama a otros servicios en su nombre. En esta guía, creará 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 cualquier recurso de Amplify Studio o CLI, y crear y administrar backends. Para obtener más información sobre Amplify Studio, consulte Cómo empezar en los documentos de Amplify. Para obtener más información sobre CLI de Amplify, consulte Amplify CLI en los documentos de Amplify.

Paso 1: Iniciar sesión en la consola de IAM

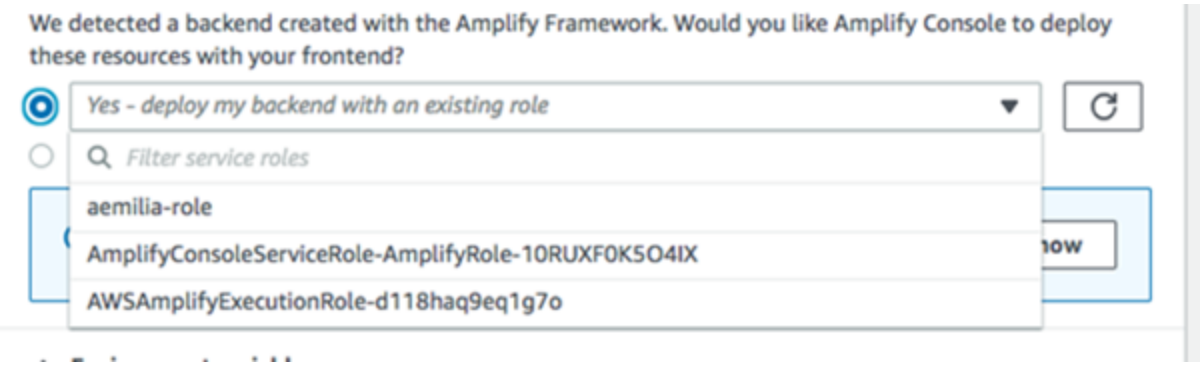
Abra la consola de IAM y elija Roles en la barra de navegación izquierda y, a continuación, Crear rol.

Paso 2: Crear un rol de Amplify

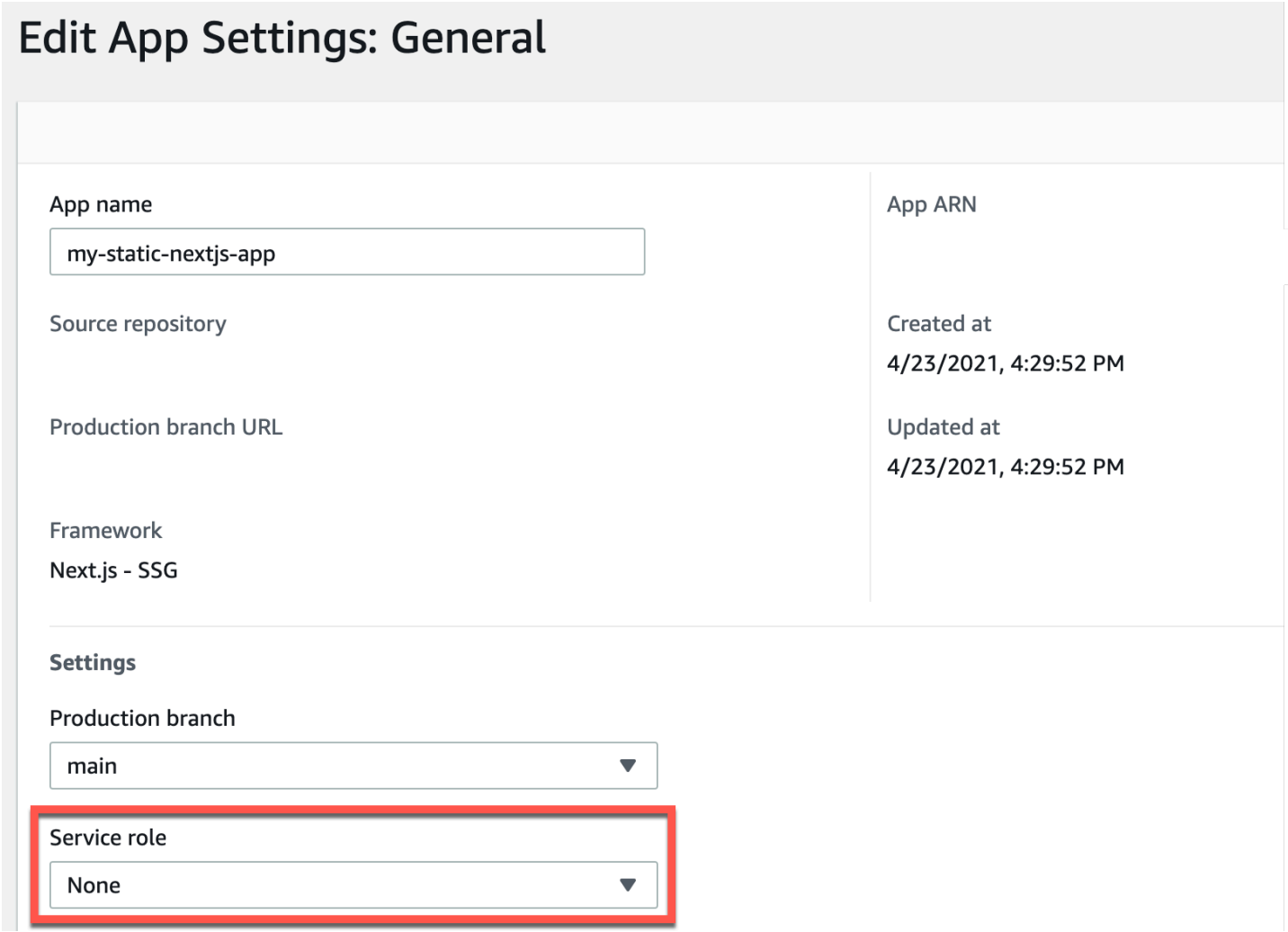
En la pantalla de selección de roles, busque Amplify y elija el rol Implementación de Amplify-Backend. Acepte todos los valores predeterminados y elija un nombre para su rol (p. ej., AmplifyConsoleServiceRole-AmplifyRole).

