Guía para desarrolladores

AWS App Runner

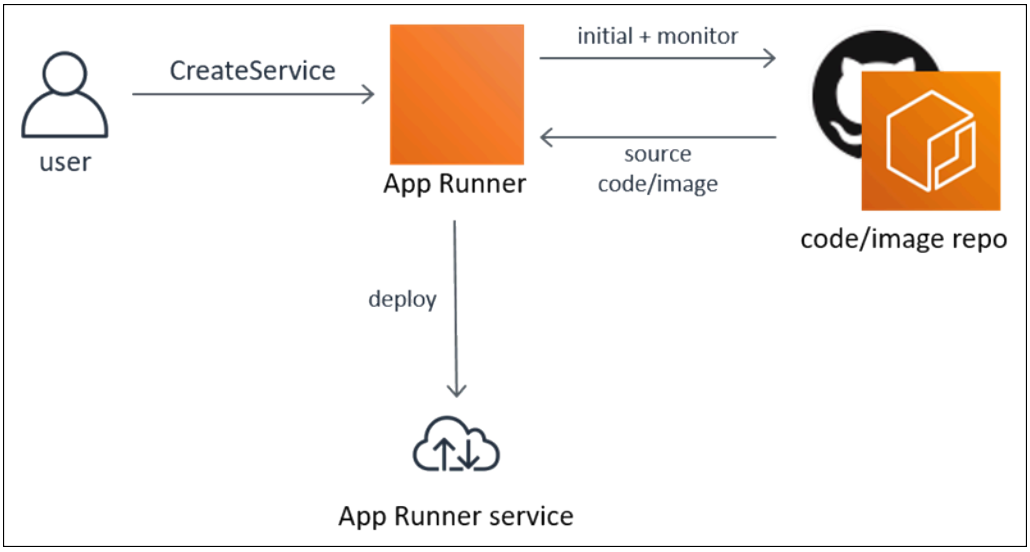
AWS App Runner: Guía para desarrolladores

Table of Contents

¿Qué es AWS App Runner?

AWS App Runneres unAWSque proporciona una forma rápida, sencilla y rentable de implementar desde el código fuente o una imagen de contenedor directamente a una aplicación web escalable y segura en elAWSLa nube. No necesita aprender nuevas tecnologías, decidir qué servicio informático usar ni saber cómo aprovisionar y configurarAWSde AWS.

App Runner se conecta directamente a su código o repositorio de imágenes. Proporciona una integración automática y una canalización de entrega con operaciones totalmente administradas, alto rendimiento, escalabilidad y seguridad.



¿Para quién es App Runner?

Si es undesarrollador, puede usar App Runner para simplificar el proceso de implementación de una nueva versión de su código o repositorio de imágenes.

ParaEquipos de operaciones, App Runner permite implementaciones automáticas cada vez que se envía una confirmación al repositorio de código o se envía una nueva versión de imagen de contenedor al repositorio de imágenes.

Acceso a App Runner

Puede definir y configurar las implementaciones del servicio App Runner mediante cualquiera de las siguientes interfaces:

- Consola de App Runner— Proporciona una interfaz web para administrar los servicios de App Runner.
- API de App Runner— Proporciona una API RESTful para realizar acciones de App Runner. Para obtener más información, consulte Referencia de la API de AWS App Runner.
- AWSInterfaz de línea de comandos deAWS CLI): proporcionan comandos para un amplio conjunto deAWS, incluido Amazon VPC, y es compatible con Windows, macOS y Linux. Para obtener más información, consulte AWS Command Line Interface.
- AWSSDK de: proporcionan API específicas de cada lenguaje y se encargan de muchos de los detalles de la conexión, tales como el cálculo de firmas, el control de reintentos de solicitud y el control de errores. Para obtener más información, consulte AWS SDK.

Precios de App Runner

App Runner proporciona una forma rentable de ejecutar su aplicación. Solo pagas por los recursos que consume tu servicio App Runner. Su servicio se reduce a menos instancias informáticas cuando el tráfico de solicitudes es más lento. Tiene control sobre la configuración de escalabilidad: el número más bajo y más alto de instancias aprovisionadas y la carga más alta que gestiona una instancia.

Para obtener más información acerca del escalado automático de App Runner, consultethe section called “Auto Scaling”.

Para obtener información sobre los precios, consulte Precios de AWS App Runner.

Pasos siguientes

Aprenda cómo empezar a utilizar App Runner en los siguientes temas:

- Configuración: complete los pasos previos para usar App Runner.
- Introducción— Implemente su primera aplicación en App Runner.

Configuración de App Runner

Si eres un nuevoAWS, complete los requisitos previos de configuración que se enumeran en esta página antes de comenzar a usarAWS App Runner.

Para estos procedimientos de configuración, utilice la herramientaAWS Identity and Access Management(IAM). Para obtener información completa sobre IAM, consulte los siguientes materiales de referencia:

- AWS Identity and Access Management (IAM)
- IAM User Guide

Crear un Cuenta de AWS

Cuando te registras conAWS, obtendrá un número de cuenta con acceso a todos los servicios queAWSOfertas, incluyendoAWS App Runner.

Si ya tiene unCuenta de AWSPara obtener información sobre el siguiente requisito previo.

Si no tiene unAWSPara crear una cuenta de.

Para inscribirse en una cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones en línea.

   Parte del procedimiento de inscripción consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Crear un usuario de IAM

Para obtener acceso aAWS, proporcione las credenciales de. Estas credenciales determinanauthentication(quién eres) yAutorización(los permisos que tiene para realizar acciones enAWSde AWS).

Cuando crea por primera vez un Amazon Web Services (AWS), comienza con una identidad de inicio de sesión única. Esa identidad tiene acceso completo a todos los servicios y recursos de AWS de

la cuenta. Esta identidad se denominaAWSaccountUsuario raíz. Cuando inicie sesión, introduzca la dirección de correo electrónico y la contraseña que utilizó para crear la cuenta.

> **Important**
>
> Le recomendamos que no utilice el usuario raíz en sus tareas cotidianas, incluso las tareas administrativas. En lugar de ello, es mejor ceñirse a la práctica recomendada de utilizar el usuario final exclusivamente para crear al primer usuario de IAM. A continuación, guarde las credenciales del usuario raíz en un lugar seguro y utilícelas tan solo para algunas tareas de administración de cuentas y servicios. Para ver las tareas que requieren que inicie sesión como usuario raíz, consulteTareas que requieren credenciales de usuario raíz.

Para obtener más información acerca del usuario raíz y las credenciales de usuario de IAM, consulteCuenta de AWScredenciales de usuario raíz y credenciales de usuario de IAMen laAWSReferencia general de.

Para crearse usted mismo un usuario administrador y agregarlo a un grupo de administradores (consola)

1. Inicie sesión en laConsola de IAMcomo propietario de la cuenta eligiendoUsuario raíze introduciendo suAWSdirección de correo electrónico de cuenta. En la siguiente página, escriba su contraseña.

   > **Note**
   >
   > Le recomendamos que siga la práctica recomendada de utilizar la**`Administrator`**Usuario de IAM que sigue y bloquea de forma segura las credenciales de usuario raíz. Inicie sesión como usuario raíz únicamente para realizar algunas tareas de administración de servicios y de cuentas.

2. En el panel de navegación, elija Users (Usuarios) y, a continuación, elija Add user (Añadir usuario).
3. En User name (Nombre de usuario), escriba **`Administrator`**.
4. Marque la casilla situada junto aAWS Management ConsoleAcceso a. A continuación, seleccione Custom password (Contraseña personalizada) y luego escriba la nueva contraseña en el cuadro de texto.

5. (Opcional) De forma predeterminada,AWSAl iniciar sesión, el nuevo usuario debe crear una nueva contraseña la primera vez que inicia sesión. Puede quitar la marca de selección de la casilla de verificación situada junto a User must create a new password at next sign-in (El usuario debe crear una nueva contraseña en el siguiente inicio de sesión) para permitir al nuevo usuario restablecer su contraseña después de iniciar sesión.
6. Seleccione Next (Siguiente): Permisos.
7. En Set permissions (Establecer persmisos), elija Add user to group (Añadir usuario a grupo).
8. Elija Create group (Crear grupo).
9. En el cuadro de diálogo Create group (Crear grupo), en Group name (Nombre del grupo) escriba **`Administrators`**.
10. SeleccionarPolíticas de filtroy, a continuación, seleccioneAWSadministrado - función de trabajopara filtrar el contenido de la tabla.
11. En la lista de políticas, active la casilla de verificación AdministratorAccess. A continuación, elija Create group (Crear grupo).

    > **Note**
    >
    > Debe activar el acceso de usuario y rol de IAM a Facturación antes de poder utilizar la`AdministratorAccess`Permisos para obtener acceso aAWS Billing and Cost Managementconsola de . Para ello, siga las instrucciones que se indican en el paso 1 del tutorial sobre cómo delegar el acceso a la consola de facturación.

12. Retroceda a la lista de grupos y active la casilla de verificación del nuevo grupo. Elija Refresh si es necesario para ver el grupo en la lista.
13. Seleccione Next (Siguiente): Tags (Etiquetas).
14. (Opcional) Añadir metadatos al rol asociando las etiquetas como pares de clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetado de entidades de IAM en la guía del usuario de IAM.
15. Seleccione Next (Siguiente): Review (Revisar)Para ver la lista de suscripciones a grupos que se agregarán al nuevo usuario. Cuando esté listo para continuar, elija Create user (Crear usuario).

Puede usar este mismo proceso para crear más grupos y usuarios y para conceder a los usuarios acceso a los recursos de la cuenta de AWS. Para obtener información sobre el uso de las políticas que restringen a los usuarios los permisos deAWSde AWS, consulteAdministración de accesosyEjemplos de políticas.

> **Important**
>
> Proteger suCuenta de AWS. Nunca envíe ni comparta sus credenciales con nadie ajeno a su organización. Nadie que represente legítimamente a Amazon le pedirá nunca sus credenciales de.

Después de crear su usuario de IAM, utilice sus credenciales para iniciar sesión en laAWS Management Console. Para obtener más información, consulteCómo inician sesión los usuarios de IAM en suCuenta de AWSen laGuía del usuario de IAM.

Crear una clave de acceso para su usuario de IAM

Las claves de acceso constan de un ID de clave de acceso y una clave de acceso secreta, que se utilizan para firmar las solicitudes de programación que se realizan aAWS. Si no tiene claves de acceso, puede crearlas desde laAWS Management Console. Como práctica recomendada, no utilice laAWSPara realizar cualquier tarea en la que no sea necesario usarlo. Por el contrario,Crear un nuevo usuario de IAMcon claves de acceso para usted.

La única vez que puede ver o descargar la clave de acceso secreta es cuando crea las claves de acceso secreta. No puede recuperarla más adelante. Sin embargo, puede crear nuevas claves de acceso en cualquier momento. Debe tener también permisos para realizar las acciones de IAM requeridas. Para obtener más información, consultePermisos obligatorios para obtener acceso a recursos de IAMen laGuía del usuario de IAM.

Para crear claves de acceso para un usuario de IAM

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Users.
3. Elija el nombre del usuario cuyas claves de acceso que desea crear y, a continuación, elija lacredenciales de seguridad dePestaña.
4. En el navegadorClaves de accesoEn la sección, elijaCrear clave de acceso.
5. Para ver el nuevo key pair de acceso, elijaMostrar. No podrá obtener acceso de nuevo a la clave de acceso secreta cuando este cuadro de diálogo se cierre. Sus credenciales tendrán el aspecto siguiente:

- ID de clave de acceso: AKIAIOSFODNN7EXAMPLE
- Clave de acceso secreta: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

6. Para descargar el par de claves, elija Download .csv file (Descargar archivo .csv). Almacene las claves en un lugar seguro. No podrá obtener acceso de nuevo a la clave de acceso secreta cuando este cuadro de diálogo se cierre.

   Mantenga la confidencialidad de las claves para proteger suAWScuenta y no las envíe nunca por correo electrónico. No las comparta fuera de su organización, aunque reciba una petición que parezca provenir de AWS o Amazon.com. Nadie que represente legítimamente a Amazon le pedirá nunca su clave secreta.

7. Después de descargar el`.csv`, elijaClose. Cuando cree una clave de acceso, el par de claves se activa de forma predeterminada, y puede utilizar el par de inmediato.

Temas relacionados

- ¿Qué es IAM?en laGuía del usuario de IAM
- AWSCredenciales de seguridad deinAWSReferencia general de

Pasos siguientes

Ha completado los pasos previos. Para implementar la primera aplicación en App Runner, consulteIntroducción.

Introducción a App Runner

AWS App Runneres unAWSque proporciona una forma rápida, sencilla y rentable de convertir una imagen de contenedor existente o código fuente directamente en un servicio web en ejecución en elNube de AWS.

En este tutorial se explica cómo se puede utilizarAWS App Runnerpara implementar la aplicación en un servicio App Runner. Recorre la configuración del código fuente y la implementación, la compilación del servicio y el tiempo de ejecución del servicio. También muestra cómo implementar una versión de código, realizar un cambio de configuración y ver registros. Por último, el tutorial muestra cómo limpiar los recursos que ha creado mientras sigue los procedimientos del aprendizaje.

Temas

Prerequisites

Antes de iniciar el tutorial, asegúrese de que realiza las siguientes acciones:

1. Realice los pasos que se indican enConfiguración.
2. Creación de unGitHubSi todavía no tiene una cuenta de. Si es la primera vez que utiliza GitHub, consulteIntroducción a GitHubenlaDocumentos de GitHub.
3. Cree un repositorio en su cuenta de GitHub. En este tutorial se utiliza el nombre del repositorio`python-hello`. Cree archivos en el directorio raíz del repositorio, con los nombres y el contenido especificados en los siguientes ejemplos.

Archivos para el**`python-hello`**repositorio de ejemplo

Example requirements.txt

```
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.1
Werkzeug==0.15.5
```

Example server.py

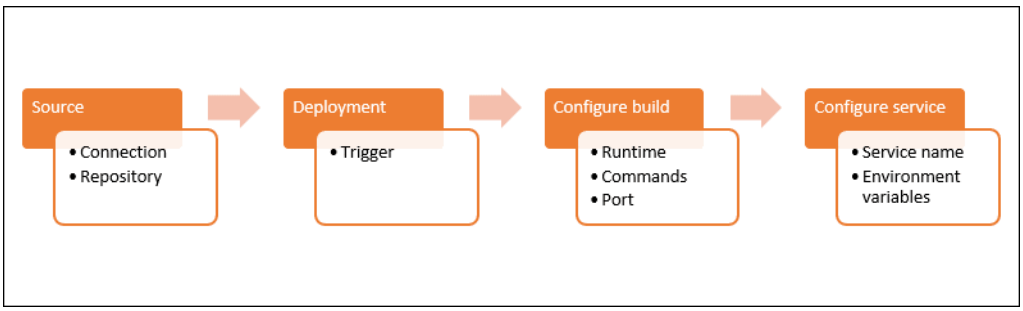
```
from flask import Flask
import os

PORT = 8080
name = os.environ['NAME']
if name == None or len(name) == 0:
  name = "world"
MESSAGE = "Hello, " + name + "!"
print("Message: '" + MESSAGE + "'")

app = Flask(__name__)


@app.route("/")
def root():
  print("Handling web request. Returning message.")
  result = MESSAGE.encode("utf-8")
  return result


if __name__ == "__main__":
  app.run(debug=True, host="0.0.0.0", port=PORT)
```

Paso 1: Creación de un servicio App Runner

En este paso, creará un servicio App Runner basado en el repositorio de código fuente de ejemplo que creó en GitHub como parte dethe section called “Prerequisites”. El ejemplo contiene un sitio web de Python simple. Estos son los pasos principales que debe seguir para crear un servicio:

1. Configure el código fuente.
2. Configurar la implementación de origen.
3. Configurar la compilación de aplicaciones.
4. Configure su servicio.
5. Revisar y confirmar.

El siguiente diagrama describe los pasos para crear un servicio de App Runner:

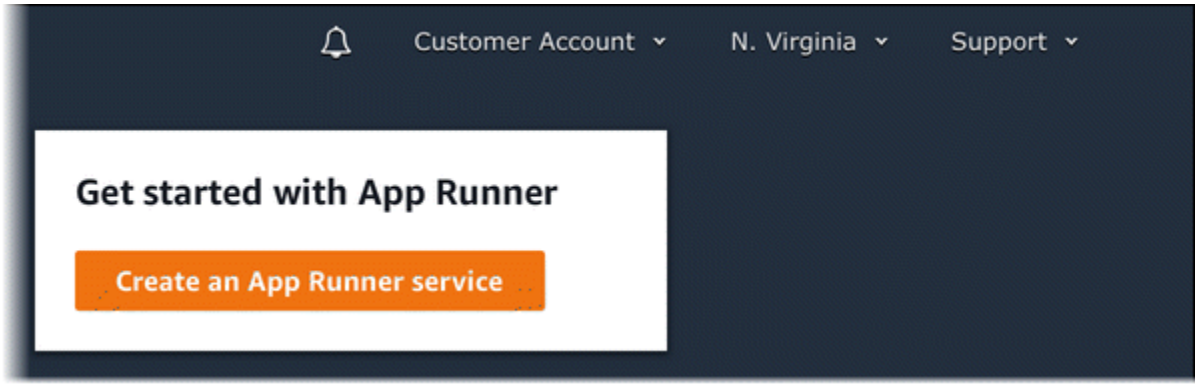


Para crear un servicio App Runner basado en un repositorio de código fuente

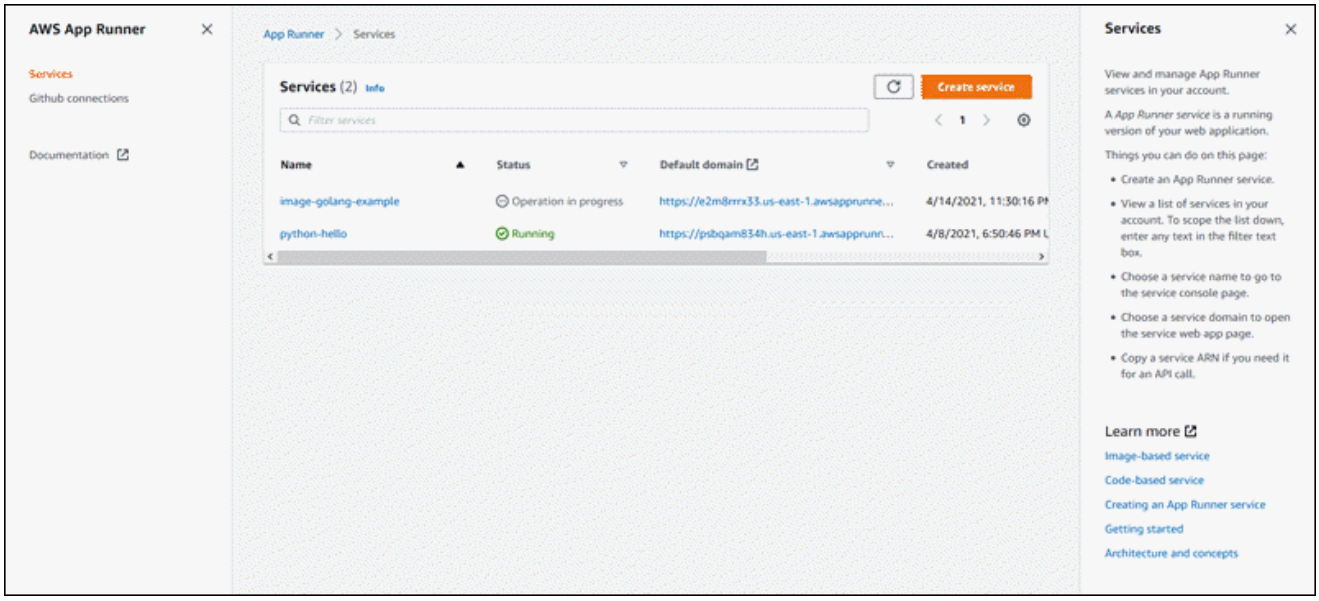
1. Configure el código fuente.

   a. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.

   b. Si el archivo deCuenta de AWSaún no tiene ningún servicio de App Runner, se muestra la página de inicio de la consola. SeleccionarCreación de un servicio App Runner.



   Si el archivo deCuenta de AWStiene servicios existentes, elServiciosAparecerá una lista de sus servicios. Elija Create service.



c. En la páginaOrigen e implementación, en laFuentesección, paraRepository type, elijaRepositorio de código fuente.

d. UNDERConnect a GitHubelijaAgregar nuevoEn y, a continuación, proporcione sus credenciales de GitHub.

> **Note**
>
> Los pasos que se indican a continuación para instalar elAWSConector para GitHub a tu cuenta de GitHub son pasos de una sola vez. Puede reutilizar la conexión para crear varios servicios de App Runner basados en repositorios de esta cuenta. Cuando tenga una conexión existente, selecciónela y salte a la selección del repositorio.

e. En el navegadorInstallAWSConector para GitHub, si se le solicita, elija el nombre de su cuenta de GitHub.

f. Si se le pide que autorice elAWSConector para GitHub, elijaAutorice las conexiones de AWS.

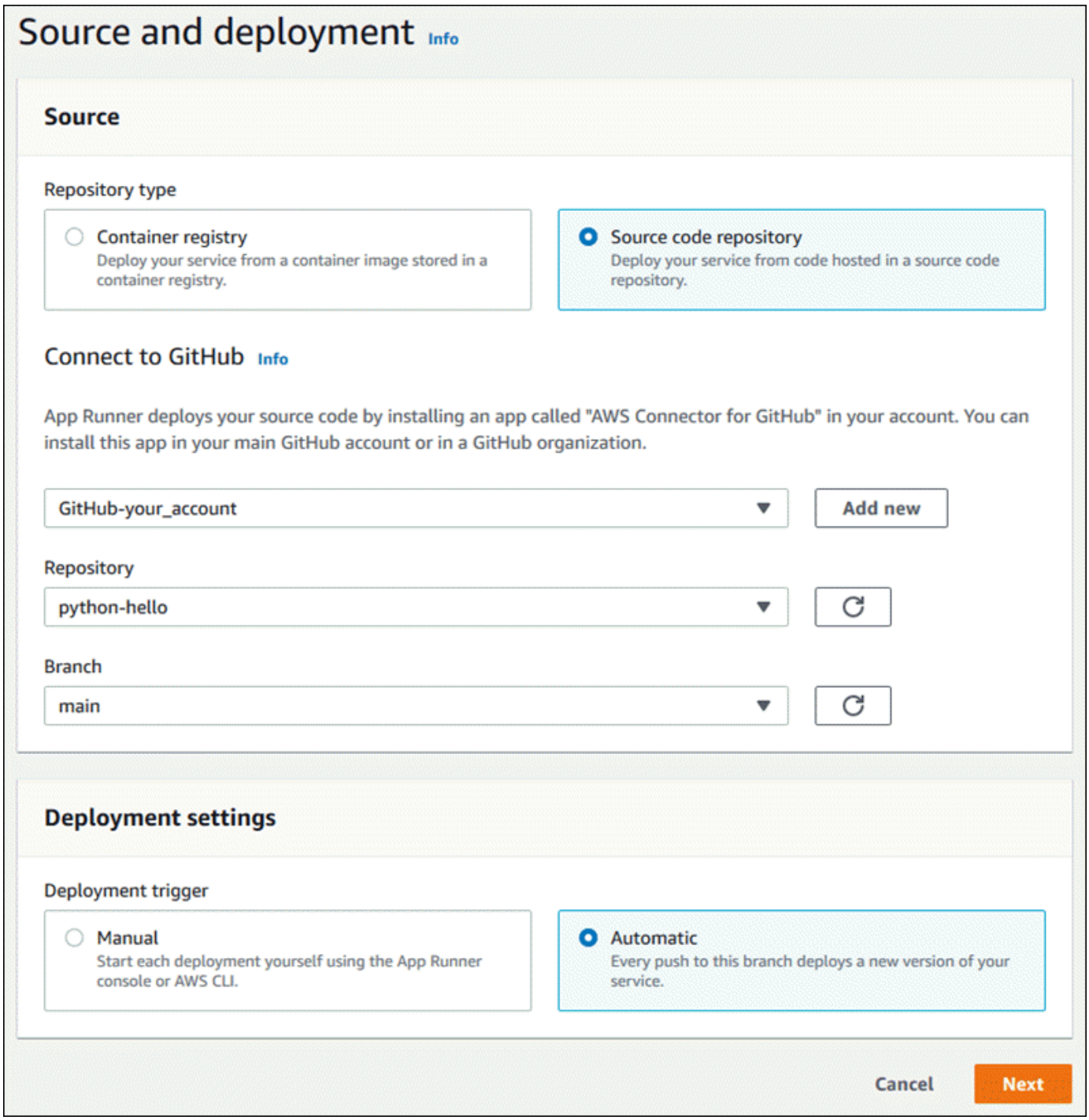
g. Elija Instalar.