Paso 3: Volver a la consola de Amplify

Abra la consola de Amplify. Si está en proceso de implementar una nueva aplicación, elija actualizar y, a continuación, elija el rol que acaba de crear. Debe tener un aspecto similar al siguiente AmplifyConsoleServiceRole-AmplifyRole.



Si ya tiene una aplicación existente, encontrará la configuración del rol de servicio en Configuración de la aplicación > General. A continuación, elija Editar en la esquina superior derecha del cuadro. Elija el rol de servicio que acaba de crear a partir del menú desplegable y elija Save (Guardar).



La consola de Amplify ahora tiene permisos para implementar los recursos de backend.

Prevención del suplente confuso

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. Sustituya el texto rojo en cursiva del ejemplo por su propia 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 AWS Management Console, consulte Modificación de un rol (consola) en la Guía del usuario de IAM.

Administración del rendimiento de las aplicaciones

La arquitectura de alojamiento predeterminada de Amplify optimiza el equilibrio entre el rendimiento de alojamiento y la disponibilidad de la implementación. Para la mayoría de los clientes, le recomendamos que utilice la arquitectura predeterminada.

Para los clientes avanzados que requieren un control más preciso sobre el rendimiento de una aplicación, Amplify Hosting admite el modo de rendimiento. El modo de rendimiento permite un rendimiento de alojamiento más rápido ya que mantiene el contenido almacenado en la memoria caché en el perímetro de la red de entrega de contenido (CDN, por sus siglas en inglés) durante un intervalo más largo. Cuando el modo de rendimiento está activado, los cambios en la configuración o el código de alojamiento pueden tardar hasta 10 minutos en implementarse y encontrarse disponibles. Para obtener más información, consulte the section called “Activar el modo de rendimiento”.

Activar el modo de rendimiento

Utilice el siguiente procedimiento para activar el modo de rendimiento en una aplicación implementada en Amplify Hosting.

Para habilitar el modo de rendimiento de una aplicación

1. Inicie sesión en AWS Management Console y abra la consola de Amplify.
2. Elija la aplicación para la que desea habilitar el modo de rendimiento.
3. En el panel de navegación, elija Configuración de la aplicación y General.
4. En el panel General, desplácese hacia abajo hasta la sección Sucursales. Seleccione la sucursal para la que desee activar el modo de rendimiento.
5. Elija Acción y Habilitar modo de rendimiento.
6. En el cuadro de diálogo Habilitar modo de rendimiento, elija Habilitar modo de rendimiento.