El nombre de su cuenta aparece comoCuenta/organización de GitHub. Ahora puede elegir un repositorio de su cuenta.

h. ParaRepositorio, elija el repositorio de ejemplo que creó,`python-hello`. ParaCrear ramificacionesEn, elija el nombre de la rama predeterminado de su repositorio (por ejemplo,Principal).

2. Configure sus implementaciones: En el navegadorConfiguración de la implementaciónsección, elijaAutomáticoelija y luego seleccioneSiguiente.

> **Note**
>
> Con la implementación automática, cada nueva confirmación en su repositorio implementa automáticamente una nueva versión de su servicio.



3. Configurar la compilación de aplicaciones.

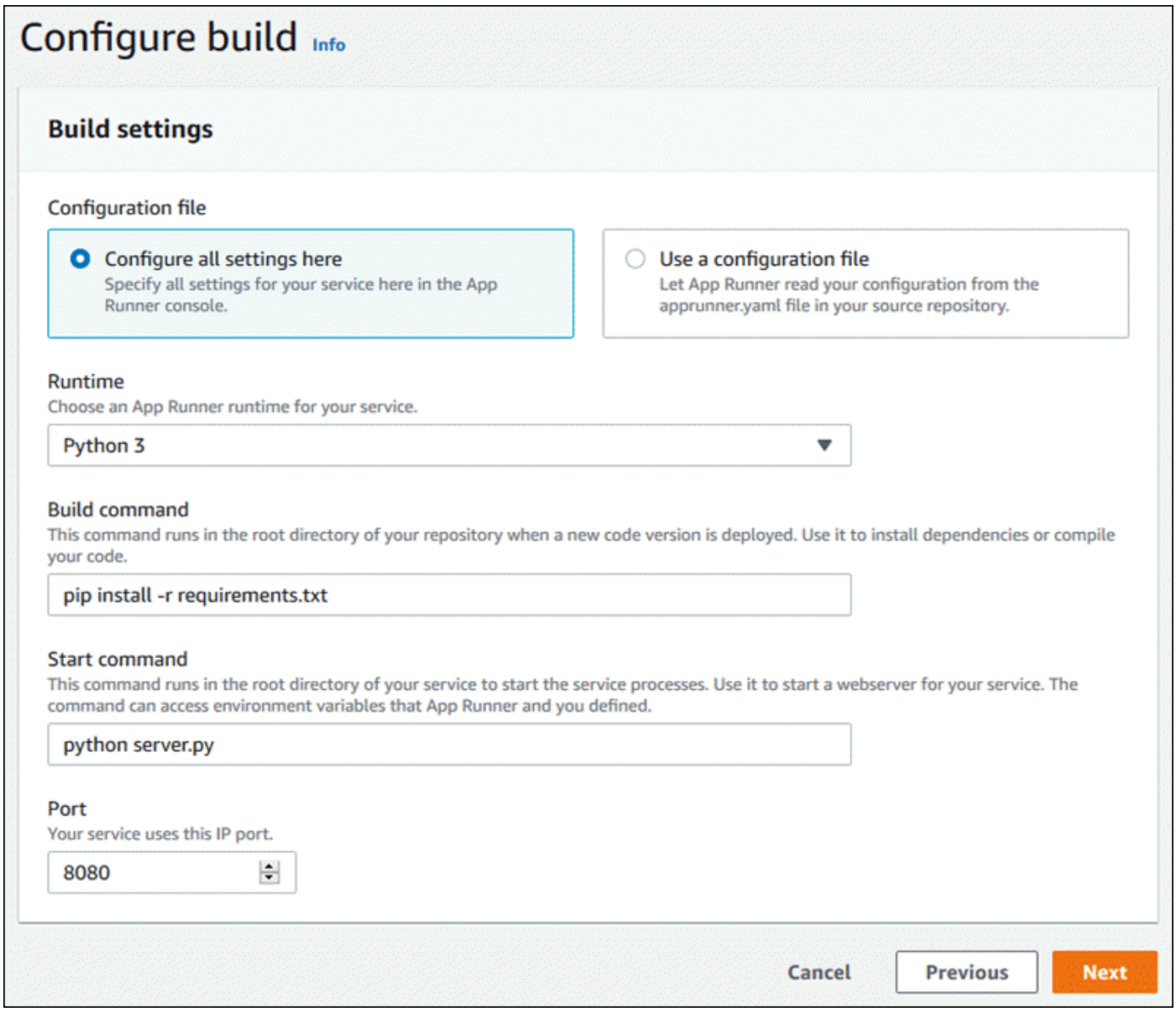
   a. En la páginaConfigurar la compilaciónPágina, paraArchivo de configuración, elijaConfigurar todas las opciones aquí.

   b. Proporcione la siguiente configuración de compilación:

- Runtime (Tiempo de ejecución):— ElegirPython 3.
- Comando de compilación— Escriba**`pip install -r requirements.txt`**.
- Comando de inicio— Escriba**`python server.py`**.
- Puerto— Escriba**`8080`**.

c. Seleccione Next (Siguiente).

> **Note**
>
> El tiempo de ejecución de Python 3 construye una imagen Docker usando una imagen base de Python 3 y su código Python de ejemplo. A continuación, inicia un servicio que ejecuta una instancia de contenedor de esta imagen.



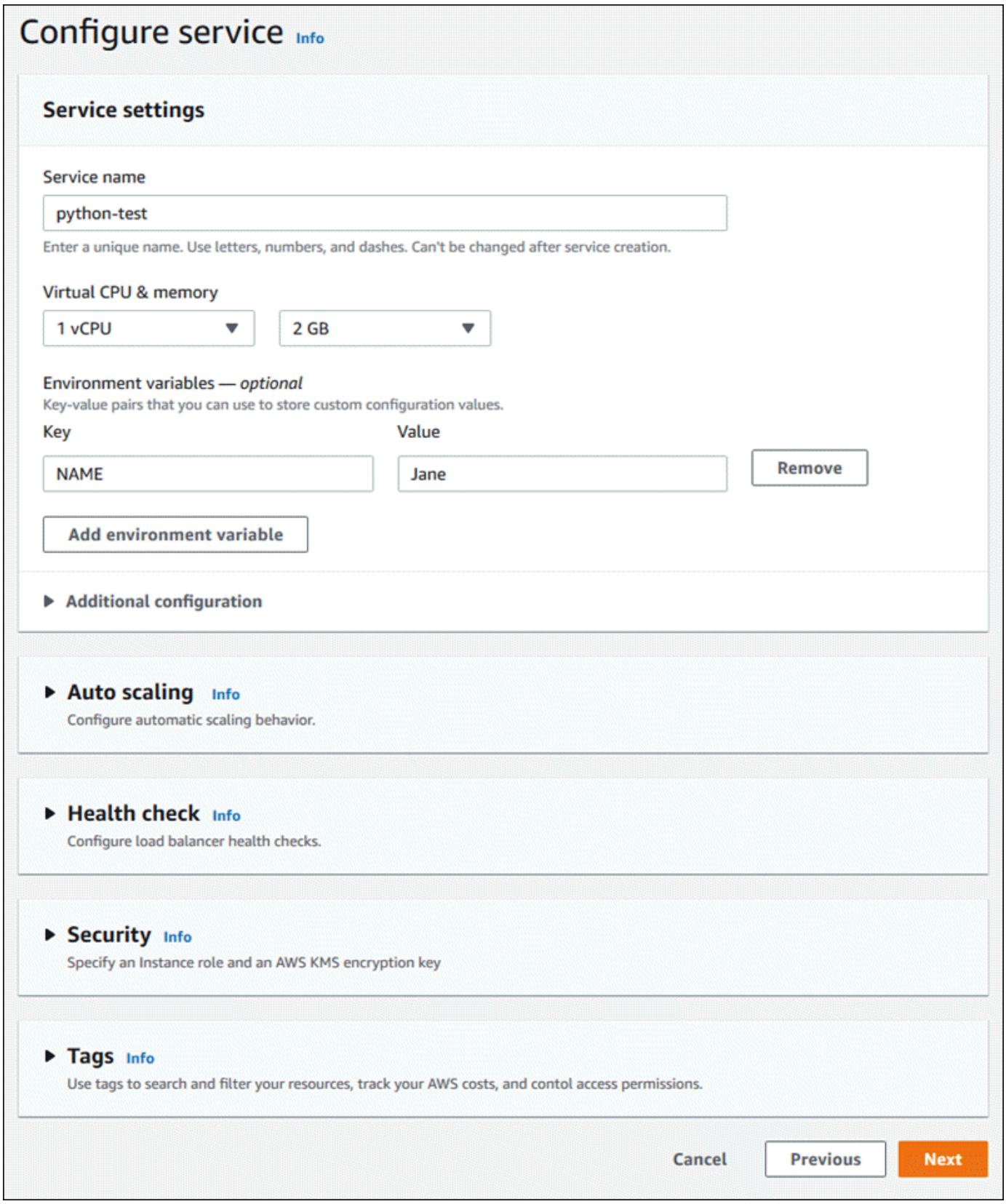
4. Configure su servicio.

   a. En la páginaConfigure el servicio, en laConfiguración del servicioEn, escriba un nombre de servicio.

   b. UNDEREnvironment variables (Variables de entorno):, agregue una única variable de entorno. ParaClave deintroduzca, introduzca**`NAME`**, y paraValorEn, escriba cualquier nombre (por ejemplo, su nombre de pila).

> **Note**
>
> La aplicación de ejemplo lee el nombre establecido en esta variable de entorno y muestra el nombre en su página web.

c. Seleccione Next (Siguiente).


Configure service Info
Service settings
Service name
python-test
Enter a unique name. Use letters, numbers, and dashes. Can't be changed after service creation.
Virtual CPU & memory
1 vCPU
2 GB
Environment variables — optional
Key-value pairs that you can use to store custom configuration values.
Key
Value
NAME
Jane
Remove
Add environment variable
Additional configuration
Auto scaling Info
Configure automatic scaling behavior.
Health check Info
Configure load balancer health checks.
Security Info
Specify an Instance role and an AWS KMS encryption key
Tags Info
Use tags to search and filter your resources, track your AWS costs, and contol access permissions.
Cancel
Previous
Next

5. En la páginaRevisar y crear, verifique todos los detalles que haya introducido y, a continuación, elijaCreación e implementación.

   Si el servicio se crea correctamente, la consola muestra el panel de servicio, con unInformación general del serviciodel nuevo servicio.

6. Compruebe que su servicio está en ejecución.

   a. En la página del panel de servicio, espere hasta que el servicioEstadoesEn ejecución.

   b. Elija el iconoDominio predeterminadovalor: es la URL del sitio web de su servicio.

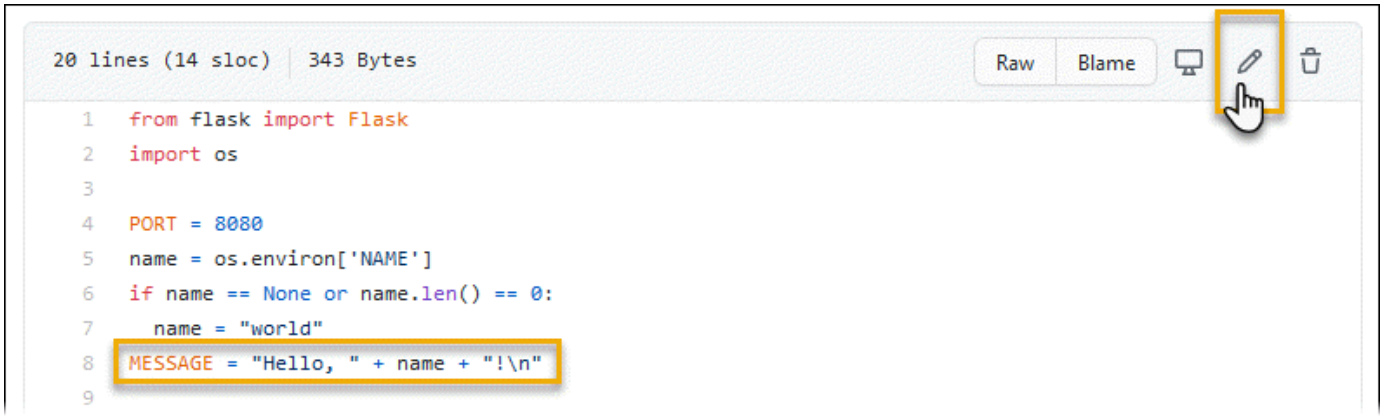
      Una página web muestra: Hola,***Su nombre***!

Paso 2: Cambiar el código de servicio

En este paso, se realiza un cambio en el repositorio de código fuente de. La capacidad de App Runner CI/CD crea e implementa automáticamente el cambio en su servicio.

Para realizar un cambio en el código de servicio

1. Vaya al repositorio de GitHub de ejemplo.
2. Elija el nombre del archivo`server.py`para navegar hasta ese archivo.
3. SeleccionarEditar este archivo(el icono del lápiz).
4. En la expresión asignada a la variable`MESSAGE`, cambie el texto`Hello`De a`Good morning`.



5. Seleccione Commit changes (Confirmar cambios).

   La nueva confirmación comienza a implementarse. En la página del panel de servicio, el servicioEstadocambios enOperación en curso.

6. Espere a que termine la implementación. En la página del panel de servicio, el servicioEstadodebería cambiar de nuevo aEn ejecución.

7. Compruebe que la implementación se ha realizado correctamente: actualice la pestaña del navegador donde se muestra la página web de su servicio.

   La página muestra ahora el mensaje modificado: Buenos días.***Su nombre***!

Paso 3: Realizar un cambio de configuración

En este paso, se realiza un cambio en el**NAME**, para demostrar un cambio en la configuración del servicio.

Para ver los registros del servicio

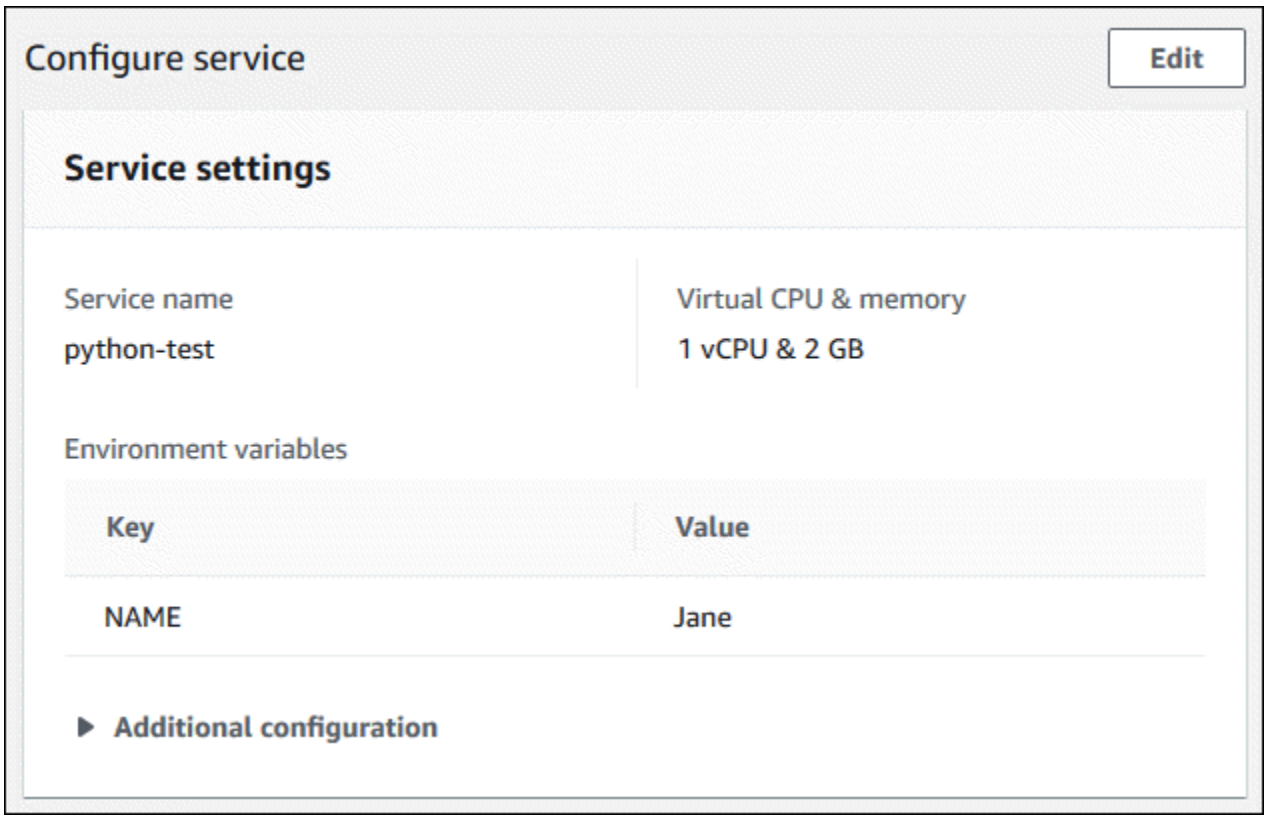
1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, elijaServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.

3. En la página del panel de servicios, seleccione la opciónConfiguraciónPestaña.

   La consola muestra los ajustes de configuración del servicio en varias secciones.

4. En el navegadorConfigure el serviciosección, elijaEditar.



5. Para la variable de entorno con la clave**NAME**Cambie el valor a un nombre diferente.
6. Elija Apply changes.

   App Runner inicia el proceso de actualización. En la página del panel de servicio, el servicioEstadocambios enOperación en curso.
7. Espere a que finalice la actualización. En la página del panel de servicio, el servicioEstadodebería cambiar de nuevo aEn ejecución.
8. Compruebe que la actualización se ha realizado correctamente: actualice la pestaña del navegador donde se muestra la página web de su servicio.

   La página muestra ahora el nombre modificado: Buenos días.***Nombre nuevo***!

Paso 4: Ver registros de su servicio

En este paso, utilice la consola de App Runner para ver los registros del servicio App Runner. App Runner transmite los registros a los registros de Amazon CloudWatch Logs (CloudWatch Logs) y los muestra en el panel de control de su servicio. Para obtener información acerca de los registros de App Runner, consultethe section called “Registros (registros de CloudWatch Logs)”.

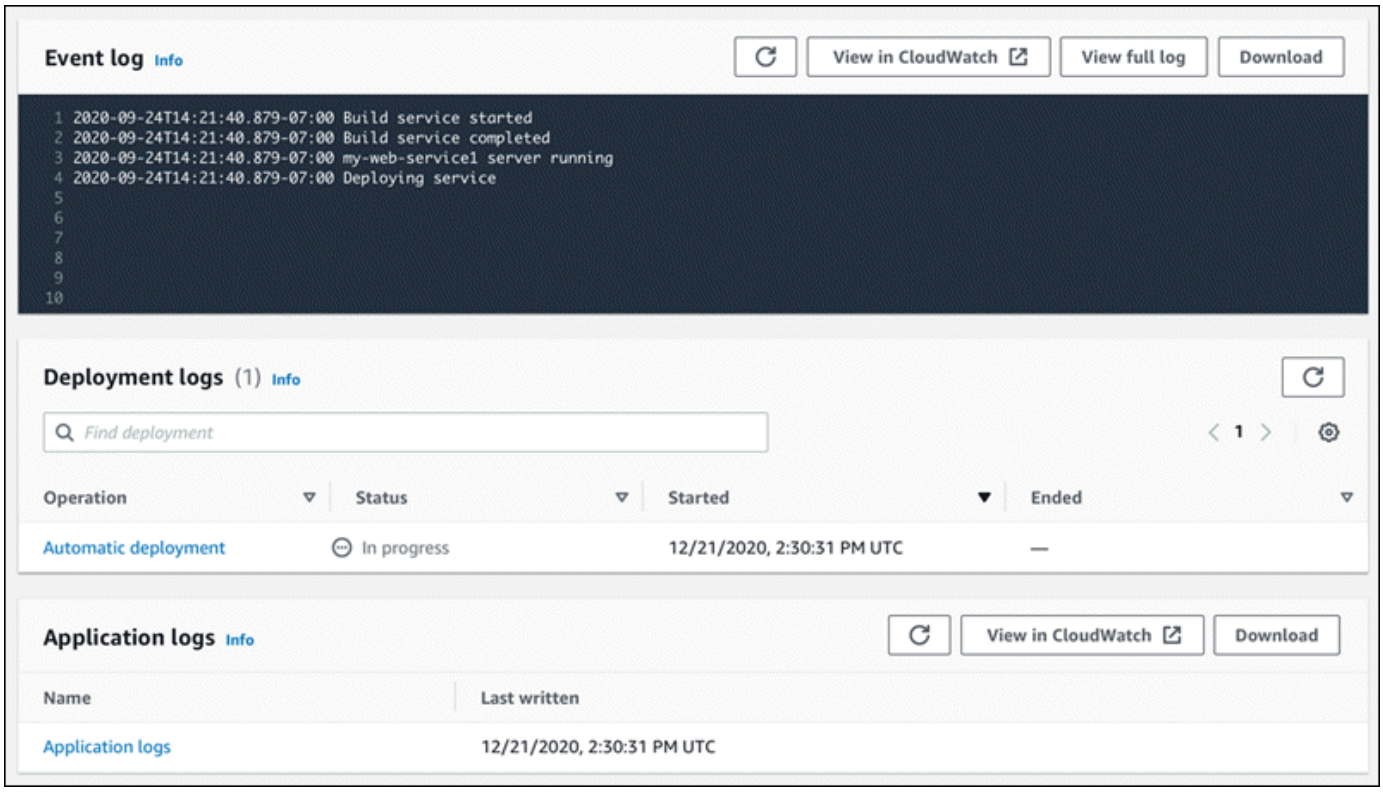
Para ver los registros del servicio

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, elijaServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.
3. En la página del panel de servicios, seleccione la opciónRegistrosPestaña.

   La consola muestra algunos tipos de registros en varias secciones:

   - Registro de eventos— Actividad en el ciclo de vida de tu servicio App Runner. La consola muestra los eventos más recientes.
   - Registros de implementación— Implementaciones de repositorio de origen en su servicio App Runner. La consola muestra una secuencia de registro independiente para cada implementación.
   - Logs de aplicaciones— El resultado de la aplicación web que se implementa en el servicio App Runner. La consola combina la salida de todas las instancias en ejecución en un único flujo de registro.



4. Para buscar implementaciones específicas, introduzca un término de búsqueda en la lista de registros de implementación. Puede buscar cualquier valor que aparezca en la tabla.
5. Para ver el contenido de un registro, elijaVer registro completo(registro de eventos) o el nombre del flujo de registro (registros de implementación y aplicación).
6. SeleccionarDescargarpara descargar un registro. Para una secuencia de registro de implementación, seleccione primero una secuencia de registro.
7. SeleccionarVer en CloudWatchpara abrir la consola de CloudWatch y utilizar todas sus capacidades para explorar los registros de servicio de App Runner. Para una secuencia de registro de implementación, seleccione primero una secuencia de registro.

> **Note**
>
> La consola de CloudWatch es especialmente útil si desea ver los registros de aplicaciones de instancias específicas en lugar del registro combinado de aplicaciones.

Paso 5: Eliminar recursos

Ahora has aprendido a crear un servicio de App Runner, ver registros y realizar algunos cambios. En este paso, elimine el servicio para quitar recursos que ya no necesite.

Para eliminar el servicio

1. En la página del panel de servicios, elijaActionselija y luego seleccioneEliminación del servicio.
2. En el cuadro de diálogo de confirmación, escriba el texto solicitado y, a continuación, elijaEliminar.

   Resultado: La consola navega a laServicios(Se ha creado el certificado). El servicio que acaba de eliminar muestra un estado deDELETING. Poco tiempo después desaparece de la lista.

Considere también eliminar la conexión de GitHub que creó como parte de este tutorial. Para obtener más información, consulte the section called “Conexiones”.

Siguientes pasos

Ahora que ha implementado su primer servicio App Runner, obtenga más información en los siguientes temas:

- Arquitectura y conceptos— La arquitectura, los conceptos principales yAWSrecursos relacionados con App Runner.
- Servicio basado en la imagenyServicio basado en código de— Los dos tipos de origen de aplicaciones que App Runner puede implementar.
- Desarrollo para App Runner: cosas que debe saber al desarrollar o migrar código de aplicación para su implementación a App Runner.
- Consola de App Runner: gestiona y supervisa tu servicio mediante la consola de App Runner.
- Administrar su servicioAdministre el ciclo de vida de su servicio App Runner.
- Registro y monitoreo: supervise su servicio App Runner mediante la visualización de métricas, la lectura de registros y el seguimiento de las llamadas de acción del servicio.
- Archivo de configuración de App Runner— Una forma basada en la configuración de especificar opciones para el comportamiento de compilación y tiempo de ejecución del servicio App Runner.
- App Runner APIUtilice la interfaz de programación de aplicaciones (API) de App Runner para crear, leer, actualizar y eliminar recursos de App Runner.
- Seguridad— Las diferentes formas en queAWSy garantiza la seguridad en la nube mientras usa App Runner y otros servicios.

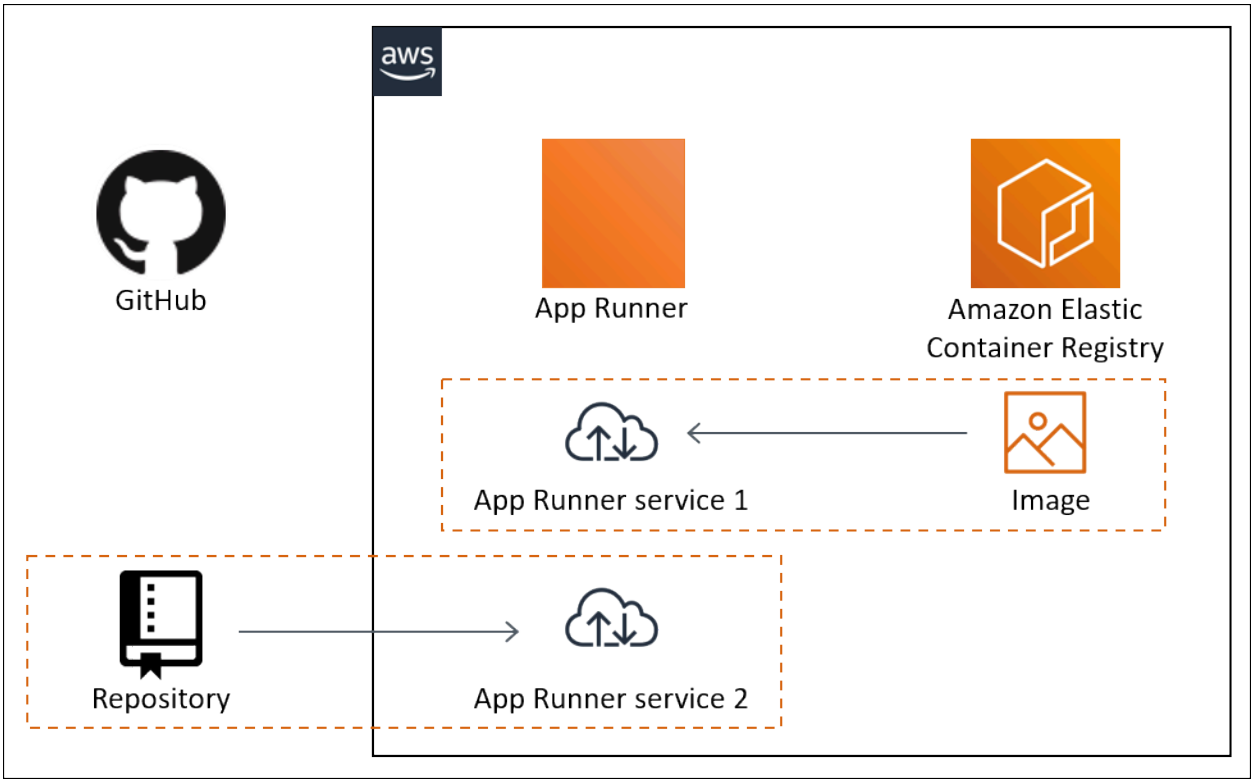
Arquitectura y conceptos de App Runner

AWS App Runnertoma el código fuente o la imagen fuente de un repositorio, y crea y mantiene un servicio web en ejecución para usted en elNube de AWS. Por lo general, debe llamar a una sola acción de App Runner,CreateService, para crear el servicio.

Con un repositorio de imágenes de origen, proporciona una imagen de contenedor lista para usar que App Runner puede implementar para ejecutar su servicio web. Con un repositorio de código fuente, puede proporcionar código e instrucciones para crear y ejecutar un servicio web diseñado para uno de los varios entornos de tiempo de ejecución administrados por App Runner.

En este momento, App Runner puede recuperar su código fuente de unGitHub, o recuperar su imagen de origen desdeAmazon Elastic Container Registry (Amazon ECR)En suCuenta de AWS.

En el diagrama siguiente se muestra información general de la arquitectura del servicio App Runner. En el diagrama, hay dos servicios de ejemplo: uno implementa el código fuente de GitHub y el otro implementa una imagen de origen de Amazon ECR.



Conceptos de App Runner

Los siguientes son conceptos clave relacionados con el servicio web que se ejecuta en App Runner:

- Servicio App Runner— UnAWSque App Runner utiliza para implementar y administrar la aplicación en función de su repositorio de código fuente o imagen de contenedor. Un servicio App Runner es una versión en ejecución de su aplicación. Para obtener más información acerca de la creación de un servicio, consultethe section called “Creación”.
- Tipo de origen— El tipo de repositorio de origen que proporciona para implementar el servicio App Runner:Código fuente de .orimagen de origen.
- Proveedor de repositorio— El servicio de repositorio que contiene el origen de la aplicación (por ejemplo,GitHuborAmazon ECR).
- Conexión de App Runner— UnAWSque permite a App Runner acceder a una cuenta de proveedor de repositorio (por ejemplo, una cuenta u organización de GitHub). Para obtener más información acerca de las conexiones, consulte the section called “Conexiones”.
- Runtime (Tiempo de ejecución):— Una imagen base para implementar un repositorio de código fuente. App Runner ofrece una variedad deTiempos de ejecuciónpara diferentes entornos de programación. Para obtener más información, consulte Servicio basado en código de.
- Deployment: acción que aplica una versión del repositorio de origen (código o imagen) a un servicio de App Runner. La primera implementación en el servicio se produce como parte de la creación del servicio. Las implementaciones posteriores pueden producirse de una de las dos formas siguientes:
  - Implementación automática— Una capacidad de CI/CD. Puede configurar un servicio App Runner para compilar automáticamente (para el código fuente) e implementar cada versión de la aplicación tal y como aparece en el repositorio. Esto puede ser una nueva confirmación en un repositorio de código fuente o una nueva versión de imagen en un repositorio de imágenes de origen.
  - Implementación manual— Una implementación en el servicio App Runner que inicia explícitamente.
- Dominio Personalizado— Un dominio que asocias con el servicio App Runner. Los usuarios de su aplicación web pueden utilizar este dominio para acceder a su servicio web en lugar del subdominio de App Runner predeterminado. Para obtener más información, consulte the section called “Nombres de dominio personalizados”.
- Mantenimiento: actividad que App Runner realiza ocasionalmente en la infraestructura que ejecuta el servicio App Runner. Cuando el mantenimiento está en curso, el estado del servicio cambia temporalmente a`OPERATION_IN_PROGRESS`(Operación en cursoEn la consola) durante unos minutos. Las acciones de su servicio (por ejemplo, implementación, actualización de configuración, pausa/reanudación o eliminación) se bloquean durante este tiempo. Vuelva a intentar la acción unos minutos más tarde, cuando el estado del servicio vuelva a`RUNNING`.

> **Note**
>
> Si tu acción falla, no significa que tu servicio App Runner esté inactivo. Su aplicación está activa y mantiene el manejo de solicitudes. Es poco probable que su servicio experimente algún tiempo de inactividad.

En particular, App Runner migra su servicio si detecta problemas en el hardware subyacente que aloja el servicio. Para evitar cualquier tiempo de inactividad del servicio, App Runner implementa el servicio en un nuevo conjunto de instancias y desplaza el tráfico hacia ellas (una implementación azul-verde). Ocasionalmente, es posible que veas un ligero aumento temporal de los cargos.

Recursos de App Runner

Cuando usa App Runner, crea y administra algunos tipos de recursos en suCuenta de AWS. Estos recursos se utilizan para acceder a su código y administrar sus servicios.

En la siguiente tabla se ofrece una descripción general de estos recursos:

Nombre del recurso	Descripción
Service	Representa una versión en ejecución de la aplicación. Gran parte del resto de esta guía describe los tipos de servicio, la administración, la configuración y la supervisión. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:service/` *`service-name`* `[/`*`service-id`* `]`
Connection	Proporciona a tus servicios de App Runner acceso a repositorios privados almacenados con proveedores externos. Existe como un recurso independiente para compartir entre varios servicios. Para obtener más información acerca de las conexiones, consulte the section called “Conexiones”. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:connection/` *`connection-name`* `[/`*`connection-id`* `]`

Nombre del recurso	Descripción
AutoScalingConfiguration	Proporciona a los servicios de App Runner configuraciones que controlan el escalado automático de la aplicación. Existe como un recurso independiente para compartir entre varios servicios. Para obtener más información acerca de la escala automática, consultethe section called “Auto Scaling”. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`*` :autoscalingconfiguration/` *`config-name`* `[/`*`config-revision`* `[/`*`config-id`* `]]`

Cuotas de recursos

AWSimpone algunas cuotas (también conocidas como límites) en su cuenta paraAWSUso de recursos en cadaRegión de AWS. En la siguiente tabla se muestran las cuotas relacionadas con los recursos de App Runner. Las cuotas también se describen enAWS App RunnerCuotas y puntos de enlace deen laAWSReferencia general de.

Cuotas de recursos	Descripción	Valor predeterminado	¿Ajustable?
Services	Número máximo de servicios que puede crear en su cuenta para cadaRegión de AWS.	10	✓ Sí
Connections	Número máximo de conexiones que puede crear en su cuenta para cadaRegión de AWS. Puede compartir una sola conexión a través de varios servicios.	10	✓ Sí
Auto scaling configurations—nombres	El número máximo de nombres únicos que puede tener en las configuraciones de escalado automático que cree en su cuenta para cadaRegión de AWS. Puede utilizar una configuración de escalado automático en varios servicios.	10	✓ Sí

Cuotas de recursos	Descripción	Valor predeterminado	¿Ajustable?
Auto scaling configurations—revisiones para cada nombre	Número máximo de revisiones de configuración de escalado automático que puede crear en su cuenta para cadaRegión de AWSPara cada nombre único. Puede utilizar una revisión de configuración de escalado automático en varios servicios.	10	✗ No

La mayoría de las cuotas son ajustables y puede solicitar un aumento de cuota para ellas. Para obtener más información, consulteSolicitud de aumento de cuotaEn la Guía del usuario de Service Quotas.

Servicio App Runner basado en una imagen de origen

Puede usarAWS App Runnerpara crear y administrar servicios basados en dos tipos de fuente de servicios fundamentalmente diferentes:Código fuente de .yimagen de origen. Independientemente del tipo de origen, App Runner se encarga de iniciar, ejecutar, escalar y equilibrar la carga del servicio. Puede utilizar la capacidad CI/CD de App Runner para realizar un seguimiento de los cambios en su imagen o código fuente. Cuando App Runner descubre un cambio, se construye automáticamente (para el código fuente) e implementa la nueva versión en el servicio App Runner.

En este capítulo se tratan los servicios basados en una imagen de origen. Para obtener información acerca de los servicios basados en el código fuente, consulteServicio basado en código de.

Aimagen de origenes una imagen de contenedor pública o privada almacenada en un repositorio de imágenes. Apunta App Runner a una imagen e inicia un servicio que ejecuta un contenedor basado en esta imagen. No es necesaria ninguna etapa de compilación. Más bien, proporciona una imagen lista para implementar.

Proveedores de repositorio de

App Runner es compatible con los siguientes proveedores de repositorio de imagen:

- Amazon Elastic Container Registry (Amazon ECR)— Almacena imágenes privadas en tuCuenta de AWS.
- Amazon Elastic Container Registry Public: almacena imágenes legibles públicamente.

Implementación desde Amazon ECR

Amazon ECRalmacena imágenes en repositorios. Hay repositorios privados y públicos. Para implementar la imagen en un servicio App Runner desde un repositorio privado, App Runner necesita permiso para leer la imagen desde Amazon ECR. Para conceder ese permiso a App Runner, debes proporcionar a App Runner unFunción de acceso a. Este es unAWS Identity and Access ManagementEl rol de (IAM) que tiene los permisos de acción de Amazon ECR necesarios. Cuando utilice la consola de App Runner para crear el servicio, puede elegir un rol existente en su cuenta. Como alternativa, puede usar la consola de IAM para crear un nuevo rol personalizado o elegir que la consola de App Runner cree un rol basado en directivas administradas.

Cuando utiliza la API de App Runner o laAWS CLI, puede completar un proceso de dos pasos. En primer lugar, puede utilizar la consola de IAM para crear un rol de acceso a. Puede usar una directiva administrada que App Runner proporciona o especificar sus propios permisos personalizados. A continuación, se proporciona el rol de acceso durante la creación del servicio mediante el comandoCreateServiceAcción de la API.

Para obtener información acerca de la creación del servicio App Runner, consultethe section called "Creación".

Implementación desde Amazon ECR Public

Amazon ECR Publicalmacena imágenes legibles públicamente. Estas son las principales diferencias entre Amazon ECR y Amazon ECR Public que debes tener en cuenta en el contexto de los servicios de App Runner:

- Las imágenes públicas de Amazon ECR son legibles públicamente. No es necesario que proporcione un rol de acceso al crear un servicio basado en una imagen pública de Amazon ECR.
- App Runner no admite la implementación automática de imágenes públicas de Amazon ECR.

Servicio App Runner basado en el código fuente

Puede usarAWS App Runnerpara crear y administrar servicios basados en dos tipos de fuente de servicios fundamentalmente diferentes:Código fuente de .yimagen de origen. Independientemente del tipo de origen, App Runner se encarga de iniciar, ejecutar, escalar y equilibrar la carga del servicio. Puede utilizar la capacidad CI/CD de App Runner para realizar un seguimiento de los cambios realizados en la imagen o el código fuente. Cuando App Runner descubre un cambio, se construye automáticamente (para el código fuente) e implementa la nueva versión en el servicio App Runner.

En este capítulo se tratan los servicios basados en el código fuente. Para obtener información acerca de los servicios basados en una imagen de origen, consulteServicio basado en la imagen.

El código fuente es el código de aplicación que App Runner crea e implementa para usted. Usted apunta App Runner a un repositorio de código fuente y elige unruntime. App Runner crea una imagen basada en la imagen base del motor de ejecución y el código de la aplicación. A continuación, inicia un servicio que ejecuta un contenedor basado en esta imagen.

App Runner proporciona un práctico idioma específicoTiempos de ejecución dirigidos. Cada uno de estos tiempos de ejecución crea una imagen de contenedor a partir de su código fuente y agrega dependencias de tiempo de ejecución del lenguaje a su imagen. No es necesario proporcionar instrucciones de configuración de contenedor ni de compilación como Dockerfile.

Los subtemas de este capítulo discuten los diversosTiempos de ejecución de .que es compatible con App Runner, el tiempo de ejecución de Dockerfile genérico y elTiempos de ejecución dirigidospara diferentes entornos de programación.

Temas

Proveedores de repositorio de código

App Runner implementa el código fuente al leerlo desde un repositorio de código fuente. App Runner admite un proveedor de repositorio de código fuente: GitHub.

Implementación desde GitHub

Para implementar el código fuente en un servicio de App Runner desdeGitHub, App Runner establece una conexión con GitHub. Si su repositorio es privado (es decir, no es accesible públicamente en GitHub), debe proporcionar a App Runner detalles de conexión. Cuando utilice la consola de App Runner paraCrear un servicioProporciona detalles de la conexión como parte del procedimiento de creación de servicios.

Cuando utiliza la API de App Runner o laAWS CLI, una conexión es un recurso independiente. En primer lugar, debe crear la conexión utilizando la herramientaCreateConnectionAcción de la API. A continuación, se proporciona el ARN de la conexión durante la creación del servicio mediante el comandoCreateServiceAcción de la API.

Para obtener más información sobre la creación de servicios de App Runner, consulte, consultethe section called “Creación”. Para obtener más información sobre las conexiones de App Runner, consulte, consultethe section called “Conexiones”.

Tiempo de ejecución administrado de App Runner

App Runner proporciona tiempos de ejecución administrados para diversos entornos de programación. Cada tiempo de ejecución administrado facilita la creación y ejecución de contenedores basados en un lenguaje de programación determinado o entorno de tiempo de ejecución. Cuando se utiliza un tiempo de ejecución administrado, App Runner comienza con una imagen de tiempo de ejecución administrado. Esta imagen se basa en elimagen de Docker Linux de Amazony contiene un paquete de tiempo de ejecución de lenguaje, así como algunas herramientas y paquetes de dependencias populares. App Runner utiliza esta imagen de tiempo de ejecución administrado como imagen base y agrega el código de la aplicación para crear una imagen Docker. A continuación, implementa esta imagen para ejecutar el servicio web en un contenedor.

Especifique un tiempo de ejecución para el servicio App Runner cuandoCrear un serviciomediante la consola de App Runner o elCreateServiceAPI DE. También puede especificar un tiempo de ejecución como parte del código fuente. Usar`runtime`Palabra clave enArchivo de configuración de App Runnerque incluya en su repositorio de código. La convención de nomenclatura de un tiempo de ejecución administrado es*`<language-name> <major-version>`*.

App Runner actualiza el tiempo de ejecución de su servicio a la versión más reciente en cada implementación o actualización de servicio. Si la aplicación requiere una versión específica de un tiempo de ejecución administrado, puede especificarla mediante la herramienta`runtime-version`Palabra clave enArchivo de configuración de App Runner. Especifique una versión

secundaria como*<major>*.*<minor>*para bloquear las versiones principales y secundarias (App Runner actualiza solo las versiones de parches). Especifique un nivel de parche concreto como*<major>*.*<minor>*.*<patch>*para bloquear el servicio en una versión de tiempo de ejecución específica (App Runner nunca actualiza el tiempo de ejecución).

Uso del tiempo de ejecución administrado de Python

AWS App Runnerproporciona un tiempo de ejecución administrado de Python. El tiempo de ejecución le facilita la tarea de crear y ejecutar contenedores con aplicaciones web basadas en Python. Cuando utiliza el motor de ejecución de Python, App Runner comienza con una imagen de tiempo de ejecución de Python administrada. Esta imagen se basa en elImagen de Docker Linux de Amazony contiene el paquete de tiempo de ejecución de Python y algunas herramientas y paquetes de dependencias populares. App Runner utiliza esta imagen de tiempo de ejecución administrado como imagen base y agrega el código de la aplicación para crear una imagen Docker. A continuación, implementa esta imagen para ejecutar el servicio web en un contenedor.

Especifique un tiempo de ejecución para el servicio App Runner cuandoCrear un serviciomediante la consola de App Runner o elCreateServiceAPI. También puede especificar un tiempo de ejecución como parte del código fuente. Usar`runtime`Palabra clave clave en unArchivo de configuración de App Runnerque incluya en su repositorio de código. La convención de nomenclatura de un tiempo de ejecución administrado es*<language-name> <major-version>*.

Para obtener nombres válidos de tiempo de ejecución de Python, consultethe section called “Información de la publicación”.

App Runner actualiza el tiempo de ejecución de su servicio a la versión más reciente en cada implementación o actualización de servicio. Si la aplicación requiere una versión específica de un tiempo de ejecución administrado, puede especificarla mediante la herramienta`runtime-version`Palabra clave clave deArchivo de configuración de App Runner. Especifique una versión secundaria como*<major>*.*<minor>*para bloquear las versiones principales y secundarias (App Runner actualiza solo las versiones de parches). Especifique un nivel de parche concreto como*<major>*.*<minor>*.*<patch>*para bloquear el servicio en una versión de tiempo de ejecución específica (App Runner nunca actualiza el tiempo de ejecución).

Temas

- Configuración de tiempo de ejecución de Python
- Ejemplos de tiempo de ejecución de Python
- Información de versión de tiempo de ejecución de Python

Configuración de tiempo de ejecución de Python

Cuando elige un tiempo de ejecución administrado, también debe configurar, como mínimo, comandos de compilación y ejecución. Los configura mientrascreatingoractualizartu servicio App Runner. Existen varias maneras de hacerlo:

- Uso de la consola de App Runner— Especifique los comandos de laConfigurar la compilaciónde la pestaña de proceso de creación o configuración.
- Uso de la API de App Runner— Llame aCreateServiceorUpdateService. Especifique los comandos mediante el comando`BuildCommand`y`StartCommand`miembros de laCodeConfigurationValuesTipo de datos.
- Uso de unaArchivo de configuración: especifique uno o más comandos de compilación en un máximo de tres fases de compilación y un único comando de ejecución que sirve para iniciar la aplicación. Hay opciones de configuración adicionales opcionales.

Proporcionar un archivo de configuración es opcional. Al crear un servicio de App Runner mediante la consola o la API, se especifica si App Runner obtiene los ajustes de configuración directamente durante la creación o desde un archivo de configuración.

Ejemplos de tiempo de ejecución de Python

Los siguientes ejemplos muestran los archivos de configuración de App Runner para crear y ejecutar un servicio de Python. El último ejemplo es el código fuente de una aplicación completa de Python que puede implementar en un servicio de tiempo de ejecución de Python.

Archivo de configuración de Python

En este ejemplo se muestra un archivo de configuración mínimo que se puede utilizar con el motor de ejecución administrado de Python. Para conocer las suposiciones que App Runner hace con un archivo de configuración mínimo, consultethe section called “Archivos de configuración”.

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    build:
```

```
      - pip install pipenv
      - pipenv install
run:
  command: python app.py
```

Archivo de configuración de Python extendido

Este ejemplo muestra el uso de todas las claves de configuración con el tiempo de ejecución administrado de Python.

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    pre-build:
      - wget -c https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/test-lib.tar.gz -O - | tar
 -xz
    build:
      - pip install pipenv
      - pipenv install
    post-build:
      - python manage.py test
  env:
    - name: DJANGO_SETTINGS_MODULE
      value: "django_apprunner.settings"
    - name: MY_VAR_EXAMPLE
      value: "example"
run:
  runtime-version: 3.7.7
  command: pipenv run gunicorn django_apprunner.wsgi --log-file -
  network:
    port: 8000
    env: MY_APP_PORT
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
```

Origen de aplicación de Python de extremo a extremo

Este ejemplo muestra el código fuente de una aplicación completa de Python que puede implementar en un servicio de tiempo de ejecución de Python.