Uso de los encabezados para controlar la duración del almacenamiento en caché

Las directivas HTTP de encabezados `Cache-Control`, `max-age` y `s-maxage` 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 el perímetro de CDN antes de que se actualice desde el servidor de origen. Las aplicaciones alojadas con Amplify respetan y reutilizan los encabezados de `Cache-Control` solicitud enviados por los clientes, a menos que se sustituyan por un encabezado personalizado que tú definas.

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).

> **Note**
>
> Cuando se activa el modo de rendimiento para una aplicación, Amplify aumenta el TTL máximo, que puedes configurar para la aplicación mediante un encabezado personalizado, de 10 minutos (600 segundos) a un día (86 400 segundos). Amplify limita el `s-maxage` que puede configurar con un encabezado personalizado en un día. Por ejemplo, si se establece `s-maxage` en una semana (604 800 segundos), Amplify utiliza el TTL máximo de un día.

Puede definir encabezados personalizados para una aplicación en la sección Encabezados personalizados de la consola de Amplify. Para ver un ejemplo del formato de YAML, consulte Ejemplo de encabezado de control de caché.

Registro de llamadas a la API de Amplify mediante AWS CloudTrail

AWS Amplify se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un servicio de AWS en Amplify. CloudTrail captura 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 un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos de Amplify. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el historial de eventos. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amplify, la dirección IP desde la que se realizó la solicitud, quién realizó la solicitud, cuándo se realizó y otros detalles adicionales.

Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Información de Amplify en CloudTrail

CloudTrail se habilita de forma predeterminada en su cuenta de AWS. Cuando se produce una actividad en Amplify, dicha actividad se registra en un evento de CloudTrail junto con los eventos de los demás servicios de AWS 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 eventos de CloudTrail en la Guía del usuario de AWS CloudTrail.

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de Amplify, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar 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. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte lo indicado en la Guía del usuario de AWS CloudTrail:

- Creación de un seguimiento para su cuenta de AWS
- Servicios e integraciones compatibles con CloudTrail

- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas las operaciones de Amplify y se documentan en la referencia de la API de la consola de AWS Amplify, la referencia de la API de Amplify Admin UI de AWS y la referencia de la API de Amplify UI Builder. Por ejemplo, las llamadas a las operaciones `CreateApp`, `DeleteApp` y `DeleteBackendEnvironment` generan entradas en los archivos de registro de CloudTrail.

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:

- ¿Se ha realizado la solicitud con las credenciales de usuario de AWS Identity and Access Management (IAM) o nodo raíz?
- Si la solicitud se ha realizado con credenciales de seguridad temporales de un rol o de un usuario federado.
- ¿Se ha realizado la solicitud con otro servicio de AWS?

Para obtener más información, consulte el elemento userIdentity de CloudTrail en la Guía del usuario de AWS CloudTrail.

Interpretación de las entradas de archivos de registro de Amplify

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos de registro de CloudTrail pueden contener una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas públicas a la API, por lo que estas no aparecen en ningún orden específico.

En el siguiente ejemplo se muestra una entrada de registro de CloudTrail que ilustra la operación de AWS Amplify referencia de la API de la consola de `ListApps`.

```
{
    "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 registro de CloudTrail que ilustra la operación de AWS Amplify referencia de la API Admin UI de `ListBackendJobs`.