Example requirements.txt

```
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.1
Werkzeug==0.15.5
```

Example server.py

```
from flask import Flask
import os

PORT = 8080
name = os.environ['NAME']
if name == None or len(name) == 0:
  name = "world"
MESSAGE = "Hello, " + name + "!"
print("Message: '" + MESSAGE + "'")

app = Flask(__name__)


@app.route("/")
def root():
  print("Handling web request. Returning message.")
  result = MESSAGE.encode("utf-8")
  return result


if __name__ == "__main__":
  app.run(debug=True, host="0.0.0.0", port=PORT)
```

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    build:
      - pip install -r requirements.txt
```

```
run:
  command: python server.py
```

Información de versión de tiempo de ejecución de Python

En este tema se enumeran todos los detalles de las versiones en tiempo de ejecución de Python que admite App Runner.

Python 3

Detalle	Descripción
Nombre del runtime	Python3
Versiones secundarias	3.7, 3.8
Paquetes incluidos	Python, pip, herramientas de configuración, rueda, virtualenv

Uso del tiempo de ejecución administrado Node.js

AWS App Runnerproporciona un tiempo de ejecución administrado Node.js. El tiempo de ejecución le facilita la tarea de crear y ejecutar contenedores con aplicaciones web basadas en Node.js. Cuando utiliza el tiempo de ejecución Node.js, App Runner comienza con una imagen de tiempo de ejecución Node.js administrada. Esta imagen se basa en elImagen de Docker Linuxy contiene el paquete de tiempo de ejecución Node.js y algunas herramientas. App Runner utiliza esta imagen de tiempo de ejecución administrado como imagen base y agrega el código de la aplicación para crear una imagen Docker. A continuación, implementa esta imagen para ejecutar el servicio web en un contenedor.

Especifique un tiempo de ejecución para el servicio App Runner cuandoCrear un serviciomediante la consola de App Runner o elCreateServiceAPI. También puede especificar un tiempo de ejecución como parte del código fuente. Usar`runtime`Palabra clave enArchivo de configuración de App Runnerque incluya en su repositorio de código. La convención de nomenclatura de un tiempo de ejecución administrado es*<language-name> <major-version>*.

Para obtener nombres válidos en tiempo de ejecución de Node.js, consultethe section called “Información de la versión”.

App Runner actualiza el tiempo de ejecución de su servicio a la versión más reciente en cada implementación o actualización de servicio. Si la aplicación requiere una versión específica de un tiempo de ejecución administrado, puede especificarla mediante la herramienta`runtime-version`Palabra clave deArchivo de configuración de App Runner. Especifique una versión secundaria como*<major>*.*<minor>*para bloquear las versiones principales y secundarias (App Runner actualiza solo las versiones de parches). Especifique un nivel de parche concreto como*<major>*.*<minor>*.*<patch>*para bloquear el servicio en una versión de tiempo de ejecución específica (App Runner nunca actualiza el tiempo de ejecución).

Temas

- Configuración de tiempo de ejecución Node.js
- Ejemplos de tiempo de ejecución Node.js
- Información de la versión de tiempo de ejecución Node.js

Configuración de tiempo de ejecución Node.js

Cuando elige un tiempo de ejecución administrado, también debe configurar, como mínimo, comandos de compilación y ejecución. Los configura mientrascreatingoractualizartu servicio App Runner. Existen varias maneras de hacerlo:

- Uso de la consola de App Runner— Especifique los comandos en el cuadro de diálogoConfigurar la compilaciónde la pestaña de proceso de creación o configuración.
- Uso de la API de App RunnerLlame aCreateServiceorUpdateService. Especifique los comandos mediante el comando`BuildCommand`y`StartCommand`miembros de laCodeConfigurationValuesTipo de datos.
- Uso de unaArchivo de configuración: especifique uno o más comandos de compilación en un máximo de tres fases de compilación y un único comando de ejecución que sirve para iniciar la aplicación. Hay opciones de configuración adicionales opcionales.

Proporcionar un archivo de configuración es opcional. Al crear un servicio de App Runner mediante la consola o la API, se especifica si App Runner obtiene los ajustes de configuración directamente durante la creación o desde un archivo de configuración.

Con el tiempo de ejecución Node.js específicamente, también puede configurar la compilación y el tiempo de ejecución utilizando un archivo JSON llamado`package.json`en la raíz de su repositorio de código fuente. Con este archivo, puede configurar la versión del motor Node.js, los paquetes

de dependencias y varios comandos (aplicaciones de línea de comandos). Los administradores de paquetes como npm o yarn interpretan este archivo como entrada para sus comandos.

Por ejemplo:

- npm installinstala los paquetes definidos por eldependenciesydevDependenciesNodopackage.json.
- npm startornpm run startejecuta el comando definido por el comandoscripts/startNodopackage.json.

A continuación se muestra un ejemplo de un archivo package.json.

package.json

```
{
  "name": "node-js-getting-started",
  "version": "0.3.0",
  "description": "A sample Node.js app using Express 4",
  "engines": {
    "node": "12.18.4"
  },
  "scripts": {
    "start": "node index.js",
    "test": "node test.js"
  },
  "dependencies": {
    "cool-ascii-faces": "^1.3.4",
    "ejs": "^2.5.6",
    "express": "^4.15.2"
  },
  "devDependencies": {
    "got": "^11.3.0",
    "tape": "^4.7.0"
  }
}
```

Para obtener más información acerca depackage.json, consulteLa guía package.jsonen el sitio web de Node.js.

Tips

- Si las recetaspackage.jsondefine un archivostart, puede usarlo comorunen el archivo de configuración de App Runner, como se muestra en el siguiente ejemplo.

  Example

  package.json

  ```
  {
    "scripts": {
      "start": "node index.js"
    }
  }
  ```

  apprunner.yaml

  ```
  run:
    command: npm start
  ```

- Cuando ejecutanpm installen su entorno de desarrollo, npm crea el archivopackage-lock.json. Este archivo contiene una instantánea de las versiones del paquete npm que acaba de instalar. A partir de entonces, cuando npm instala dependencias, utiliza estas versiones exactas. Del mismo modo, hilo creayarn.json. Confirme estos archivos en su repositorio de código fuente para asegurarse de que su aplicación está instalada con las versiones de dependencias con las que desarrolló y probó.
- También puede utilizar un archivo de configuración de App Runner para configurar la versión Node.js y el comando start. Cuando lo haga, estas definiciones anulan las depackage.json. Un conflicto entre elnodeversión enpackage.jsony laruntime-versionen el archivo de configuración de App Runner hace que la fase de compilación de App Runner falle.

Ejemplos de tiempo de ejecución Node.js

En los ejemplos siguientes se muestran los archivos de configuración de App Runner para crear y ejecutar un servicio Node.js.

Archivo de configuración de Node.js

En este ejemplo se muestra un archivo de configuración mínimo que se puede utilizar con el motor de ejecución administrado Node.js. Para conocer las suposiciones que App Runner hace con un archivo de configuración mínimo, consultethe section called “Archivos de configuración”.

Example apprunner.yaml

```
version: 1.0
runtime: nodejs12
build:
  commands:
    build:
      - npm install --production
run:
  command: node app.js
```

Archivo de configuración Node.js

En este ejemplo se muestra el uso de todas las claves de configuración con el tiempo de ejecución administrado Node.js.

Example apprunner.yaml

```
version: 1.0
runtime: nodejs12
build:
  commands:
    pre-build:
      - npm install --only=dev
      - node test.js
    build:
      - npm install --production
    post-build:
      - node node_modules/ejs/postinstall.js
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
run:
  runtime-version: 12.18.4
  command: node app.js
  network:
    port: 8000
```

```
    env: APP_PORT
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
```

Aplicación Node.js con Grut

En este ejemplo se muestra cómo configurar una aplicación Node.js desarrollada con Grunt. Gruñidoes un corredor de tareas JavaScript de línea de comandos. Ejecuta tareas repetitivas y gestiona la automatización de procesos para reducir los errores humanos. Los plugins Grunt y Grunt se instalan y administran usando npm. Puede configurar Grunt mediante la inclusión de la`Gruntfile.js`Archivo en la raíz de su repositorio de código fuente.

Example package.json

```
{
  "scripts": {
    "build": "grunt uglify",
    "start": "node app.js"
  },
  "devDependencies": {
    "grunt": "~0.4.5",
    "grunt-contrib-jshint": "~0.10.0",
    "grunt-contrib-nodeunit": "~0.4.1",
    "grunt-contrib-uglify": "~0.5.0"
  },
  "dependencies": {
    "express": "^4.15.2"
  },
}
```

Example Gruntfile.js

```
module.exports = function(grunt) {

  // Project configuration.
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    uglify: {
      options: {
        banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-dd") %> */\n'
      },
```

```
      build: {
        src: 'src/<%= pkg.name %>.js',
        dest: 'build/<%= pkg.name %>.min.js'
      }
    }
  });

  // Load the plugin that provides the "uglify" task.
  grunt.loadNpmTasks('grunt-contrib-uglify');

  // Default task(s).
  grunt.registerTask('default', ['uglify']);

};
```

Example apprunner.yaml

```
version: 1.0
runtime: nodejs12
build:
  commands:
    pre-build:
      - npm install grunt grunt-cli
      - npm install --only=dev
      - npm run build
    build:
      - npm install --production
run:
  runtime-version: 12.18.4
  command: node app.js
  network:
    port: 8000
    env: APP_PORT
```

Información de la versión de tiempo de ejecución Node.js

En este tema se enumeran todos los detalles de las versiones en tiempo de ejecución de Node.js compatibles con App Runner.

Node.js 12

Detalle	Descripción
Nombre del runtime	nodejs12
Versiones secundarias	latest
Paquetes incluidos	nodejs (incluido npm), hilados

Desarrollo de código de aplicación para App Runner

En este capítulo se describen las pautas de desarrollo y la información en tiempo de ejecución que debe tener en cuenta al desarrollar o migrar código de aplicación para su implementación aAWS App Runner.

Información de tiempo de ejecución

Tanto si proporciona una imagen de contenedor como si App Runner crea una para usted, App Runner ejecuta el código de la aplicación en una instancia de contenedor. Estos son algunos aspectos clave del entorno de tiempo de ejecución de instancia de contenedor.

- Soporte marco— App Runner admite cualquier imagen que implemente una aplicación web. Es independiente del lenguaje de programación que elija y del servidor de aplicaciones web o marco de trabajo que utilice, si utiliza alguno. Para su comodidad, proporcionamos tiempos de ejecución administrados específicos del idioma para optimizar el proceso de compilación de aplicaciones y la creación de imágenes abstractas.
- Solicitudes Web— La instancia de contenedor debe escuchar solicitudes HTTP, en el puerto 8080 de forma predeterminada. Para obtener más información sobre la configuración de su servicio, consultethe section called “Configuración”. No es necesario que implemente la gestión del tráfico seguro HTTPS. App Runner requiere tráfico HTTPS entrante y finaliza HTTPS antes de pasar solicitudes a la instancia de contenedor.
- Aplicaciones sin estado— App Runner no garantiza la persistencia del estado más allá de la duración del procesamiento de una sola solicitud web entrante.
- Almacenamiento de— App Runner implementa el sistema de archivos en tu instancia de contenedor comoalmacenamiento efímero. Los archivos son transitorios. Por ejemplo, no persisten cuando pausas y reanuda el servicio App Runner. De manera más general, no se garantiza que los archivos persistan más allá del procesamiento de una sola solicitud, como parte de la naturaleza sin estado de su aplicación. Sin embargo, los archivos almacenados ocupan parte de la asignación de almacenamiento de su servicio App Runner durante toda su vida útil.

> **Note**
>
> Aunque es posible que los archivos de almacenamiento efímeros no persistan en las solicitudes,A vecesPersisten. Esto puede ser útil en determinadas situaciones. Por ejemplo, al gestionar una solicitud, puede almacenar en caché los archivos que descarga

la aplicación si las solicitudes futuras pueden necesitarlos. Esto podría acelerar el manejo de solicitudes en el futuro, pero no puede garantizar las ganancias de velocidad. Su código no debe suponer que todavía existe un archivo que se ha descargado en una solicitud anterior.
Para el almacenamiento en caché garantizado mediante un almacén de datos en memoria de alto rendimiento y baja latencia, utilice un servicio comoAmazon ElastiCache.

- Environment variables (Variables de entorno):— De forma predeterminada, App Runner hace que el`PORT`disponible en la instancia de contenedor. Puede configurar el valor de la variable con información de puerto y agregar variables y valores de entorno personalizados. Para obtener más información sobre la configuración de su servicio, consultethe section called “Configuración”.
- Rol de instancia— Si su código de aplicación realiza llamadas a cualquierAWS, utilizando las API de servicio o una de lasAWSSDK, cree un rol de instancia medianteAWS Identity and Access Management(IAM). A continuación, adjuntarlo a su servicio App Runner cuando lo cree. Incluir todoAWSpermisos de acción de servicio que el código requiere en su función de instancia. Para obtener más información, consulte the section called “Rol de instancia”.

Directrices de desarrollo de código

Tenga en cuenta estas pautas al desarrollar código para una aplicación web de App Runner.

- Código sin estado: diseñe la aplicación web que implementa en su servicio App Runner para que no tenga estado. Su código debe suponer que ningún estado persiste más allá de la duración del procesamiento de una sola solicitud web entrante.
- Eliminación de archivos temporales: cuando crea archivos, se almacenan en un sistema de archivos y ocupan parte de la asignación de almacenamiento de su servicio. Para evitar errores fuera de almacenamiento, no guarde archivos temporales durante períodos prolongados. Equilibre el tamaño del almacenamiento con la velocidad de manejo de solicitudes al tomar decisiones de almacenamiento en caché de archivos.

Uso de la consola de App Runner

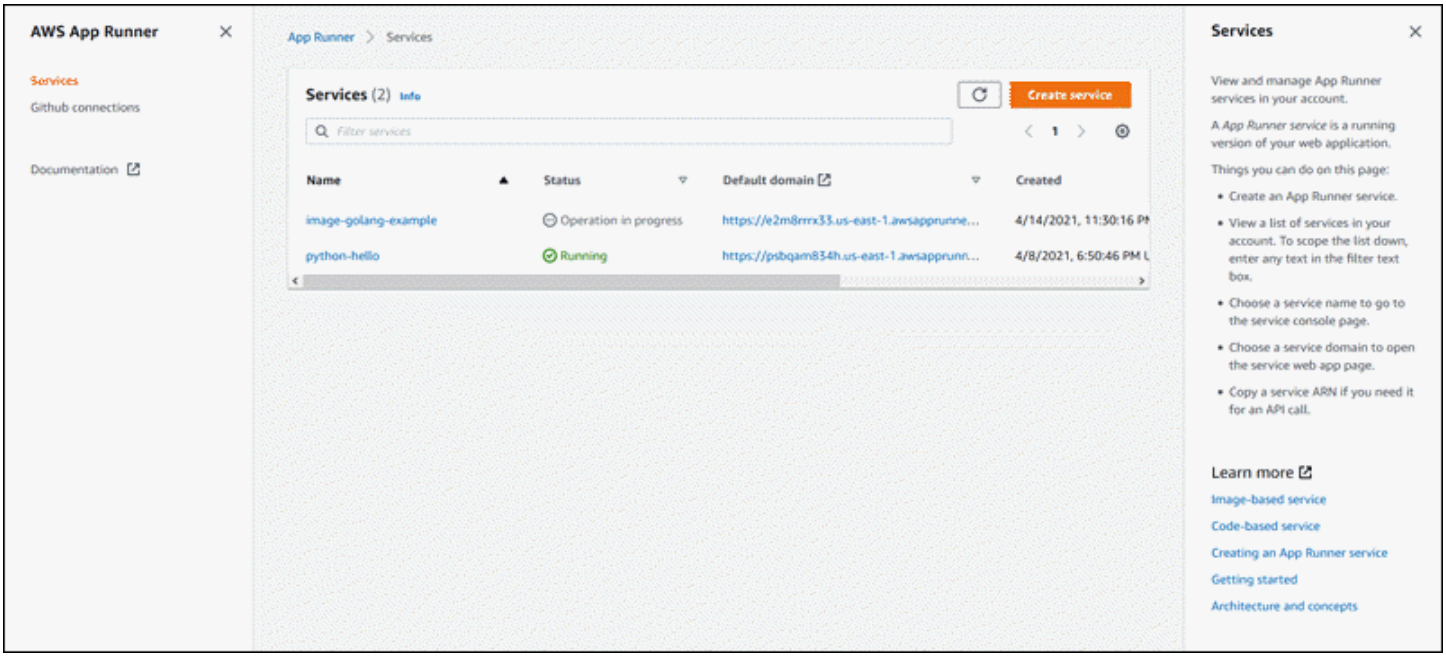
UsarAWS App Runnerpara crear, administrar y supervisar los servicios de App Runner y los recursos relacionados, como las conexiones. Puede ver los servicios existentes, crear otros nuevos y configurar un servicio. Puede ver el estado de un servicio de App Runner, así como ver registros, supervisar la actividad y realizar un seguimiento de las métricas. También puede navegar al sitio web de su servicio o al repositorio de origen.

En las secciones siguientes se describe el diseño y la funcionalidad de la consola y se indica información relacionada.

Diseño general de la consola

La consola de App Runner tiene tres áreas. De izquierda a derecha:

- Panel de navegación— Un panel lateral que se puede contraer o expandir. Úsalo para elegir la página de consola de nivel superior que desea utilizar.
- Panel de contenido— La parte principal de la página de la consola de. Utilícelo para ver información y realizar sus tareas.
- Panel de ayuda— Un panel lateral para obtener más información. Expandelo para obtener ayuda sobre la página en la que estás. O elige cualquierInformaciónen una página de consola para obtener ayuda contextual.



La página de servicios

LaServiciosmuestra los servicios de App Runner en tu cuenta. Puede reducir el alcance de la lista mediante el cuadro de texto del filtro.

Para llegar a laServicespage

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, seleccioneServicios.

Cosas que puedes hacer aquí:

- Cree un servicio de App Runner. Para obtener más información, consulte the section called “Creación”.
- Elija un nombre de servicio para ir a la página de la consola del panel de servicio.
- Elija un dominio de servicio para abrir la página de la aplicación web del servicio.

La página del panel de servicio

Puede ver información sobre un servicio de App Runner y administrarlo desde la página dashbaord del servicio. En la parte superior de la página, puede ver el nombre del servicio.

Para acceder al panel de servicio, navegue hasta elServicios(consulte la sección anterior) y, a continuación, elija su servicio App Runner.

LaInformación general del servicioproporciona detalles básicos sobre el servicio App Runner y su aplicación. Cosas que puedes hacer aquí:

- Vea los detalles del servicio, como el estado, el estado y el ARN.
- Vaya a la .Dominio predeterminado: el dominio que App Runner proporciona para la aplicación web que se ejecuta en su servicio. Este es un subdominio en el`awsapprunner.com`propiedad de App Runner.
- Desplácese hasta el repositorio de origen implementado en el servicio.
- Inicie una implementación del repositorio de origen en su servicio.
- Pausar, reanudar y eliminar el servicio.

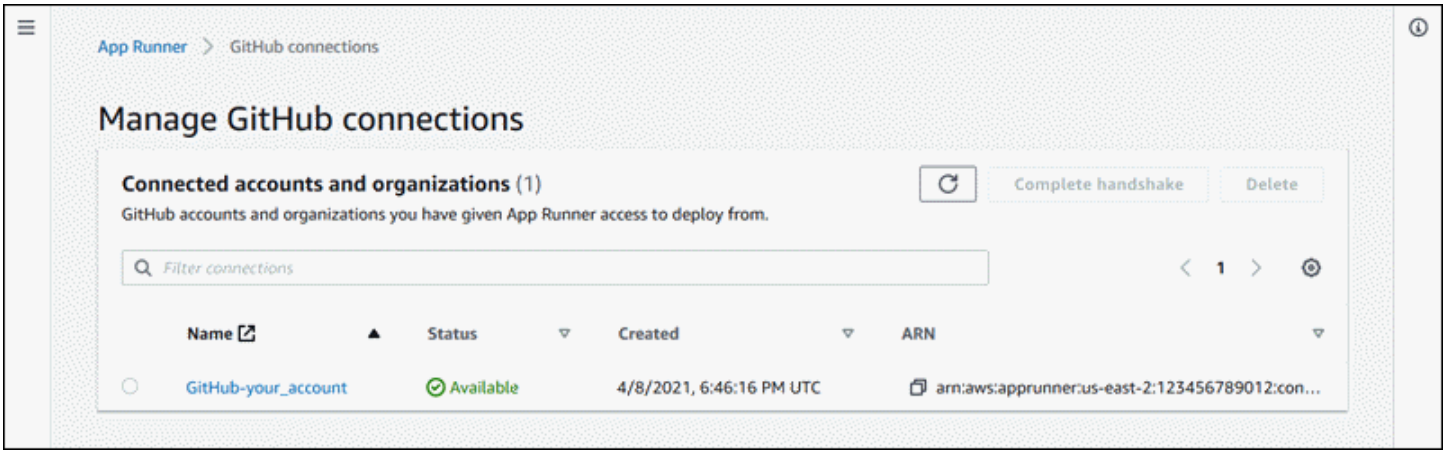
Las pestañas debajo de la descripción general del servicio son para el servicioadministraciónyMonitorización de.

La página de conexiones de GitHub

LaConexiones GitHubmuestra las conexiones de App Runner a GitHub en tu cuenta. Puede reducir el alcance de la lista mediante el cuadro de texto del filtro. Para obtener más información acerca de las conexiones, consulte the section called “Conexiones”.

Para llegar a laConexiones GitHubpage

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, seleccioneConexiones GitHub.



Cosas que puedes hacer aquí:

- Descripta cómo ver una lista de las conexiones de GitHub de una cuenta. Para reducir el alcance de la lista, escriba cualquier texto en el cuadro de texto del filtro.
- Elija un nombre de conexión para ir a la cuenta u organización relacionada de GitHub.
- Seleccione una conexión para completar el enlace de una conexión que acaba de establecer (como parte de la creación de un servicio) o para eliminar la conexión.

Gestión del ciclo de vida del servicio de App Runner

En este capítulo se describe cómo administrar el ciclo de vida de suAWS App RunnerServicio En este capítulo, aprenderá a crear, configurar y eliminar un servicio, a implementar nuevas versiones de aplicaciones en el servicio y a administrar conexiones. También aprenderá a controlar la disponibilidad de su servicio web haciendo una pausa y reanudando el servicio.

Temas

Creación de un servicio App Runner

AWS App Runnerautomatiza el proceso de pasar de una imagen de contenedor o de un repositorio de código fuente a un servicio web en ejecución que se escala automáticamente. Puede apuntar App Runner a su imagen o código fuente, especificando sólo un pequeño número de configuraciones requeridas. App Runner crea su aplicación si es necesario, aprovisiona recursos informáticos e implementa la aplicación para ejecutarlos en ellos.

Cuando crea un servicio, App Runner crea unaServicioRecurso. En algunos casos, es posible que tenga que proporcionar unconexiónRecurso. Si utiliza la consola de App Runner, la consola crea implícitamente el recurso de conexión. Para obtener detalles acerca de los tipos de recursos de App Runner, consultethe section called “Recursos de App Runner”. Estos tipos de recursos tienen cuotas asociadas a su cuenta en cadaRegión de AWS. Para obtener más información, consulte the section called “Cuotas de recursos”.

Existen sutiles diferencias en el procedimiento para crear un servicio según el tipo de origen y el proveedor. En este tema se muestran procedimientos completamente independientes para crear

estos diferentes tipos de origen, de modo que pueda seguir el que mejor se ajuste a su situación. Para obtener un procedimiento básico de inicio con un ejemplo de código, consulteIntroducción.

Prerequisites

Antes de crear el servicio App Runner, asegúrese de completar las siguientes acciones:

- Realice los pasos que se indican en laConfiguración.
- Tener listo el origen de la aplicación. Puede utilizar un repositorio de código enGitHubo una imagen de contenedor enAmazon Elastic Container Registry (Amazon ECR)para crear un servicio App Runner.

Crear un servicio

En esta sección se explica el proceso de creación de los dos tipos de servicio de App Runner: basado en el código fuente y en una imagen de contenedor.

Crear un servicio desde un repositorio de código de GitHub

En las secciones siguientes se muestra cómo crear un servicio de App Runner cuando la fuente es un repositorio de código enGitHub. Cuando usas GitHub, App Runner tiene que conectarse a la organización o cuenta de GitHub. Por tanto, no tiene que ayudar a establecer esta conexión. Para obtener más información acerca de las conexiones de App Runner, consultethe section called “Conexiones”.

Cuando crea el servicio, App Runner crea una imagen Docker que contiene el código de la aplicación y las dependencias. A continuación, inicia un servicio que ejecuta una instancia de contenedor de esta imagen.

Temas

- Creación de un servicio a partir de código mediante la consola de App Runner
- Crear un servicio a partir de código mediante la API de App Runner oAWS CLI

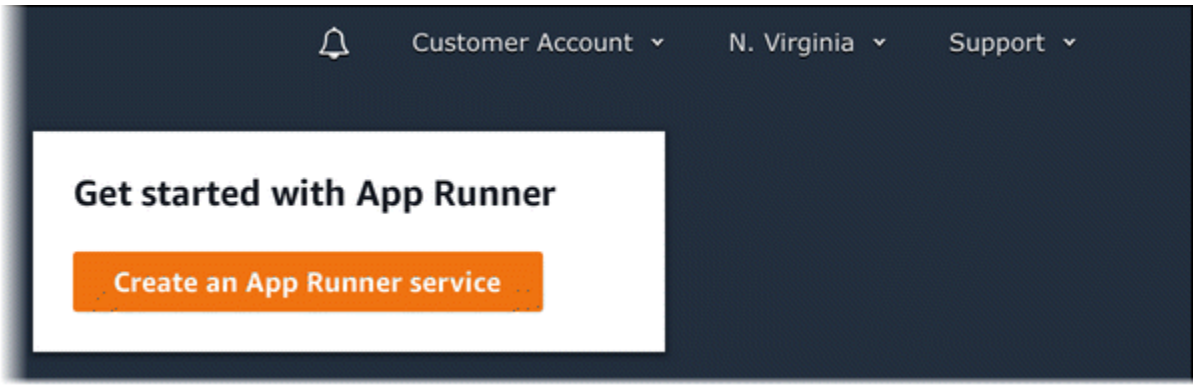
Creación de un servicio a partir de código mediante la consola de App Runner

Para crear un servicio de App Runner con la consola

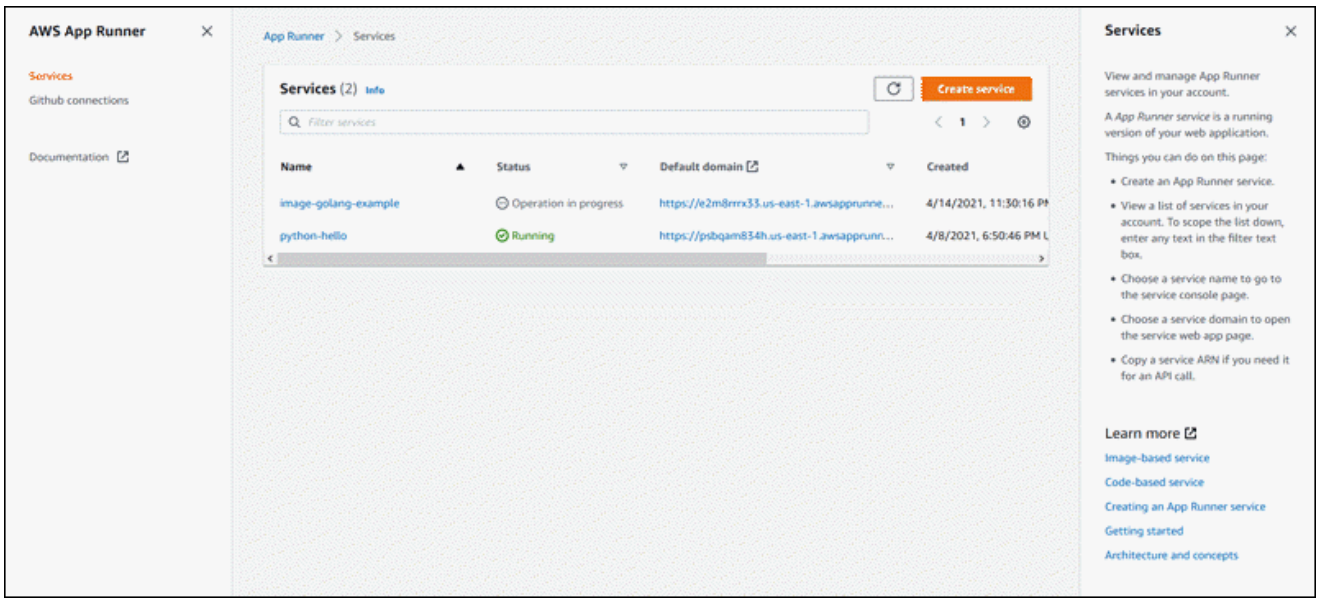
1. Configure su código fuente.

a. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.

b. Si el archivo deCuenta de AWSaún no tiene ningún servicio de App Runner, se muestra la página de inicio de la consola. SeleccionarCreación de un servicio App Runner.



Si el archivo deCuenta de AWStiene servicios existentes, elServiciosAparecerá una página con una lista de sus servicios. Elija Create service.



c. En la páginaOrigen e implementaciónPágina Web, en laFuentesección, paraRepository type, elijaRepositorio de código fuente.

d. ParaConnect a GitHub, seleccione una cuenta u organización de GitHub que haya utilizado anteriormente o elijaAgregar nuevo. A continuación, realice el proceso de proporcionar sus credenciales de GitHub y elija una cuenta de GitHub u organización a la que conectarse.

e. ParaRepositorioSeleccione el repositorio que contenga el código de la aplicación.

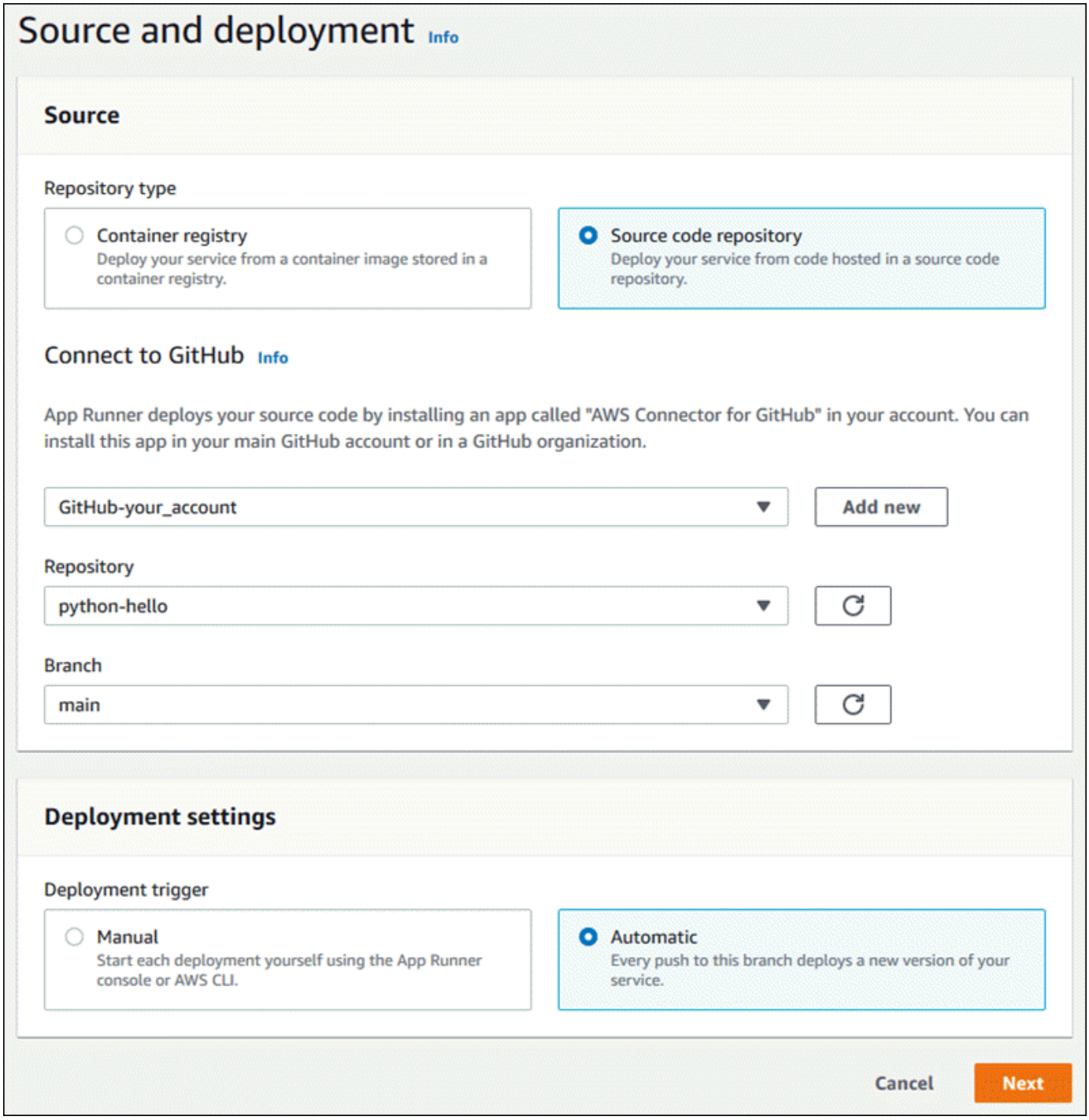
f. ParaCrear ramificacionesSeleccione la ramificación que desea implementar.

2. Configure sus implementaciones.

a. En el navegadorConfiguración de la implementaciónsección, elija.ManualorAutomático.

   Para obtener más información acerca de los métodos de implementación, consultethe section called “Métodos de implementación”.

b. Seleccione Next (Siguiente).



3. Configure la compilación de la aplicación.

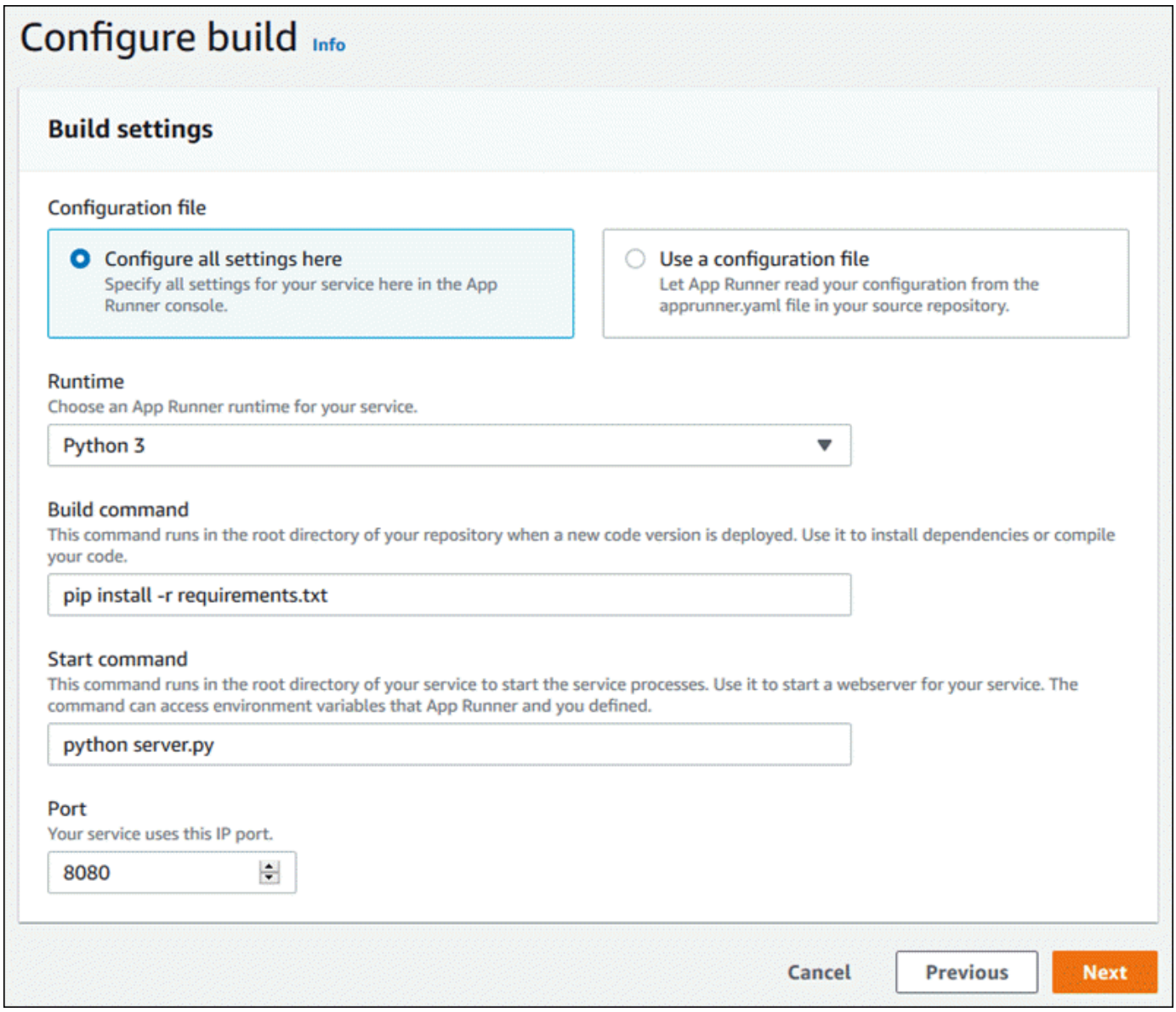
   a. En la páginaConfigurar la compilaciónPágina Web, paraArchivo de configuración, elijaConfigurar todas las opciones aquísi el repositorio no contiene un archivo de configuración de App Runner, oUso de un archivo de configuraciónSi es así.

      > **Note**
      >
      > Un archivo de configuración de App Runner es una forma de mantener la configuración de compilación como parte del origen de la aplicación. Cuando proporciona uno, App Runner lee algunos valores del archivo y no permite configurarlos en la consola.

   b. Proporcione la siguiente configuración de compilación:

      - Runtime (Tiempo de ejecución):: elija un tiempo de ejecución administrado específico para su aplicación.
      - Comando de compilación— Introduzca un comando que construya la aplicación a partir del código fuente. Esta puede ser una herramienta específica del idioma o un script proporcionado con el código.
      - Comando de inicio— Introduzca el comando que inicia el servicio web.
      - Puerto— Introduzca el puerto IP que escucha el servicio web.

   c. Seleccione Next (Siguiente).



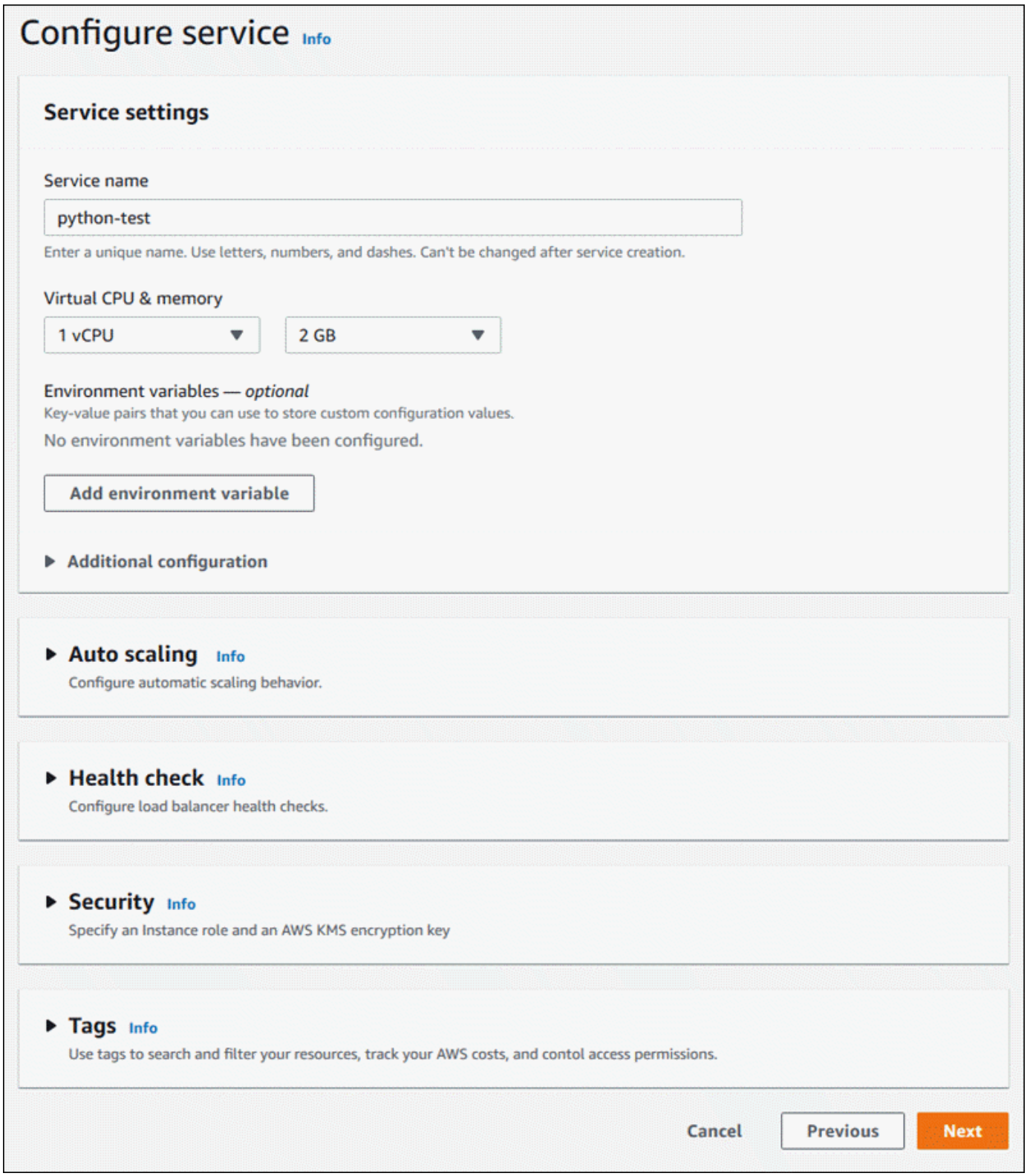
4. Configure su servicio.

   a. En la páginaConfigure el servicioPágina Web, en laConfiguración del servicio, escriba un nombre de servicio.

   > **Note**
   >
   > Todas las demás configuraciones de servicio son opcionales o tienen valores predeterminados proporcionados por la consola.

   b. Si lo desea, cambie o agregue otros ajustes para satisfacer los requisitos de la aplicación.

c. Seleccione Next (Siguiente).



5. En la páginaRevisar y crear, verifique todos los detalles que haya introducido y, a continuación, elijaCreación e implementación.

   Resultado: Si la creación del servicio se realiza correctamente, la consola debe mostrar el panel de servicio, con unInformación general del serviciodel nuevo servicio.

6. Compruebe que su servicio está en ejecución.

   a. En la página del panel de servicio, espere hasta que el servicioEstadoesEn ejecución.

   b. Elija el iconoDominio predeterminadovalor: es la URL del sitio web de su servicio.

   c. Utilice su sitio web y verifique que está funcionando correctamente.

Crear un servicio a partir de código mediante la API de App Runner oAWS CLI

Para crear un servicio mediante la API de App Runner oAWS CLI, llame a la`CreateService`Acción de la API. Para obtener más información y un ejemplo, consulte.CreateService. Si es la primera vez que crea un servicio usando una organización o cuenta específica de GitHub, comience llamando aCreateConnection. Esto establece una conexión entre App Runner y la organización o cuenta de GitHub. Para obtener más información acerca de las conexiones de App Runner, consultethe section called “Conexiones”.

La creación de su servicio se inicia si la llamada devuelve una respuesta correcta con unService (Servicio)objeto que muestra`"Status": "CREATING"`.

Para ver una llamada de ejemplo, consulte.Creación de un servicio de repositorio de código fuenteen laAWS App RunnerReferencia de la API

Creación de un servicio a partir de una imagen de Amazon ECR

En las secciones siguientes se muestra cómo crear un servicio de App Runner cuando la fuente es una imagen de contenedor almacenada enAmazon ECR. Amazon ECR es unAWSservicioservicio Por lo tanto, para crear un servicio basado en una imagen de Amazon ECR, debe proporcionar a App Runner un rol de acceso que contenga los permisos de acción de Amazon ECR necesarios.

> **Note**
>
> No se requiere un rol de acceso si la imagen está almacenada en Amazon ECR Public, donde las imágenes están disponibles públicamente.

Durante la creación del servicio, App Runner inicia un servicio que ejecuta una instancia de contenedor de la imagen que proporciona. No hay fase de compilación en este caso.

Temas

- Crear un servicio a partir de una imagen mediante la consola de App Runner
- Crear un servicio a partir de una imagen utilizando la API de App Runner oAWS CLI

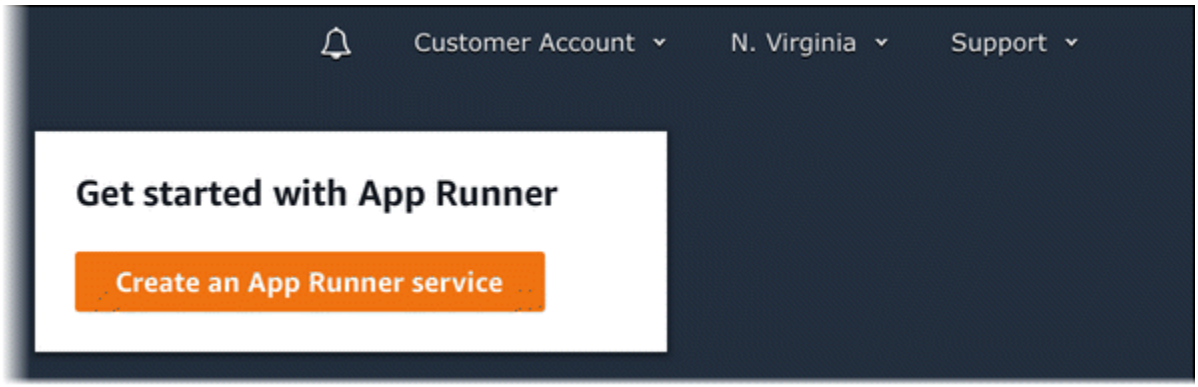
Crear un servicio a partir de una imagen mediante la consola de App Runner

Para crear un servicio de App Runner con la consola

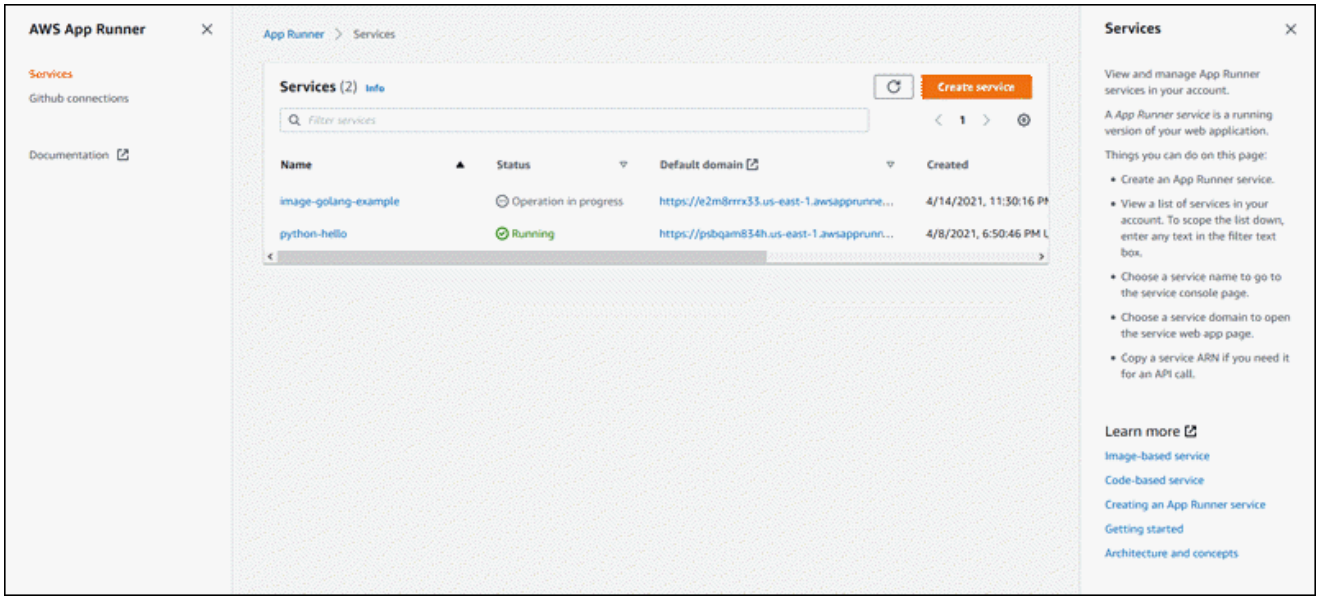
1. Configure su código fuente.

   a. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.

   b. Si el archivo deCuenta de AWSaún no tiene ningún servicio de App Runner, se muestra la página de inicio de la consola. SeleccionarCreación de un servicio App Runner.



   Si el archivo deCuenta de AWStiene servicios existentes, elServiciosAparecerá una página con una lista de sus servicios. Elija Create service.



c. En la páginaOrigen e implementaciónPágina Web, en laFuentesección, paraRepository type, elijaRegistro de contenedores.

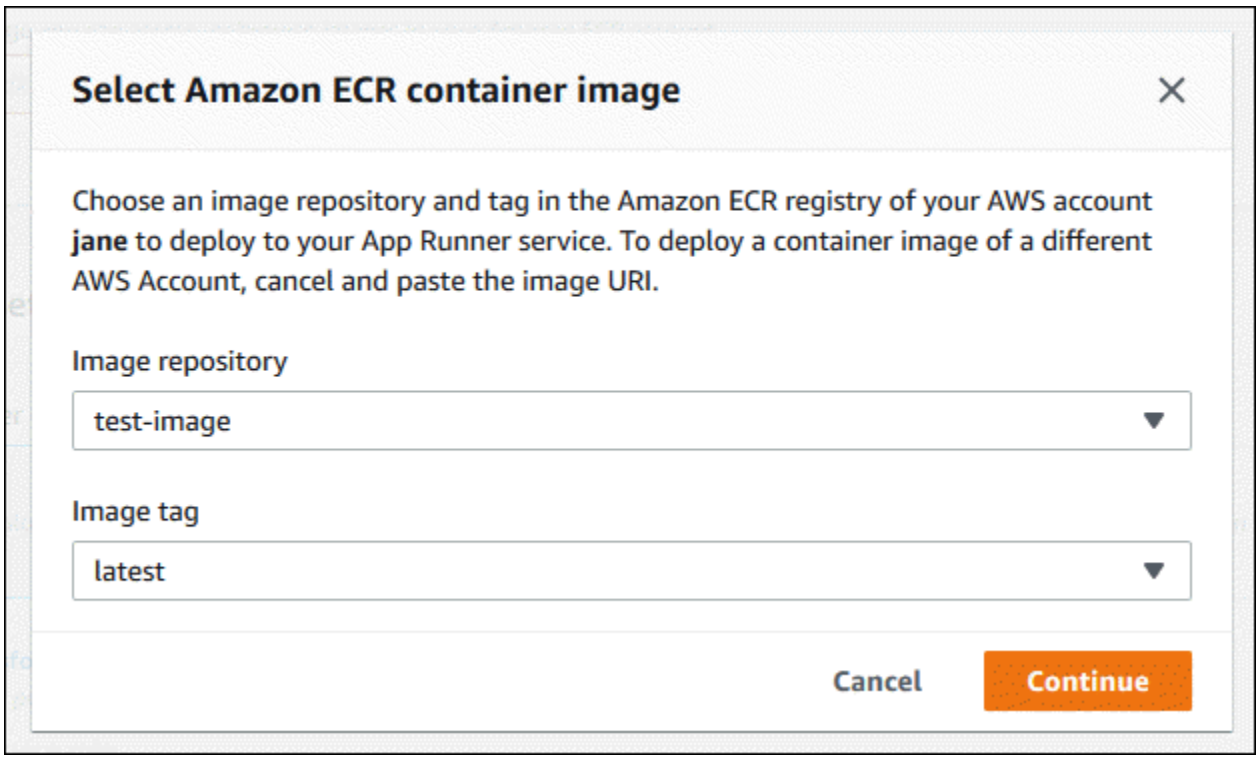
d. ParaProveedor, elija el proveedor donde se almacena su imagen:

- Amazon ECR— Una imagen privada almacenada en Amazon ECR en suCuenta de AWS.
- Amazon ECR Public: imagen legible públicamente almacenada en Amazon ECR Public.

e. ParaURI de imágenes de contenedores, elijaBrowse.

f. En el navegadorSeleccionar imagen de contenedor de Amazon ECRCuadro de diálogo, pararepositorio de imágenes, seleccione el repositorio que contiene la imagen.

g. ParaEtiqueta de imágenesSeleccione la etiqueta de imagen específica que desea implementar, por ejemploLa más recientey luego elijaContinuar.



2. Configure sus implementaciones.

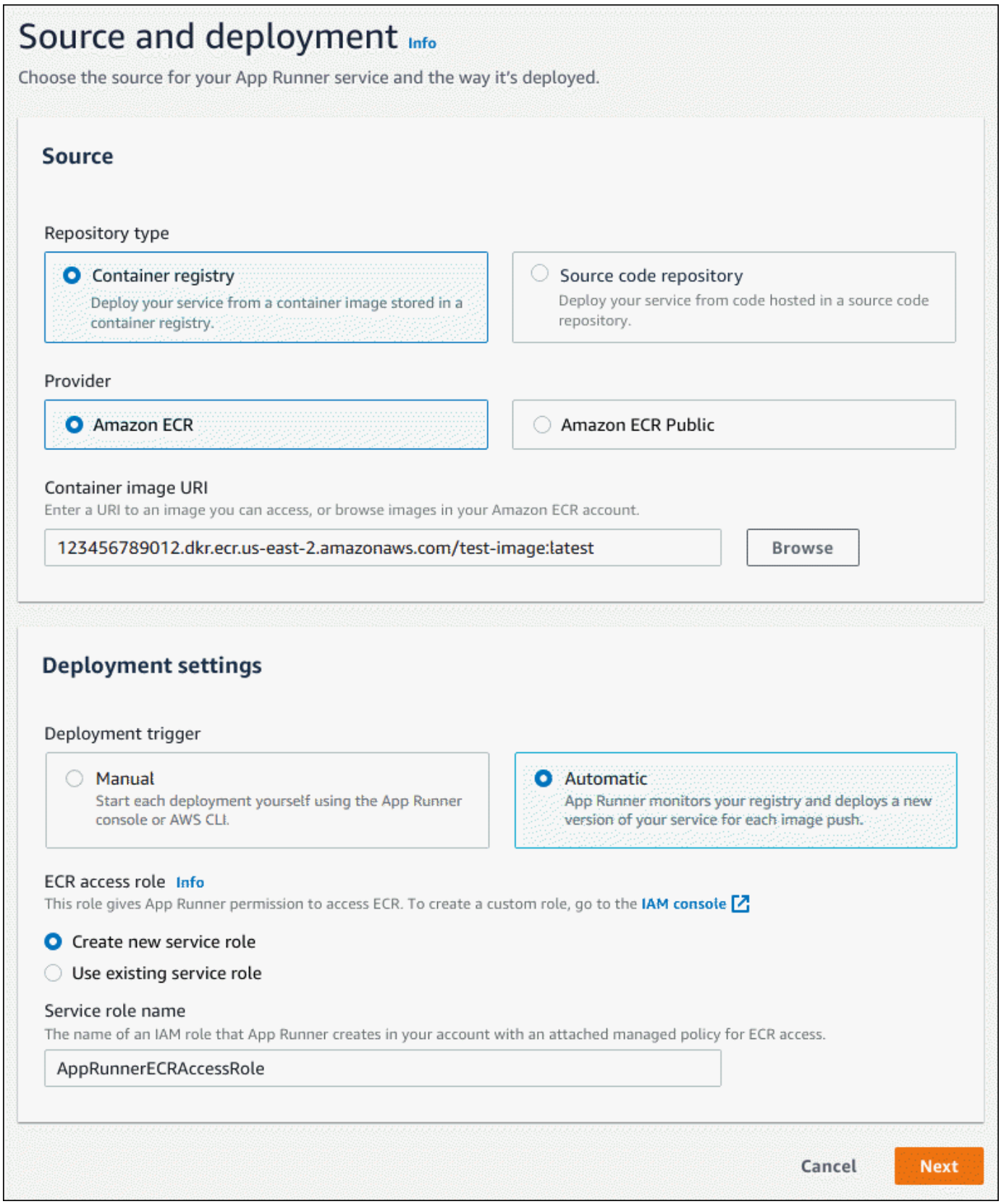
   a. En el navegadorConfiguración de la implementaciónsección, elija.ManualorAutomático.

      Para obtener más información acerca de los métodos de implementación, consultethe section called “Métodos de implementación”.

      > **Note**
      >
      > App Runner no admite la implementación automática de imágenes públicas de Amazon ECR.

   b. [Amazon ECRProveedor] ParaFunción de acceso ECRElija un rol de servicio existente en su cuenta o elija crear uno nuevo. Si utiliza la implementación manual, también puede optar por usar la función de usuario de IAM en el momento de la implementación.

   c. Seleccione Next (Siguiente).


Source and deployment Info
Choose the source for your App Runner service and the way it's deployed.
Source
Repository type
Container registry
Deploy your service from a container image stored in a container registry.
Source code repository
Deploy your service from code hosted in a source code repository.
Provider
Amazon ECR
Amazon ECR Public
Container image URI
Enter a URI to an image you can access, or browse images in your Amazon ECR account.
123456789012.dkr.ecr.us-east-2.amazonaws.com/test-image:latest
Browse
Deployment settings
Deployment trigger
Manual
Start each deployment yourself using the App Runner console or AWS CLI.
Automatic
App Runner monitors your registry and deploys a new version of your service for each image push.
ECR access role Info
This role gives App Runner permission to access ECR. To create a custom role, go to the IAM console
Create new service role
Use existing service role
Service role name
The name of an IAM role that App Runner creates in your account with an attached managed policy for ECR access.
AppRunnerECRAccessRole
Cancel
Next

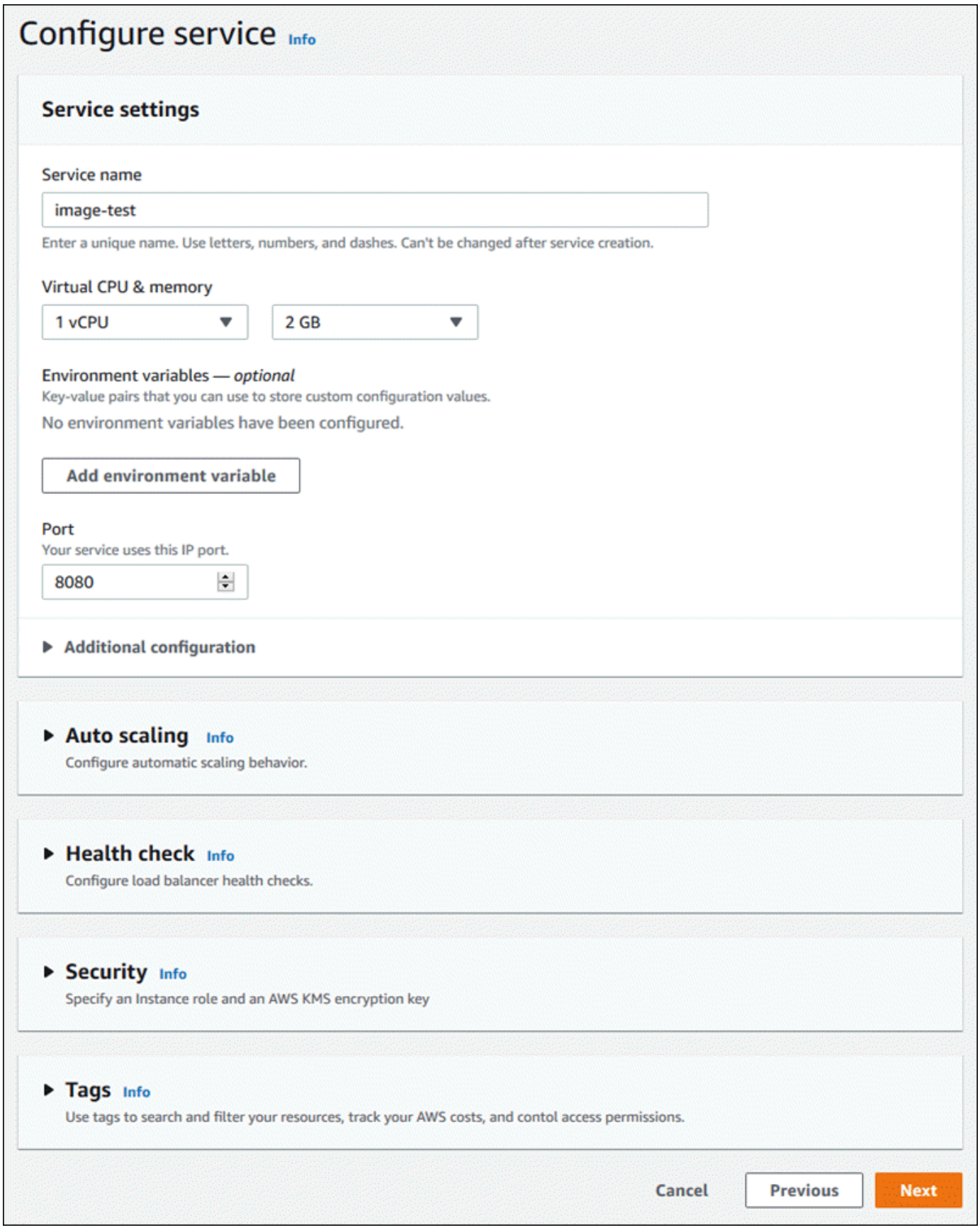
3. Configure su servicio.

   a. En la páginaConfigure el servicioPágina Web, en laConfiguración del servicio, introduzca un nombre de servicio y el puerto IP que el sitio web de servicio escucha.

      > **Note**
      >
      > Todas las demás configuraciones de servicio son opcionales o tienen valores predeterminados proporcionados por la consola.

   b. (Opcional) Cambie o agregue otros ajustes que se adapten a las necesidades de su aplicación.

   c. Seleccione Next (Siguiente).


Configure service Info
Service settings
Service name
image-test
Enter a unique name. Use letters, numbers, and dashes. Can't be changed after service creation.
Virtual CPU & memory
1 vCPU
2 GB
Environment variables — optional
Key-value pairs that you can use to store custom configuration values.
No environment variables have been configured.
Add environment variable
Port
Your service uses this IP port.
8080
Additional configuration
Auto scaling Info
Configure automatic scaling behavior.
Health check Info
Configure load balancer health checks.
Security Info
Specify an Instance role and an AWS KMS encryption key
Tags Info
Use tags to search and filter your resources, track your AWS costs, and contol access permissions.
Cancel
Previous
Next

4. En la páginaRevisar y crearEn la página, verifique todos los detalles que ha especificado y, a continuación, elijaCreación e implementación.

   Resultado: Si la creación del servicio se realiza correctamente, la consola debe mostrar el panel de servicio, con unInformación general del serviciodel nuevo servicio.

5. Compruebe que su servicio está en ejecución.

   a. En la página del panel de servicio, espere hasta que el servicioEstadoesEn ejecución.
   b. Elija el iconoDominio predeterminadovalor: es la URL del sitio web de su servicio.
   c. Utilice su sitio web y verifique que está funcionando correctamente.

Crear un servicio a partir de una imagen utilizando la API de App Runner oAWS CLI

Para crear un servicio mediante la API de App Runner oAWS CLI, llame a laCreateServiceAcción de la API.

La creación de su servicio se inicia si la llamada devuelve una respuesta correcta con unService (Servicio)objeto que muestra`"Status": "CREATING"`.

Para ver una llamada de ejemplo, consulte.Crear un servicio de repositorio de imágenes de origenen laAWS App RunnerReferencia de la API

Cuando la creación del servicio falla

Si se produce un error en el intento de crear un servicio de App Runner, el servicio muestra un estado de`CREATE_FAILED`(Creación fallidaen la consola).

Su intento de crear un servicio puede fallar debido a problemas en el código de la aplicación, el proceso de compilación o la configuración, porque ha alcanzado cuotas de recursos o debido a problemas temporales con elAWSque su servicio necesita usar. Para solucionar un error, recomendamos que realice las acciones siguientes: Primero, lea los eventos del servicio y los registros para averiguar qué causó el error. A continuación, realice los cambios necesarios en su código o configuración. Por último, elimine uno o más servicios si alcanzó su cuota de servicio. Luego, después de completar todos estos pasos, intente crear el servicio de nuevo.