```
{
    "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"
}
```

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. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos 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 ayudarlo 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 administrador 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 en que inicias sesión para AWS usar tus 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 administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo 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 cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS 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 Single Sign-On y Uso de la autenticación multifactor (MFA) en AWS 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 utilice 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, consulte 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 utilice AWS Single Sign-On. 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, consulte ¿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, consulte 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 puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo 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 permanentes a largo plazo y los roles proporcionan credenciales temporales.

Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) 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 determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM 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, puede 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 para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza 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 pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS Single Sign-On.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede 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, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común 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 ellas AWS, se te considera director. 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, consulte 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 puede 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 puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar una AWS función a una instancia EC2 y ponerla 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 a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) 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, consulte 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. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice 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 puede adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. 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 pueden 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 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. 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 pueden utilizarlos para controlar el acceso a un recurso específico.

Para el recurso al que se asocia la política, la política define qué acciones puede 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 (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL 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 que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de 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 pueden 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 puede conceder a una entidad de IAM (usuario o rol de IAM). Puede 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 especifique 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, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): las SCP 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 gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus 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 acerca de

Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.

- 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 pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte 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 AWS determinar 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
Recursos de políticas	Sí
Claves de condición de políticas	Sí
ACL	No
ABAC (etiquetas en políticas)	Parcial

Característica de IAM	Compatibilidad de Amplify
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 identidades	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 adjunto. Para 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

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. 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 pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede 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 en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones de política para Amplify

Admite acciones de política	Sí

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.

El elemento `Action` de una política JSON describe las acciones que puede 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

Admite 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 puede 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). Puede 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, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para una lista de los tipos de recursos de Amplify y sus ARN, consulte Recursos definidos por AWS Amplify en la Referencia de autorizaciones 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

Admite 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 puede 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. Puede crear expresiones condicionales que utilicen 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 lógica `AND`. 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 puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede 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, consulte 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 (ACL) de Amplify

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL 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

Admite 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 ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con Amplify

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que 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 más información sobre el cambio de roles, consulte Cambio a un rol (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 más información, consulte Credenciales de seguridad temporales en IAM.

Sesiones de acceso directo para Amplify

Admite Forward access sessions (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, consulte Reenviar sesiones de acceso.

Roles de servicio para Amplify

Compatible 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

Compatible 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 puede 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 puede 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. Para conceder permiso a los usuarios

para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles, y los usuarios pueden 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 en la Guía del usuario de IAM.

A fin de obtener más información sobre las acciones y los tipos de recursos definidos por Amplify, incluido el formato de los ARN para cada tipo de recurso, consulte Acciones, recursos y claves de condición para AWS Amplify en la Referencia de autorizaciones de servicio.

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, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía del 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 pueden 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, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo,

puede 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, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

- Utilice 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 Política de validación de Analizador de acceso de IAM en la Guía de 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 solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las 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 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. Si

eres un administrador que redacta políticas, determina los permisos correctos para conceder a los usuarios que necesiten realizar 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 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 administradas 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 gestionadas para AWS Amplify

Una política AWS administrada 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 por AWS en la Guía del usuario de IAM.

Política gestionada por AWS: AdministratorAccess -Amplify

Puede adjuntar la política de `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 de `AmplifyBackendDeployFullAccess` a las identidades de IAM.

Esta política otorga a Amplify permisos de acceso total para implementar los recursos de backend de Amplify (Amazon AWS AppSync Cognito, Amazon S3 y otros servicios relacionados) a través de. AWS Cloud Development Kit (AWS CDK) Los permisos se transfieren a los AWS CDK roles que tienen los permisos de política necesarios. `AdministratorAccess`

Para ver los permisos de esta política, consulte la Referencia AmplifyBackendDeployFullAccessde políticas AWS administradas.

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	Añada la acción `cloudformation:DeleteStack` política para permitir la eliminación de pilas cuando se	5 de abril de 2024

Cambio	Descripción	Fecha
	llame a la `DeleteBranch` API. Añada la acción `lambda:GetFunction` política para admitir las funciones de intercambio en caliente. Añada la acción `lambda:UpdateFunctionConfiguration` de política para admitir las actualizaciones de la función Lambda.	
AdministratorAccess-Amplificar: actualización de una política existente	Agregue los `cloudformation:UnTagResource` permisos `cloudformation:TagResource` y para admitir las llamadas a AWS CloudFormation las API.	4 de abril de 2024

Cambio	Descripción	Fecha
AmplifyBackendDeployFullAccess: actualización de una política actual	Añada la acción `lambda:InvokeFunction` política para apoyar 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. Añada la acción `lambda:UpdateFunctionCode` política para respaldar las funciones de intercambio en caliente.	02 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 de 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 administrador 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 administrador 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

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para 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 tus recursos a través de los Cuentas de AWS que eres propietario, consulta Cómo proporcionar acceso a un usuario de IAM en otro Cuenta de AWS de tu propiedad en la 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, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos 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.

Con fines de protección de 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:

- Utilice 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 trabajas con Amplify u otros AWS servicios mediante la consola, la API o AWS los AWS CLI SDK. 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 SSD cifrados para los puntos de presencia (POP) de ubicación perimetral y volúmenes EBS cifrados para

las cachés perimetrales regionales (REC). El código de función y la configuración de CloudFront Functions siempre se almacenan en un formato cifrado en los SSD cifrados, en los POP ubicados en las ubicaciones de borde y en otras ubicaciones de almacenamiento utilizadas por ellos. 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:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- Diseño de arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar AWS las empresas para crear aplicaciones aptas para la HIPAA.

> **Note**
>
> No Servicios de AWS todas cumplen los requisitos de la HIPAA. Para más información, consulte la Referencia de servicios compatibles con 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, consulte la Referencia de controles de Security Hub.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS consumo 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 puede 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
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde instancias de Amazon EC2 y otras fuentes. AWS CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcancen 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 AWS y servicios de oftware-as-a S-Service (SaaS), 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 puede 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 puede 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, utilice 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
Solicita fichas por segundo	Cada región admitida: 20 000	Sí	El número máximo de tokens de solicitud por segundo para una aplicación. Amplify Hosting asigna los 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 encuentra errores o problemas de implementación al trabajar con Amplify Hosting, consulte los temas de esta sección.

Temas

Solución de problemas generales de Amplify

La siguiente información puede ayudarte 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 están excediendo el tiempo de procesamiento y transferencia de datos asignado a su aplicación. Este límite de aplicaciones se gestiona mediante la cuota de servicio de `REQUEST_TOKENS_PER_SECOND` Amplify. Para obtener más información acerca de 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 el RPS de la aplicación. Por ejemplo, con los mismos 20 000 tokens, una página SSR altamente optimizada que responda en 100 milisegundos puede admitir un RPS más alto 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 un tamaño de respuesta de 250 KB.

También le recomendamos que aproveche la CloudFront caché de Amazon configurando encabezados de control de caché que maximicen el tiempo que una respuesta determinada se mantiene 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 memoria 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.

Solución de problemas de 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

- ¿Cómo ejecuto las funciones de Amplify con el motor de ejecución de Python?
- ¿Cómo ejecuto comandos que requieren privilegios de superusuario o root

¿Cómo ejecuto las funciones de Amplify con el motor 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 AL2023 tiene preinstalados enlaces simbólicos para versiones anteriores de Python. Por lo tanto, ya no necesitas actualizar los comandos de compilación en la especificación de compilación de tu aplicación siguiendo las instrucciones disponibles en las Preguntas frecuentes de Amplify Hosting GitHub .

De forma predeterminada, la versión 3.10 de Python se usa globalmente. 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
```