> **Important**
>
> El servicio fallido no se puede utilizar. No se le aplicará ningún cargo adicional por ello más allá del intento de creación inicial. Sin embargo, App Runner no elimina automáticamente el

servicio fallido y sigue teniendo en cuenta para su cuota de servicio. Cuando haya terminado de analizar el error, asegúrese de eliminar el servicio fallido.

Implementación de una nueva versión de la aplicación en App Runner

CuandoCrear un servicioinAWS App Runner, puede configurar una fuente de aplicación: una imagen de contenedor o un repositorio de origen. App Runner aprovisiona recursos para ejecutar el servicio e implementa la aplicación en ellos.

En este tema se describen formas de volver a implementar el origen de la aplicación en el servicio App Runner cuando una nueva versión esté disponible. Puede ser una nueva versión de imagen en el repositorio de imágenes o una nueva confirmación en el repositorio de código. App Runner proporciona dos métodos para implementar en un servicio:AutomáticoyManual.

Métodos de implementación

App Runner proporciona los siguientes métodos para controlar cómo se inician las implementaciones de aplicaciones.

Implementación automática

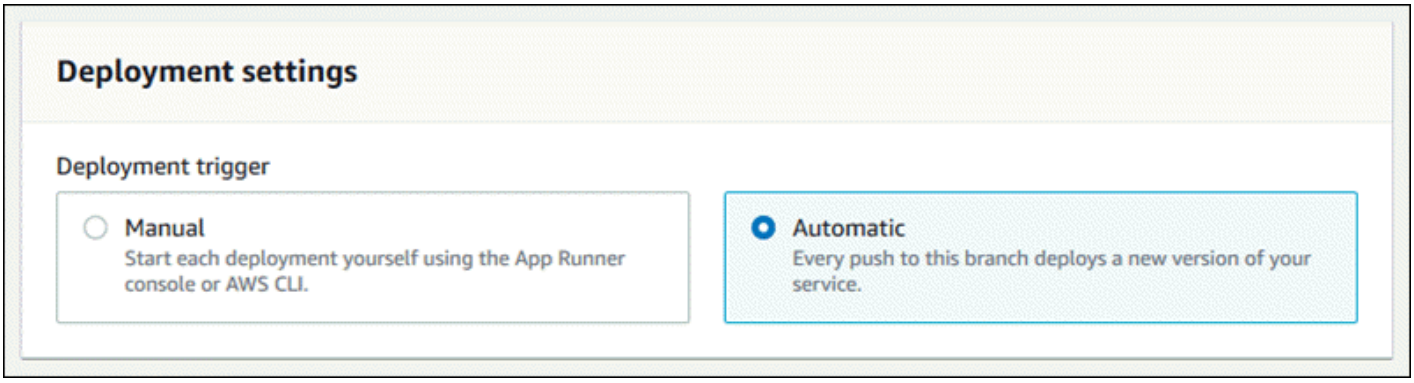
Utilice la implementación automática cuando desee un comportamiento de integración e implementación continuas (CICD) para su servicio. App Runner supervisa el repositorio de imágenes o códigos. Cada vez que inserte una nueva versión de imagen en su repositorio de imágenes, o una nueva confirmación en su repositorio de código, App Runner la implementa automáticamente en su servicio sin ninguna otra acción por su lado.

Implementación manual

Utilice la implementación manual cuando desee iniciar explícitamente cada implementación en su servicio. Inicie una implementación si el repositorio que configuró para el servicio tiene una nueva versión que desea implementar. Para obtener más información, consulte the section called “Implementación manual”.

Puede configurar el método de implementación del servicio de las siguientes formas:

- Consola— Para un nuevo servicio que está creando o para un servicio existente, en elConfiguración de la implementaciónSección sobre de laOrigen e implementaciónPágina de configuración, seleccioneManualorAutomático.



- API oAWS CLI— En una llamada a laCreateServiceorUpdateService, establezca la`AutoDeploymentsEnabled`miembro deSourceConfigurationPara`False`para la implementación manual o`True`para la implementación automática.

Implementación manual

Con la implementación manual, debe iniciar explícitamente cada implementación en su servicio. Cuando tenga una nueva versión de la imagen o el código de la aplicación lista para implementar, puede consultar las secciones siguientes para obtener información sobre cómo realizar una implementación mediante la consola y la API.

Implementar una versión de aplicación con la consola de App Runner

Para realizar la implementación mediante la consola de App Runner

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, seleccioneServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unDescripción general del servicio.
3. Elija Deploy (Implementar).

   Resultado: Se inicia la implementación de la nueva versión. En la página del panel de servicio, el servicioEstadocambios enOperación en curso.
4. Espere a que termine la implementación. En la página del panel de servicio, el servicioEstadodebería cambiar de nuevo aEn ejecución.

5. Para comprobar que la implementación se realiza correctamente, en la página del panel de servicio, seleccione la opciónDominio predeterminadovalor: es la URL del sitio web de su servicio. Inspeccione o interactúe con su aplicación web y verifique el cambio de versión.

Implemente una versión de aplicación mediante la API de App Runner oAWS CLI

Para implementar usando la API de App Runner oAWS CLI, llame aStartDeploymentAcción de la API El único parámetro a pasar es su ARN de servicio. Ya configuró la ubicación de origen de la aplicación cuando creó el servicio y App Runner puede encontrar la nueva versión. La implementación se inicia si la llamada devuelve una respuesta correcta.

Configuración de un servicio de App Runner

CuandoCree unAWS App RunnerServicio, se establecen varios valores de configuración. Puede cambiar algunas de las opciones de configuración después de crear el servicio. Otros ajustes sólo se pueden aplicar mientras se crea el servicio y no se pueden cambiar a partir de entonces. En este tema se describe la configuración del servicio mediante la API de App Runner, la consola de App Runner y un archivo de configuración de App Runner.

Configure su servicio mediante la API de App Runner oAWS CLI

La API define qué configuración se puede cambiar después de la creación del servicio. En la siguiente lista se describen las acciones, tipos y limitaciones relevantes.

- UpdateServiceaction: se puede llamar después de la creación para actualizar algunos ajustes de configuración.
  - Se puede actualizar— Puede actualizar los ajustes en el cuadro de diálogo`SourceConfiguration`,`InstanceConfiguration`, y`HealthCheckConfiguration`Parámetros. Sin embargo, en`SourceConfiguration`, no puede cambiar su tipo de fuente de código a imagen o al revés. Debe proporcionar el mismo parámetro de repositoryparameters que proporcionó al crear el servicio. Es cualquiera`CodeRepository`or`ImageRepository`.

    También puede actualizar`AutoScalingConfigurationArn`, el ARN del recurso de configuración de escalado automático asociado con el servicio.
  - No se puede actualizar— No se puede cambiar el`ServiceName`y`EncryptionConfiguration`que están disponibles en

elCreateServiceaction. No se pueden cambiar después de ser creados. LaUpdateServiceno incluye estos parámetros.

- Archivo API frente a— Puede establecer el`ConfigurationSource`Parámetro del parámetroCodeConfiguration(utilizado para repositorios de código fuente como parte de`SourceConfiguration`) a`Repository`. En este caso, App Runner ignora la configuración de`CodeConfigurationValues`, y lee estos ajustes desde unArchivo de configuraciónEn el repositorio. Si configura`ConfigurationSource`De a`API`, App Runner obtiene todos los ajustes de configuración de la llamada a la API e ignora el archivo de configuración, incluso si existe uno.

- TagResourceaction: se puede llamar después de crear el servicio para agregar etiquetas al servicio o actualizar los valores de las etiquetas existentes.
- UntagResourceaction: se puede llamar después de crear el servicio para eliminar etiquetas del servicio.

Configurar el servicio mediante la consola de App Runner

La consola utiliza la API de App Runner para aplicar las actualizaciones de configuración. Las reglas de actualización que impone la API, tal como se definen en la sección anterior, determinan lo que se puede configurar mediante la consola. Algunas opciones de configuración que estaban disponibles durante la creación del servicio no están disponibles para su modificación más adelante. Además, si decide usar unArchivo de configuración, los ajustes adicionales se ocultan en la consola y App Runner los lee desde el archivo.

Para configurar el servicio

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, elijaServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.
3. En la página del panel de servicio, elija la opciónConfiguraciónPestaña.

   Resultado: La consola muestra la configuración actual de su servicio en varias secciones: Origen e implementación,Configuración de la compilación, yConfigure el servicio.
4. Para actualizar la configuración de cualquier categoría, elijaEditar.
5. En la página de edición de la configuración, realice los cambios deseados y, a continuación, elijaGuarde los cambios.

Configurar el servicio mediante un archivo de configuración de App Runner

Al crear o actualizar un servicio de App Runner, puede indicar a App Runner que lea algunas opciones de configuración de un archivo de configuración que proporcione como parte del repositorio de origen. Al hacer esto, puede administrar la configuración relacionada con su código fuente bajo control fuente, junto con el código en sí. El archivo de configuración también proporciona ciertas opciones avanzadas que no se pueden establecer mediante la consola o la API. Para obtener más información, consulte Archivo de configuración de App Runner.

Administración de conexiones de App Runner

CuandoCrear un servicioinAWS App Runner, se configura una fuente de aplicación: una imagen de contenedor o un repositorio de origen que se almacena con un proveedor. Si un repositorio almacenado con un proveedor externo es privado (no legible públicamente), App Runner tiene que establecer una conexión autenticada y autorizada con el proveedor. A continuación, App Runner puede leer su repositorio e implementarlo en su servicio. App Runner no requiere el establecimiento de la conexión cuando crea un servicio que accede al código almacenado en suCuenta de AWSo en una ubicación de código público.

App Runner mantiene la información de conexión en un recurso llamadoconexión. App Runner requiere un recurso de conexión cuando crea un servicio que necesita información de conexión de terceros. A continuación se muestra información importante sobre las conexiones:

- Proveedores— App Runner actualmente requiere recursos de conexión conGitHub.
- Compartido: puede utilizar un recurso de conexión para crear varios servicios de App Runner que utilicen la misma cuenta de proveedor de repositorio.
- Administración de recursos— En App Runner, puedes crear y eliminar conexiones. Sin embargo, no puede modificar una conexión existente.
- Cuotas de recursos— Los recursos de conexión tienen una cuota establecida asociada a suCuenta de AWSEn cadaRegión de AWS. Si alcanza esta cuota, es posible que deba eliminar una conexión antes de poder conectarse a una nueva cuenta de proveedor. Puede eliminar una conexión con el App Runnerconsoleor API. Para obtener más información, consulte the section called “Cuotas de recursos”.

Administrar conexiones con la consola de App Runner

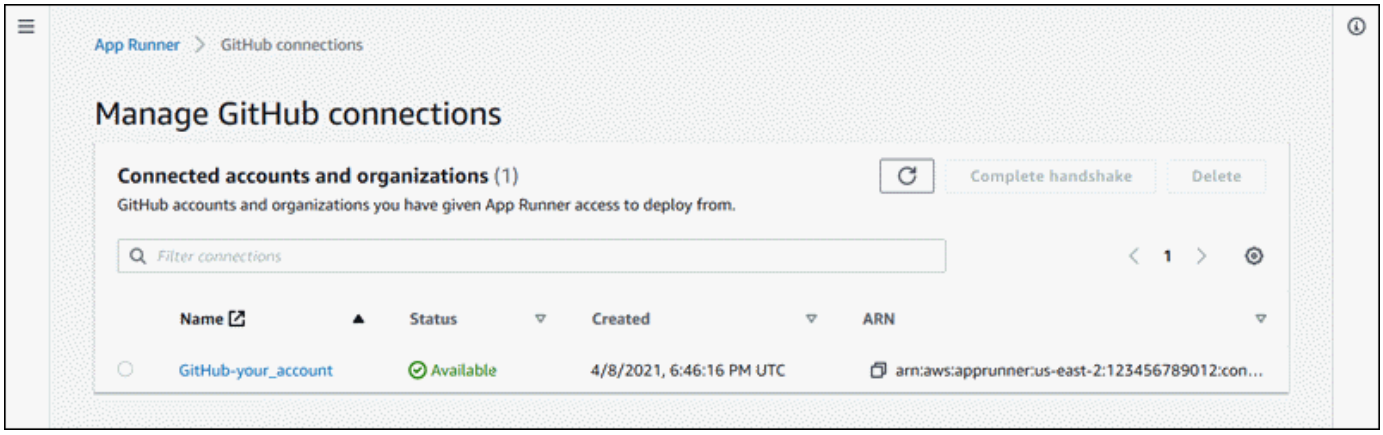
Cuando utilice la consola de App Runner paraCrear un servicio, proporciona los detalles de la conexión. No es necesario crear explícitamente un recurso de conexión. En la consola, puedes elegir conectarte a una cuenta de GitHub a la que te hayas conectado anteriormente o conectarte a una cuenta nueva. Cuando sea necesario, App Runner crea un recurso de conexión para usted. Para una nueva conexión, algunos proveedores (por ejemplo, GitHub) requieren que complete un protocolo de enlace de autenticación antes de poder usar la conexión. La consola le guiará por este proceso.

La consola también tiene una página para administrar las conexiones existentes. Puede completar el protocolo de enlace de autenticación para una conexión si no lo hizo cuando creó el servicio. También puede eliminar conexiones que ya no utiliza. En el siguiente procedimiento se muestra cómo puede administrar las conexiones de GitHub.

Para administrar las conexiones de GitHub en tu cuenta

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, elijaConecciones de GitHub.

   A continuación, la consola muestra una lista de conexiones de GitHub en su cuenta.



3. Ahora puede realizar una de las siguientes acciones con cualquier conexión de la lista:

   - Abrir cuenta u organización de GitHub— Elija el nombre de la conexión.
   - protocolo de enlace de autenticación completo— Seleccione la conexión de y, a continuación, elijaprotocolo de enlace completo. La consola le lleva a través del proceso de protocolo de enlace de autenticación.
   - Eliminar conexión— Seleccione la conexión de y, a continuación, elijaEliminar. Siga las instrucciones de la solicitud de eliminación.

Gestione las conexiones mediante la API de App Runner oAWS CLI

Puede utilizar las siguientes acciones de la API de App Runner para administrar las conexiones.

- CreateConnection— Crea una conexión con una cuenta de proveedor de repositorio. Después de crear la conexión, debe completar manualmente el protocolo de enlace de autenticación mediante la consola de App Runner. Este proceso se explica en la sección anterior.
- ListConnections— Devuelve una lista de las conexiones de App Runner asociadas a tuCuenta de AWS.
- DeleteConnection— Elimina una conexión. Es posible que tenga que eliminar conexiones innecesarias si alcanza la cuota de conexión de suCuenta de AWS.

Administración del escalado automático de App Runner

AWS App Runnerescala automáticamente los recursos informáticos (instancias) hacia arriba o hacia abajo para su aplicación App Runner. El escalado automático proporciona un manejo adecuado de solicitudes cuando el tráfico entrante es alto y reduce el costo cuando el tráfico se ralentiza. Puede configurar algunos parámetros para ajustar el comportamiento de escalado automático para su servicio.

App Runner mantiene la configuración de escalado automático en un recurso llamadoAutoescalingConfiguration. Puede proporcionar un recurso de configuración de Auto Scaling al crear o actualizar un servicio. La consola de App Runner crea uno para usted cuando crea un nuevo servicio de App Runner. Proporcionar una configuración de escalado automático es opcional. Si no proporcionan una, App Runner proporciona una configuración predeterminada de Auto Scaling con los valores recomendados.

Una configuración de escalado automático tiene unNombre dey numericrevisión. Varias revisiones de una configuración tienen el mismo nombre y números de revisión diferentes. Puede utilizar diferentes nombres de configuración para diferentes escenarios de escalado automático, como alta disponibilidad o bajo costo. Para cada nombre, puede agregar varias revisiones para ajustar la configuración de un escenario específico.

A continuación se muestra información importante acerca de las configuraciones de escalado automático:

- ConfiguraciónEste es el procedimiento:

- Simultaneidad máxima— Número máximo de solicitudes simultáneas que una instancia procesa. Cuando el número de solicitudes simultáneas supera esta cuota, App Runner amplía el servicio.
- Tamaño máximo— Número máximo de instancias que el servicio puede escalar. Como máximo, este número de instancias sirven activamente al tráfico de su servicio.
- Tamaño Min— El número mínimo de instancias que App Runner proporciona para su servicio. El servicio siempre tiene al menos este número de instancias aprovisionadas. Algunos de ellos sirven activamente al tráfico. El resto de ellos (instancias aprovisionadas e inactivas) se mantienen como una reserva de capacidad informática rentable, que está lista para activarse rápidamente. Usted paga por el uso de memoria de todas las instancias aprovisionadas. Usted paga por el uso de CPU sólo del subconjunto activo.

  App Runner duplica temporalmente el número de instancias aprovisionadas durante las implementaciones, para mantener la misma capacidad para código antiguo y nuevo.

- Revisiones— La primera configuración que cree con un nombre obtiene el número de revisión 1. Las configuraciones posteriores con el mismo nombre obtienen números de revisión consecutivos (empezando por 2). Puede asociar su servicio App Runner con una revisión de configuración de escalado automático específica o con la última revisión de configuración.
- Compartido: puede compartir un único recurso de configuración de escalado automático en varios servicios de App Runner. Esto es útil si tienen requisitos de escala similares. En particular, puede configurar varios servicios para que todos utilicen la versión más reciente de una configuración especificando el nombre de la configuración pero no especificando una revisión. Al hacerlo, cualquiera de los servicios que configuró de esta manera recibe actualizaciones de configuración de escalado automático al actualizar el servicio. Para obtener más información acerca de los cambios de configuración, consultethe section called “Configuración”.
- Administración de recursos— Puede utilizar App Runner para crear y eliminar configuraciones de escalado automático. No se puede actualizar directamente una configuración. En su lugar, puede crear una nueva revisión a un nombre de configuración existente para actualizar eficazmente la configuración.

> **Note**
>
> En este momento, solo puede crear una configuración con una sola revisión en la consola de App Runner. Para crear más revisiones y eliminar configuraciones, utilice App RunnerAPI.

- Cuotas de recursos— Hay cuotas establecidas para el número de nombres de configuración únicos y revisiones que puede tener para sus recursos de configuración de escalado automático en cadaRegión de AWS. Si alcanza estas cuotas, debe eliminar un nombre de configuración o al menos algunas de sus revisiones para poder crear más. Usar App RunnerAPIpara eliminarlos. Para obtener más información, consulte the section called “Cuotas de recursos”.

Administrar el escalado automático con la consola de App Runner

CuandoCrear un servicioen la consola de App Runner, puede usar la configuración predeterminada de escalado automático o una configuración personalizada. Para utilizar una configuración personalizada, elija una configuración existente o proporcione un nombre y una configuración nuevos. Si se trata de una nueva configuración, App Runner crea un nuevo recurso de configuración de escalado automático para usted y, a continuación, lo asocia con el nuevo servicio.

Gestione el escalado automático mediante la API de App Runner oAWS CLI

Puede utilizar las siguientes acciones de la API de de App Runner para administrar las configuraciones de Auto Scaling.

- CreateAutoScalingConfiguration— Crea una nueva configuración de escalado automático o una revisión a una existente.
- ListAutoScalingConfigurations— Devuelve una lista de las configuraciones de escalado automático asociadas a suCuenta de AWSCon información resumida.
- DescribeAutoescalingConfiguration— Devuelve una descripción completa de una configuración de escalado automático.
- DeleteAutoescalingConfiguration— Elimina una configuración de escalado automático. Puede eliminar una revisión específica o la última revisión activa. Es posible que deba eliminar configuraciones de escalado automático innecesarias si alcanza la cuota de configuración de escalado automático para suCuenta de AWS.

Administración de nombres de dominio personalizados para un servicio de App Runner

Al crear unAWS App Runner, App Runner le asigna un nombre de dominio. Este es un subdominio en el`awsapprunner.com`que es propiedad de App Runner. Se puede utilizar para acceder a la aplicación web que se está ejecutando en su servicio.

Si dispone de un nombre de dominio, puede asociarlo a su servicio App Runner. Después de que App Runner valida tu nuevo dominio, se puede utilizar para acceder a tu aplicación además del dominio App Runner. Puede asociar hasta cinco dominios personalizados.

> **Note**
>
> Si lo desea, puede incluir el`www`subdominio de su dominio. Sin embargo, en la actualidad sólo se admite en la API. La consola de App Runner no lo admite.

Cuando asocia un dominio personalizado con su servicio, App Runner le proporciona un conjunto de registros de validación de certificados. Añádelos a tu Sistema de nombres de dominio (DNS) para que App Runner pueda validar que eres propietario o controla el dominio. Además, agregue los registros CNAME o ALIAS a su DNS para dirigirse al dominio de App Runner. Debe agregar un registro para el dominio personalizado y otro para el`www`, si elige esta opción. Luego espere a que el estado del dominio personalizado se convierta en*Activa*en la consola de App Runner. Esto suele tardar varios minutos (pero puede tardar entre 24 y 48 horas). En este punto, su dominio personalizado se valida y App Runner comienza a enrutar el tráfico de este dominio a su aplicación web.

Puede especificar un dominio para asociar con el servicio App Runner de las siguientes maneras:

- Un dominio raíz— Por ejemplo,`example.com`. Si lo desea, puede asociar`www.example.com`También como parte de la misma operación.
- Un subdominio— Por ejemplo,`login.example.com`o`admin.login.example.com`. Opcionalmente, puede asociar el`www`como parte de la misma operación.
- Un comodín (`*`)— Por ejemplo,`*.example.com`. No puede usar el`www`En este caso. Puede especificar un comodín solo como el subdominio inmediato de un dominio raíz y solo por sí mismo (estas especificaciones no son válidas:`login*.example.com`,`*.login.example.com`). Esta

especificación comodín asocia todos los subdominios inmediatos y no asocia el dominio raíz en sí (el dominio raíz tendría que estar asociado en una operación separada).

Una asociación de dominio más específica anula una menos específica. Por ejemplo,`login.example.com`Anulaciones`*.example.com`. Se utilizan el certificado y CNAME de la asociación más específica.

En el ejemplo siguiente se muestra cómo se pueden utilizar varias asociaciones de dominio personalizadas:

1. Asociativo`example.com`con la página de inicio de su servicio. Habilitación de`www`para asociar también`www.example.com`.
2. Asociativo`login.example.com`con la página de inicio de sesión de su servicio.
3. Asociativo`*.example.com`con una página personalizada «no encontrada».

Puede desasociar (desvincular) un dominio personalizado del servicio App Runner. Cuando desvincula un dominio, App Runner deja de enrutar el tráfico de este dominio a la aplicación web. Debe eliminar los registros de este dominio de su DNS.

App Runner crea internamente certificados que rastrean la validez del dominio. Están almacenados en AWS Certificate Manager (ACM). App Runner no elimina estos certificados durante siete días después de que un dominio se haya desasociado del servicio o después de que se haya eliminado el servicio.

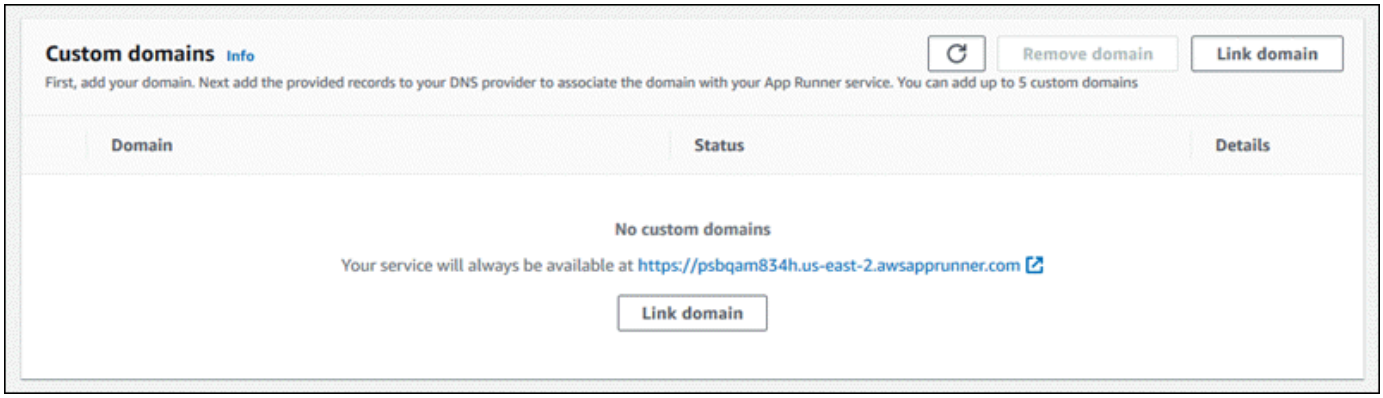
Administrar dominios personalizados con la consola de App Runner

Para asociar (vincular) un dominio personalizado mediante la consola de App Runner

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, elijaServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.
3. En la página del panel de servicio, elija la opciónDominios personalizadosPestaña.

   La consola muestra los dominios personalizados asociados con el servicio, oNo hay dominios personalizados.



4. En la páginaDominios personalizados, elijaDominio de enlace.
5. En el navegadorVincular dominio personalizado, escriba un nombre de dominio y, a continuación, elijaVer configuración de DNS.
6. Siga las instrucciones de laConfiguración de DNSpara iniciar el proceso de validación del dominio.
7. Cuando el estado del dominio cambia aActiva, compruebe que el dominio funciona para enrutar el tráfico navegando hasta él.

Para desasociar (desvincular) un dominio personalizado mediante la consola de App Runner

1. En la páginaDominios personalizados, seleccione el mosaico del dominio que desea desasociar y, a continuación, elijaDesvinculación de dominio.
2. En el navegadorDesvinculación de dominio, verifique la acción seleccionandoDesvinculación de dominio.

Administrar dominios personalizados mediante la API de App Runner oAWS CLI

Para asociar un dominio personalizado con su servicio mediante la API de App Runner oAWS CLI, llame alAssociateCustomDomainAcción de la API. Cuando la llamada se realiza correctamente, devuelve unCustomDomainque describe el dominio personalizado que se está asociando con el servicio. El objeto debe mostrar un estado de`CREATING`y contiene una lista deCertificateValidationRecordObjetos. Estos son registros que puede agregar a su DNS.

Para desasociar un dominio personalizado del servicio mediante la API de App Runner oAWS CLI, llame alDisasociateCustomDomainAcción de la API. Cuando la llamada se realiza correctamente,

devuelve unCustomDomainque describe el dominio personalizado que se está desasociando del servicio. El objeto debe mostrar un estado de`DELETING`.

Pausar y reanudar un servicio de App Runner

Si necesita desactivar su aplicación web temporalmente y detener la ejecución del código, puede pausar suAWS App Runnerservicio App Runner reduce la capacidad informática del servicio a cero.

Cuando esté listo para ejecutar la aplicación de nuevo, puede reanudar el servicio App Runner. App Runner aprovisiona nueva capacidad informática, implementa la aplicación en ella y ejecuta la aplicación. El origen de la aplicación no se vuelve a desplegar y no es necesario compilar. En su lugar, App Runner se reanuda con la versión actualmente implementada. La aplicación conserva su dominio App Runner.

Important

- Cuando pausa el servicio, la aplicación pierde su estado. Por ejemplo, cualquier almacenamiento efímero que el código utilizado se pierde. Para su código, pausar y reanudar el servicio equivale a implementar en un nuevo servicio.
- Si pausa un servicio debido a una falla en el código (por ejemplo, un error detectado o un problema de seguridad), no puede implementar una nueva versión antes de reanudar el servicio.

  Por lo tanto, le recomendamos que mantenga el servicio en ejecución y que vuelva a su última versión estable de la aplicación en su lugar.
- Al reanudar el servicio, App Runner implementa la última versión de la aplicación que se usó antes de pausar el servicio. Si ha agregado alguna nueva versión de origen desde que ha pausado el servicio, App Runner no las implementa automáticamente incluso si se ha seleccionado la implementación automática. Por ejemplo, suponga que tiene nuevas versiones de imagen en el repositorio de imágenes o nuevas confirmaciones en el repositorio de código. Estas versiones no se implementan automáticamente.

  Para implementar una versión más reciente, realice una implementación manual o agregue otra versión al repositorio de origen después de reanudar el servicio App Runner.

Pausa y eliminación comparadas

Pausesu servicio App Runner atemporalmenteLo deshabilita. Solo se terminan los recursos informáticos y los datos almacenados (por ejemplo, la imagen del contenedor con la versión de la aplicación) permanecen intactos. Reanudar el servicio es rápido: su aplicación está lista para implementarse en nuevos recursos informáticos. Su dominio de App Runner sigue siendo el mismo.

Eliminarsu servicio App Runner aPermanentementeElimine. Los datos almacenados se eliminan. Si necesita recrear el servicio, App Runner necesita recuperar su fuente nuevamente, y también construirlo si se trata de un repositorio de código. Su aplicación web obtiene un nuevo dominio de App Runner.

Cuando su servicio está en pausa

Cuando pausa el servicio y está en elPaused, responde de manera diferente a las solicitudes de acción, incluidas las llamadas API o las operaciones de consola. Cuando un servicio está en pausa, puede realizar acciones de App Runner que no modifiquen la definición o configuración del servicio de una manera que afecte a su tiempo de ejecución. En otras palabras, si una acción cambia el comportamiento, la escala u otras características de un servicio en ejecución, no puede realizar esa acción en un servicio en pausa.

Las siguientes listas proporcionan información sobre las acciones de API que puede y no puede realizar en un servicio en pausa. Las operaciones de consola equivalentes están permitidas o denegadas de forma similar.

Acciones quecanrealizar en un servicio en pausa

- *`List*`*y*`Describe*`*Acciones de— Acciones que solo leen información.
- *`DeleteService`*— Siempre puede eliminar un servicio.
- *`TagResource`*,*`UntagResource`*— Las etiquetas están asociadas a un servicio, pero no forman parte de su definición y no afectan a su comportamiento en tiempo de ejecución.

Acciones quecannotrealizar en un servicio en pausa

- *`StartDeployment`*Acciones de(o unImplementación manualUso de la consola)
- *`UpdateService`*(o un cambio de configuración usando la consola, excepto los cambios de etiquetado)
- *`CreateCustomDomainAssociations`*, *`DeleteCustomDomainAssociations`*

- *`CreateConnection`*, *`DeleteConnection`*

Pausa y reanuda el servicio con la consola de App Runner

Para pausar el servicio mediante la consola de App Runner

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, seleccioneServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.
3. SeleccionarActionsHaga clic en y luego enPause.

   En la página del panel de servicio, el servicioEstadocambios enOperación en cursoy, a continuación, cambia aPaused. Su servicio ahora está en pausa.

Para reanudar el servicio mediante la consola de App Runner

1. SeleccionarActionsHaga clic en y luego enResume.

   En la página del panel de servicio, el servicioEstadocambios enOperación en curso.
2. Espere a que se reanude el servicio. En la página del panel de servicio, el servicioEstadocambia de nuevo aEn ejecución.
3. Para comprobar que la reanudación del servicio es correcta, en la página del panel de servicio, seleccione la opciónDominio de App RunnerValor . Es la URL del sitio web de tu servicio. Compruebe que la aplicación web se está ejecutando correctamente.

Detenga y reanude su servicio mediante la API de App Runner oAWS CLI

Para pausar el servicio mediante la API de App Runner oAWS CLI, llame a laServicio de pauseServiceAcción de la API. Si la llamada devuelve una respuesta correcta con unService (Servicio)objeto que muestra`"Status": "OPERATION_IN_PROGRESS"`, App Runner comienza a pausar el servicio.

Para reanudar el servicio mediante la API de App Runner oAWS CLI, llame a laResumeServiceAcción de la API. Si la llamada devuelve una respuesta correcta con unService (Servicio)objeto que muestra`"Status": "OPERATION_IN_PROGRESS"`, App Runner comienza a reanudar el servicio.

Eliminación de un servicio App Runner

Cuando desee terminar la aplicación web que se está ejecutando en suAWS App RunnerPuede eliminar el servicio. Al eliminar un servicio se detiene el servicio web en ejecución, se eliminan los recursos subyacentes y se eliminan los datos asociados.

Puede decidir eliminar un servicio de App Runner por uno o varios de los motivos siguientes:

- Ya no necesita la aplicación web— Por ejemplo, está retirado o es una versión de desarrollo que has terminado de usar.
- Ha llegado a la cuota de servicio de App RunnerDesea crear un nuevo servicio en el mismoRegión de AWSy has alcanzado la cuota asociada a tu cuenta. Para obtener más información, consulte the section called “Cuotas de recursos”.
- Consideraciones de seguridad o privacidad: desea que App Runner elimine los datos que almacena para su servicio.

Pausar frente a eliminar

Pausesu servicio App Runner atemporalmenteDeshabilítalo. Solo se terminan los recursos informáticos y los datos almacenados (por ejemplo, la imagen del contenedor con la versión de la aplicación) permanecen intactos. Reanudar el servicio es rápido: su aplicación está lista para implementarse en nuevos recursos informáticos. Su dominio de App Runner sigue siendo el mismo.

Eliminarsu servicio App Runner aPermanentementeElimine. Los datos almacenados se eliminan. Si necesita recrear el servicio, App Runner necesita recuperar su fuente nuevamente, y también construirlo si se trata de un repositorio de código. Su aplicación web obtiene un nuevo dominio de App Runner.

¿Qué elimina App Runner?

Cuando eliminas el servicio, App Runner elimina algunos elementos asociados y no elimina otros. Las listas siguientes proporcionan los detalles.

Elementos que App Runner elimina:

- Imagen de contenedor— Una copia de la imagen que implementó o de la imagen que App Runner creó a partir del código fuente. Se almacena en Amazon Elastic Container Registry (Amazon ECR) medianteCuentas de AWSque son propiedad de App Runner.

- Configuración de servicio: los ajustes de configuración asociados con el servicio App Runner. Están almacenados en Amazon DynamoDB medianteCuentas de AWSque son propiedad de App Runner.

Elementos que App Runner no elimina:

- Conexión a— Es posible que tenga una conexión asociada a su servicio. Una conexión de App Runner es un recurso independiente que puede compartirse entre varios servicios de App Runner. Si ya no necesita la conexión, puede eliminarla explícitamente. Para obtener más información, consulte the section called “Conexiones”.
- Certificados de dominio personalizados: si vincula dominios personalizados a un servicio de App Runner, App Runner crea internamente certificados que rastrean la validez del dominio. Están almacenados enAWS Certificate Manager(ACM). App Runner no elimina el certificado durante siete días después de que un dominio se desvincula del servicio o después de que se elimine el servicio. Para obtener más información, consulte the section called “Nombres de dominio personalizados”.

Eliminar el servicio mediante la consola de App Runner

Para eliminar el servicio mediante la consola de App Runner

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, seleccioneServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.
3. Elija Actions (Acciones) y, a continuación, Delete (Eliminar).

   La consola le lleva a laServicios(Se ha creado el certificado). El servicio eliminado muestra elOperación en cursoy, a continuación, el servicio desaparece de la lista. Su servicio ahora se ha eliminado.

Elimine su servicio mediante la API de App Runner oAWS CLI

Para eliminar el servicio mediante la API de App Runner oAWS CLI, llame aDeleteServiceAcción de la API. Si la llamada devuelve una respuesta correcta con unService (Servicio)objeto que muestra`"Status": "OPERATION_IN_PROGRESS"`, App Runner comienza a eliminar el servicio.

Registro y monitoreo de App Runner

AWS App Runnerse integra con variosAWSpara proporcionarle un amplio conjunto de herramientas de registro y supervisión para su servicio App Runner. Temas de este capítulo describen estas capacidades.

Temas

Seguimiento de la actividad del servicio App Runner

AWS App Runnerutiliza una lista de operaciones para realizar un seguimiento de la actividad en el servicio App Runner. Una operación representa una llamada asincrónica a una acción de API, como la creación de un servicio, la actualización de una configuración y la implementación de un servicio. Las secciones siguientes muestran cómo realizar un seguimiento de la actividad en la consola de App Runner y utilizar la API de.

Seguimiento de la actividad del servicio App Runner en la consola

La consola de App Runner muestra la actividad de servicio de App Runner y proporciona más formas de explorar las operaciones.

Para ver la actividad de su servicio

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, seleccioneServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.
3. En la página del panel de servicio, seleccione la opciónActividadEn caso de que aún no esté seleccionada.

   La consola de muestra una lista de operaciones.

4. Para buscar operaciones específicas, introduzca un término de búsqueda en la lista. Puede buscar cualquier valor que aparezca en la tabla.
5. Elija cualquier operación de la lista para ver o descargar el registro relacionado.

Recuperación de operaciones de servicio App Runner mediante la API de App Runner oAWS CLI

LaListOperationsEn vista del nombre de recurso de Amazon (ARN) de un servicio de App Runner, se devuelve una lista de operaciones que se han producido en este servicio. Cada elemento de la lista contiene un ID de operación y algunos detalles de seguimiento.

Visualización de registros de App Runner transmitidos a CloudWatch Logs

Puede utilizar los Amazon CloudWatch Logs para supervisar, almacenar y acceder a los archivos de registro que los recursos de variosAWSservicios generados. Para obtener más información, consulteGuía del usuario de Amazon CloudWatch Logs.

AWS App Runnerrecopila el resultado de las implementaciones de aplicaciones y del servicio activo y lo transmite a CloudWatch Logs. En las siguientes secciones se enumeran los flujos de registro de App Runner y se muestra cómo verlos en la consola de App Runner.

Streams y grupos de registro de App Runner

CloudWatch Logs mantiene los datos de registro en flujos de registro que organiza en grupos de registros. AFlujo de registroes una secuencia de eventos de registro de un origen específico. Un grupo de registros es un grupo de flujos de registros que comparten la misma configuración de retención, monitorización y control de acceso.

App Runner define dos grupos de registro de CloudWatch Logs, cada uno con múltiples flujos de registro, para cada uno de los servicios de App Runner en suCuenta de AWS.

Servicios de registro

El grupo de registro de servicios contiene la salida de registro generada por App Runner a medida que administra el servicio de App Runner y actúa sobre él.

nombre de grupo de registro	Ejemplo
`/aws/apprunner/` *`service-name`* `/` *`service-id`* `/service`	`/aws/apprunner/python-test/ac7ec8b51ff34746bcb6654e0bcb23da/service`

Dentro del grupo de registro de servicios, App Runner crea un flujo de registro de eventos para capturar la actividad durante el ciclo de vida del servicio App Runner. Por ejemplo, esto podría estar iniciando la aplicación o pausándola.

Además, App Runner crea un flujo de registro para cada operación asíncrona de larga duración relacionada con su servicio. El nombre del flujo de registro refleja el tipo de operación y el identificador de operación específico.

AImplementación dees un tipo de operación. Los registros de implementación contienen el resultado de registro de los pasos de compilación e implementación que App Runner realiza al crear un servicio o implementar una nueva versión de la aplicación. Los nombres de flujo de registro de implementación comienzan con`deployment/`y finalice con el ID de la operación que realiza la implementación. Esta operación es unaCreateServicepara la implementación inicial de la aplicación oStartDeploymentpara cada despliegue posterior.

Dentro de un registro de implementación, cada mensaje de registro comienza con un prefijo:

- `[AppRunner]`— Salida que App Runner genera durante la implementación.
- `[Build]`— Salida de sus propios scripts de compilación.

Nombre del flujo de registros	Ejemplo
`events`	N/A (nombre fijo)
`operation-type` `/` *`operation-id`*	`deployment/c2c8eeedea164f459cf78f12a8953390`

Logs de aplicaciones

El grupo de registro de aplicaciones contiene el resultado del código de aplicación en ejecución.

nombre de grupo de registro	Ejemplo
`/aws/apprunner/ service-name /service-id /application`	`/aws/apprunner/python-test/ ac7ec8b51ff34746bcb6654e0bcb23da/ application`

Dentro del grupo de registro de aplicaciones, App Runner crea un flujo de registro para cada instancia (unidad de escala) que ejecuta la aplicación.

Nombre del flujo de registros	Ejemplo
`instance/ instance-id`	`instance/1a80bc9134a84699b7 b3432ebeebb591`

Ver los registros de App Runner en la consola de

La consola de App Runner muestra un resumen de todos los registros de su servicio y le permite verlos, explorarlos y descargarlos.

Para ver los registros del servicio

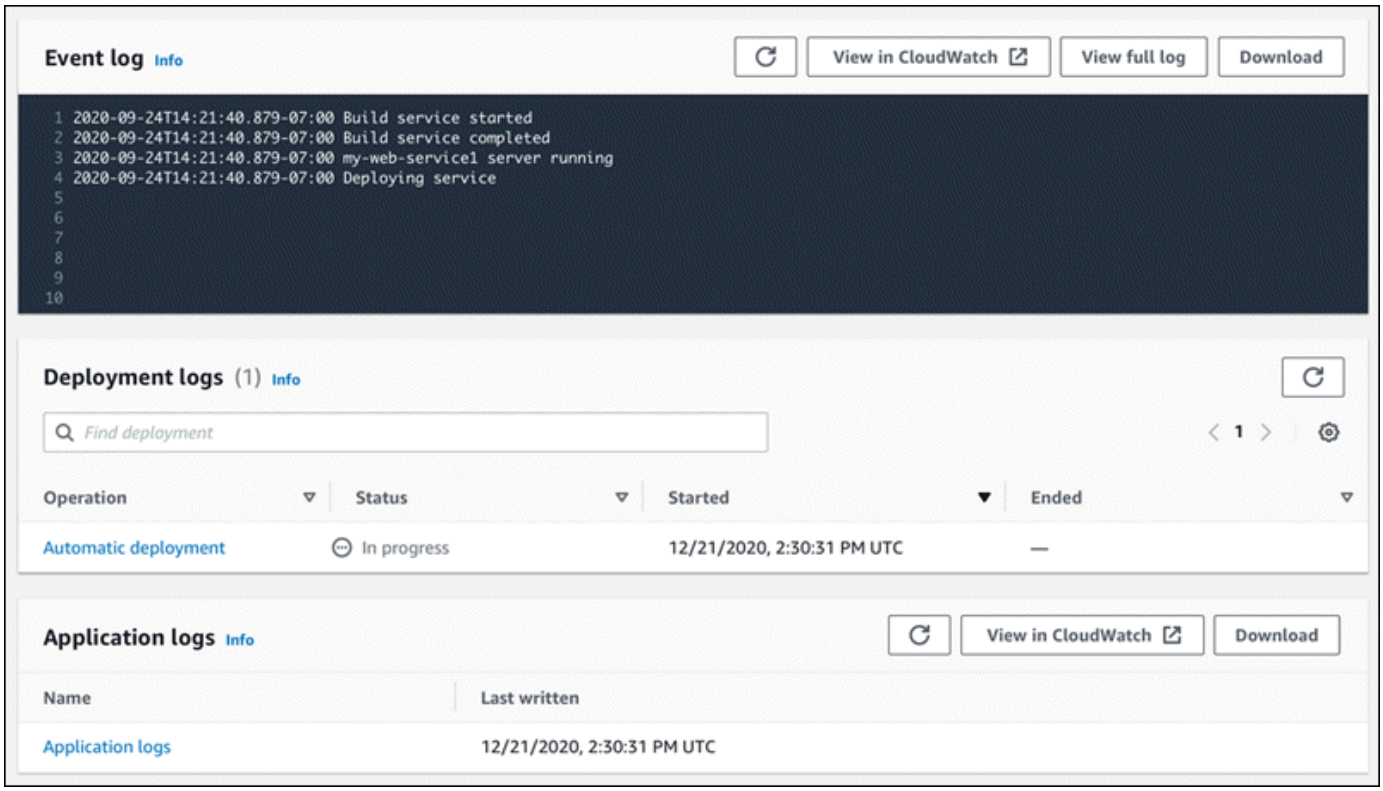
1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, elijaServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unInformación general del servicio.
3. En la página del panel de servicio, elija la opciónRegistrosPestaña.

   La consola muestra algunos tipos de registros en varias secciones:

   - Registro de eventos— Actividad en el ciclo de vida de tu servicio App Runner. La consola muestra los eventos más recientes.
   - Registros de implementación— Implementaciones de repositorio de origen en su servicio App Runner. La consola muestra una secuencia de registro independiente para cada implementación.

- Logs de aplicaciones— El resultado de la aplicación web que se implementa en el servicio App Runner. La consola combina la salida de todas las instancias en ejecución en un único flujo de registro.



4. Para buscar implementaciones específicas, introduzca un término de búsqueda en la lista de registros de implementación. Puede buscar cualquier valor que aparezca en la tabla.
5. Para ver el contenido de un registro, elijaPara ver el registro completo(registro de eventos) o el nombre del flujo de registro (registros de implementación y aplicación).
6. SeleccionarDescargarpara descargar un registro. Para una secuencia de registro de implementación, seleccione primero una secuencia de registro.
7. SeleccionarVer en CloudWatchpara abrir la consola de CloudWatch y utilizar todas sus capacidades para explorar los registros de servicio de App Runner. Para una secuencia de registro de implementación, seleccione primero una secuencia de registro.

> **Note**
>
> La consola de CloudWatch es especialmente útil si desea ver los registros de aplicaciones de instancias específicas en lugar del registro combinado de aplicaciones.

Visualización de métricas de servicio de App Runner notificadas a CloudWatch

Amazon CloudWatch monitoriza sus Amazon Web Services (AWS) y las aplicaciones que ejecuta enAWSEn tiempo real. Puede utilizar CloudWatch para recopilar y hacer un seguimiento de métricas, que son las variables que puede medir en sus recursos y aplicaciones. También puede utilizar para crear alarmas que vean métricas. Cuando se alcanza un determinado umbral, CloudWatch envía notificaciones o realiza cambios automáticamente en los recursos supervisados. Para obtener más información, consulteGuía del usuario de Amazon CloudWatch.

AWS App Runnerrecopila una variedad de métricas que te proporcionan una mayor visibilidad sobre el uso, el rendimiento y la disponibilidad de tus servicios de App Runner. Algunas métricas realizan un seguimiento de instancias individuales que ejecutan su servicio web, mientras que otras se encuentran en el nivel de servicio general. En las siguientes secciones se enumeran las métricas de App Runner y se muestra cómo verlas en la consola de App Runner.

Métricas de App Runner

App Runner recopila las siguientes métricas relacionadas con su servicio y las publica en CloudWatch en el`AWS/AppRunner`espacio de nombres.

Métricas de nivel de instanciase recopilan para cada instancia (unidad de escala) individualmente.

¿Qué se mide?	Métrica	Descripción
CPU utilization	`CPUUtilization`	El uso medio de la CPU durante periodos de un minuto.
Memory utilization	`MemoryUtilization`	El uso medio de memoria durante periodos de un minuto.

Métricas de nivel de serviciosSe recopilan para todo el servicio.

¿Qué se mide?	Métricas	Descripción
HTTP request count	`Requests`	Número de solicitudes HTTP que recibió el servicio.

¿Qué se mide?	Métricas	Descripción

HTTP status counts	`2xxStatus Responses`
	
	`4xxStatus Responses`
	
	`5xxStatus Responses`
		Número de solicitudes HTTP que devolvieron cada estado de respuesta, agrupadas por categoría (2XX, 4XX, 5XX).

HTTP request latency	`RequestLatency`	El tiempo que tardó su servicio web en procesar solicitudes HTTP.
Instance counts	`ActiveInstances`	Número de instancias que procesan solicitudes HTTP del servicio.

Consulta de métricas de App Runner en la consola de

La consola de App Runner muestra gráficamente las métricas que App Runner recopila para su servicio y proporciona más formas de explorarlas.

> **Note**
>
> En este momento, la consola solo muestra métricas de servicio. Para ver las métricas de instancia, utilice la consola de CloudWatch.

Para ver los registros del servicio

1. Abra el iconoConsola de App Runner, y en elRegiones, seleccione suRegión de AWS.
2. En el panel de navegación, elijaServiciosy, a continuación, elige tu servicio App Runner.

   La consola muestra el panel de servicio con unDescripción general del servicio.
3. En la página del panel de servicios, seleccione la opciónMétricasPestaña.

   La consola muestra un conjunto de gráficos de métricas.

4. Elija una duración (por ejemplo,12h) para los gráficos de métricas de alcance al período reciente de esa duración.
5. SeleccionarAgregar al panel deen la parte superior de una de las secciones de gráficos, o utilice el menú de cualquier gráfico, para agregar las métricas relevantes a un panel de la consola de CloudWatch para una investigación más detallada.

Gestión de eventos de App Runner en EventBridge

Con Amazon EventBridge, puede configurar reglas basadas en eventos que monitoreen un flujo de datos en tiempo real desde suAWS App Runnerpara ciertos patrones. Cuando se coincide un patrón para una regla, EventBridge inicia una acción en un destino comoAWS Lambda, Amazon ECS,AWS Batchy Amazon SNS. Por ejemplo, puede establecer una regla para el envío de notificaciones por correo electrónico al señalizar un tema de Amazon SNS siempre que se produzca un error en la implementación del servicio. O bien, puede configurar una función de Lambda para que notifique a un canal de Slack siempre que se produzca un error en una actualización de servicio. Para obtener más información sobre EventBridge, consulte.Guía del usuario de Amazon EventBridge.

App Runner envía los siguientes tipos de eventos a EventBridge

- Cambio del estado de los servicios— Un cambio en el estado de un servicio de App Runner. Por ejemplo, un estado de servicio cambió a`DELETE_FAILED`.
- Cambio del estado— Un cambio en el estado de una operación larga y asincrónica en un servicio App Runner. Por ejemplo, un servicio comenzó a crear, una actualización de servicio completada correctamente o una implementación de servicio completada con errores.

Crear una regla de EventBridge para actuar en eventos de App Runner

EventBridgeeventes un objeto que define algunos campos estándar de EventBridge, como elAWSy el tipo de detalle (evento), y un conjunto de campos específicos del evento con los detalles del evento. Para crear una regla de EventBridge, utilice la consola EventBridge para definir unpatrón de eventos(qué eventos deben realizar un seguimiento de la apuesta) y especificar unacción de destino(lo que se debe hacer en un partido). Un patrón de evento es similar a los eventos que coincide. Especifique un subconjunto de campos para que coincida y, para cada campo, especifique una lista de valores posibles. En este tema se proporcionan ejemplos de eventos y patrones de eventos de App Runner.

Para obtener más información sobre la creación de reglas EventBridge, consulte.Crear una regla para unAWSServicioen laGuía del usuario de Amazon EventBridge.

> **Note**
>
> Algunos servicios de apoyoPatrones predefinidosen EventBridge. Esto simplifica la forma en que se crea un patrón de eventos. Los valores de campo se seleccionan en un formulario y EventBridge genera el patrón por usted. En este momento, App Runner no admite patrones predefinidos. Debe ingresar el patrón como un objeto JSON. Puede utilizar los ejemplos de este tema como punto de partida.

Ejemplos de eventos de App Runner

Estos son algunos ejemplos de eventos que App Runner envía a EventBridge.

- Evento de cambio de estado de servicio. Específicamente, un servicio que cambió de`OPERATION_IN_PROGRESS`a la`RUNNING`Estado.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "Service Status Change",
  "source": "aws.apprunner",
  "account": "111122223333",
  "time": "2021-04-29T11:54:23Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:apprunner:us-east-2:123456789012:service/my-
app/8fe1e10304f84fd2b0df550fe98a71fa"
  ],
  "detail": {
    "PreviousStatus": "OPERATION_IN_PROGRESS",
    "CurrentStatus": "RUNNING",
    "ServiceName": "my-app",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "Message": "Service status is set to RUNNING.",
    "Severity": "INFO"
  }
}
```