¿Cómo ejecuto comandos que requieren privilegios de superusuario o root

Si utiliza la imagen de compilación de Amazon Linux 2023 y recibe un error al ejecutar comandos del sistema que requieren privilegios de superusuario o root, debe ejecutar estos comandos con el `sudo` comando de Linux. Por ejemplo, si se produce un error al ejecutarse`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 AL2023 de Amplify ejecuta el código con un usuario personalizado`amplify`. Amplify otorga a este usuario privilegios para ejecutar comandos mediante el comando de Linux. `sudo` Se recomienda `sudo` utilizarlos para los comandos que requieren privilegios de superusuario.

Solución de problemas de dominios personalizados

Si tienes problemas al conectar un dominio personalizado a tu aplicación Amplify, consulta Solución de problemas de dominios personalizados para obtener ayuda.

Solución de problemas de aplicaciones renderizadas del lado del servidor

Si tienes problemas para implementar una aplicación SSR en Amplify, Resolución de problemas de las implementaciones de SSR consulta para obtener ayuda.

Referencia de alojamiento de AWS Amplify

Utilice los temas de esta sección para encontrar material de referencia detallado sobre AWS Amplify.

Temas

Compatibilidad con AWS CloudFormation

Utilice las plantillas de AWS CloudFormation para suministrar los recursos de Amplify, lo que permite implementar aplicaciones web repetibles y fiables. AWS CloudFormation proporciona un lenguaje común para que describa y suministre todos los recursos de infraestructura en el entorno en la nube y simplifica la implementación mediante varias cuentas de AWS o regiones con tan solo un par de clics.

En el caso de Amplify Hosting, consulte la documentación de Amplify CloudFormation. En el caso de Amplify Studio, consulte la documentación de Amplify UI Builder CloudFormation.

Compatibilidad con AWS Command Line Interface

Utilice AWS Command Line Interface para crear aplicaciones de 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

Puede utilizar AWS Command Line Interface para etiquetar 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 API.

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: 5 de abril de 2024

Cambio	Descripción	Fecha
Tema de políticas administr adas actualizado	Se ha actualizado el tema AWS políticas gestionad as para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	5 de abril de 2024
Tema de políticas administr adas actualizado	Se ha actualizado el tema AWS políticas gestionad as para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	4 de abril de 2024
Nuevo capítulo de solución de problemas	Se agregó el Solución de problemas de Amplify Hosting capítulo para describir cómo solucionar los problemas que surgen con las aplicaciones implementadas en Amplify Hosting.	2 de abril de 2024
Nuevo soporte para certifica dos SSL/TLS personalizados	Se agregó el Uso de certifica dos SSL/TLS tema al Configuración de dominios personalizados capítulo para describir la compatibilidad	20 de febrero de 2024

Cambio	Descripción	Fecha
	de Amplify con certificados SSL/TLS personalizados al conectar una aplicación a un dominio personalizado.	
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas gestionadas 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 agregó el tema Compatibilidad de Amplify con marcos de SSR para describir la compatibilidad de Amplify con cualquier marco de SSR basado en JavaScript con un adaptador de código abierto.	19 de noviembre de 2023
Nueva compatibilidad con el lanzamiento de la característica de optimización de imágenes	Se agregó el tema Optimización de imágenes para aplicaciones de SSR para describir la compatibilidad integrada para la optimización de imágenes para aplicaciones representadas del servidor.	19 de noviembre de 2023
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas gestionadas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	17 de noviembre de 2023

Cambio	Descripción	Fecha
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas gestionadas 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 Subdominios comodín para describir la compatibilidad con los subdominios comodín en los dominios personalizados.	6 de noviembre de 2023
Nueva política administrada	Se ha actualizado el AWS políticas gestionadas 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 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

Cambio	Descripción	Fecha
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas gestionadas 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 gestionadas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	24 de febrero de 2023
Se ha actualizado el capítulo de representación en el lado del servidor	Se ha actualizado el capítulo Implemente 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 gestionadas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	30 de agosto de 2022

Cambio	Descripción	Fecha
Tema de políticas administradas actualizado	Se ha actualizado el tema Introducción a las implementaciones continuas de pila completa 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 gestionadas 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 gestionadas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	17 de abril de 2022
Lanzamiento de una nueva función de GitHub aplicación	Se agregó el Configurar el acceso de Amplify a repositorios de GitHub tema para describir la nueva GitHub aplicación para autorizar el acceso de Amplify a GitHub tu repositorio.	5 de abril de 2022

Cambio	Descripción	Fecha
Lanzamiento de la nueva característica Amplify Studio	Se ha actualizado el tema Le damos la bienvenida 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 gestionadas 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 gestionadas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	8 de noviembre de 2021
Tema de políticas administradas actualizado	Se ha actualizado el tema AWS políticas gestionadas para AWS Amplify para describir los cambios recientes en las políticas administradas de AWS para Amplify.	27 de septiembre de 2021

Cambio	Descripción	Fecha
Nuevo tema de políticas administradas	Se agregó el AWS políticas gestionadas para AWS Amplify tema para describir las políticas AWS administr adas 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 Implemente 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 Configuración de compilaci ón de monorepo para describir cómo configurar los ajustes de compilación y la nueva variable de entorno `AMPLIFY_MONOREPO_A PP_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 para describir cómo generar automáticamente el archivo `aws-exports.js` en tiempo de compilación. Se ha agregado el tema Compilaciones de backend condicionales para describir cómo habilitar las compilaciones de backend condicionales. Se ha agregado el tema Use los backends de Amplify en todas las aplicaciones 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
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

Cambio	Descripción	Fecha
Nuevo lanzamiento de compatibilidad con la característica SSR	Se ha agregado el capítulo Implemente aplicaciones 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
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

Cambio	Descripción	Fecha
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
Lanzamiento de nueva característica de interfaz de usuario de administración	Se ha actualizado el tema Le damos la bienvenida 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

Cambio	Descripción	Fecha
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
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

Cambio	Descripción	Fecha
Nuevo tema de notificaciones	Se ha agregado el tema Notificaciones para describir cómo configurar las notificaciones 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
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.