- Evento de cambio de estado de una operación. En concreto, un`UpdateService`La operación se ha completado correctamente.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "Service Operation Status Change",
  "source": "aws.apprunner",
  "account": "111122223333",
  "time": "2021-04-29T18:43:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:apprunner:us-east-2:123456789012:service/my-
app/8fe1e10304f84fd2b0df550fe98a71fa"
  ],
  "detail": {
    "Status": "UpdateServiceCompletedSuccessfully",
    "ServiceName": "my-app",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "Message": "Service update completed successfully. New application and
 configuration is deployed.",
    "Severity": "INFO"
  }
}
```

Ejemplos de patrones de eventos de App Runner

En los siguientes ejemplos se muestran los patrones de eventos que se pueden utilizar en las reglas de EventBridge para hacer coincidir uno o varios eventos de App Runner. Un patrón de evento es similar a un evento. Incluya sólo los campos que desee que coincidan y proporcione una lista en lugar de una escala a cada uno.

- Hacer coincidir todos los eventos de cambio de estado del servicio para los servicios de una cuenta específica, donde el servicio ya no está en`RUNNING`Estado.

```
{
  "detail-type": [ "Service Status Change" ],
  "source": [ "aws.apprunner" ],
  "account": [ "111122223333" ],
  "detail": {
    "PreviousStatus": [ "RUNNING" ]
```

```
  }
}
```

- Hacer coincidir todos los eventos de cambio de estado de la operación para los servicios de una cuenta específica, donde la operación falló.

```
{
  "detail-type": [ "Service Operation Status Change" ],
  "source": [ "aws.apprunner" ],
  "account": [ "111122223333" ],
  "detail": {
    "Status": [
      "CreateServiceFailed",
      "DeleteServiceFailed",
      "UpdateServiceFailed",
      "DeploymentFailed",
      "PauseServiceFailed",
      "ResumeServiceFailed"
    ]
  }
}
```

Referencia de eventos de App Runner

Cambio del estado de los servicios

Un evento de cambio de estado de un servicio tiene `detail-type` se establece en `Service Status Change`. Tiene los siguientes campos de detalle y valores:

```
"PreviousStatus": "any valid service status",
"CurrentStatus": "any valid service status",
"ServiceName": "your service name",
"ServiceId": "your service ID",
"Message": "Service status is set to CurrentStatus.",
"Severity": "varies"
```

Cambio del estado de la operación

Un evento de cambio de estado de una operación tiene `detail-type` se establece en `Service Operation Status Change`. Tiene los siguientes campos de detalle y valores:

```
"Status": "see following table",
"ServiceName": "your service name",
"ServiceId": "your service ID",
"Message": "see following table",
"Severity": "varies"
```

En la siguiente tabla se muestran todos los códigos de estado posibles y los mensajes relacionados.

Estado	Mensaje
`CreateServiceStarted`	La creación del servicio se ha iniciado
`CreateServiceCompletedSuccessfully`	La creación del servicio se ha completado correctamente.
`CreateServiceFailed`	Error en la creación del servicio. Para obtener más información, consulte los registros de servicio.
`DeleteServiceStarted`	Eliminación de servicio iniciada.
`DeleteServiceCompletedSuccessfully`	La eliminación del servicio se ha completado correctamente.
`DeleteServiceFailed`	Error al eliminar el servicio.
`UpdateServiceStarted`	
`UpdateServiceCompletedSuccessfully`	La actualización del servicio se ha completado correctamente. Se implementan nuevas aplicaciones y configuraciones.
	La actualización del servicio se ha completado correctamente. Se implementa una nueva configuración.
`UpdateServiceFailed`	Error en la actualización del servicio. Para obtener más información, consulte los registros de servicio.
`DeploymentStarted`	Implementación iniciada.
`DeploymentCompletedSuccessfully`	La implementación se ha completado correctamente.

Estado	Mensaje
`DeploymentFailed`	Error en la implementación. Para obtener más información, consulte los registros de servicio.
`PauseServiceStarted`	La pausa del servicio se ha iniciado.
`PauseServiceCompletedSuccessfully`	La pausa del servicio se ha completado correctamente.
`PauseServiceFailed`	Error en la pausa del servicio.
`ResumeServiceStarted`	reanudación del servicio iniciada.
`ResumeServiceCompletedSuccessfully`	El proceso se ha completado correctamente.
`ResumeServiceFailed`	Error de reanudación del servicio.

Registrar llamadas a la API de Runner conAWS CloudTrail

App Runner está integrado conAWS CloudTrail, un servicio que proporciona un registro de las medidas adoptadas por un usuario, un rol o unAWSen App Runner. CloudTrail captura como eventos todas las llamadas a la API de App Runner. Las llamadas capturadas incluyen las llamadas realizadas desde la consola de App Runner y las llamadas de código realizadas a las operaciones de la API de App Runner. 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 App Runner. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Event history (Historial de eventos). Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a App Runner, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc.

Para obtener más información sobre CloudTrail, consulte laAWS CloudTrailGuía del usuario.

Información de App Runner en CloudTrail

CloudTrail está habilitado en suCuenta de AWSCuando se crea la cuenta de. Cuando se produce una actividad en App Runner, dicha actividad se registra en un evento de CloudTrail junto con los demásAWSeventos de servicios enHistorial de eventos. Puede ver, buscar y descargar los últimos

eventos en laCuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail Event.

Para mantener un registro continuo de los eventos de laCuenta de AWSPara mantener un registro de seguimiento, incluidos los eventos de App Runner. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todos los usuarios deRegiones de AWS. El registro de seguimiento registra los eventos de todas las regiones deAWSy envía los archivos de registro al bucket de Amazon S3 especificado. Además, puede configurar otrosAWSPara analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Consulte Servicios e integraciones compatibles con CloudTrail.
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de registro de CloudTrail de varias regiones y Recepción de archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas las acciones de App Runner y se documentan en laAWS App RunnerReferencia de la API de. Por ejemplo, las llamadas a las acciones `CreateService`, `DeleteConnection` y `StartDeployment` generan entradas en los archivos de log de CloudTrail.

Cada entrada de registro o evento contiene información acerca de quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario AWS Identity and Access Management (IAM) o credenciales de usuario raíz.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte CloudTrail userIdentity Element.

Descripción de las entradas de archivos de registro de App Run

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registro en un bucket de Amazon S3 que especifique. Los archivos log de CloudTrail pueden

contener una o varias entradas de log. Un evento representa una única solicitud de cualquier origen e incluye información sobre la acción solicitada, la fecha y la hora de la acción y los parámetros de la solicitud. 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 ejemplo siguiente se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateService`.

> **Note**
>
> Por razones de seguridad, algunos valores de propiedad se escriben en los registros y se reemplazan por el texto`HIDDEN_DUE_TO_SECURITY_REASONS`. Esto evita la exposición no intencionada de información secreta. Sin embargo, aún puede ver que estas propiedades se pasaron en la solicitud o se devolvieron en la respuesta.

Ejemplo de entrada de registro de CloudTrail para**`CreateService`**Acción de App Runner

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/aws-user",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "aws-user",
  },
  "eventTime": "2020-10-02T23:25:33Z",
  "eventSource": "apprunner.amazonaws.com",
  "eventName": "CreateService",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/77.0.3865.75 Safari/537.36",
  "requestParameters": {
    "serviceName": "python-test",
    "sourceConfiguration": {
      "codeRepository": {
        "repositoryUrl": "https://github.com/github-user/python-hello",
        "sourceCodeVersion": {
          "type": "BRANCH",
```

```
          "value": "main"
        },
        "codeConfiguration": {
          "configurationSource": "API",
          "codeConfigurationValues": {
            "runtime": "python3",
            "buildCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
            "startCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
            "port": "8080",
            "runtimeEnvironmentVariables": "HIDDEN_DUE_TO_SECURITY_REASONS"
          }
        }
      },
      "autoDeploymentsEnabled": true,
      "authenticationConfiguration": {
        "connectionArn": "arn:aws:apprunner:us-east-2:123456789012:connection/your-
connection/e7656250f67242d7819feade6800f59e"
      }
    },
    "healthCheckConfiguration": {
      "protocol": "HTTP"
    },
    "instanceConfiguration": {
      "cpu": "256",
      "memory": "1024"
    }
  },
  "responseElements": {
    "service": {
      "serviceName": "python-test",
      "serviceId": "dfa2b7cc7bcb4b6fa6c1f0f4efff988a",
      "serviceArn": "arn:aws:apprunner:us-east-2:123456789012:service/python-test/
dfa2b7cc7bcb4b6fa6c1f0f4efff988a",
      "serviceUrl": "generated domain",
      "createdAt": "2020-10-02T23:25:32.650Z",
      "updatedAt": "2020-10-02T23:25:32.650Z",
      "status": "OPERATION_IN_PROGRESS",
      "sourceConfiguration": {
        "codeRepository": {
          "repositoryUrl": "https://github.com/github-user/python-hello",
          "sourceCodeVersion": {
            "type": "Branch",
            "value": "main"
          },
```

```
          "codeConfiguration": {
            "codeConfigurationValues": {
            "configurationSource": "API",
              "runtime": "python3",
              "buildCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
              "startCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
              "port": "8080",
              "runtimeEnvironmentVariables": "HIDDEN_DUE_TO_SECURITY_REASONS"
            }
          }
        },
        "autoDeploymentsEnabled": true,
        "authenticationConfiguration": {
          "connectionArn": "arn:aws:apprunner:us-east-2:123456789012:connection/your-
connection/e7656250f67242d7819feade6800f59e"
        }
      },
      "healthCheckConfiguration": {
        "protocol": "HTTP",
        "path": "/",
        "interval": 5,
        "timeout": 2,
        "healthyThreshold": 3,
        "unhealthyThreshold": 5
      },
      "instanceConfiguration": {
        "cpu": "256",
        "memory": "1024"
      },
      "autoScalingConfigurationSummary": {
        "autoScalingConfigurationArn": "arn:aws:apprunner:us-
east-2:123456789012:autoscalingconfiguration/
DefaultConfiguration/1/00000000000000000000000000000001",
        "autoScalingConfigurationName": "DefaultConfiguration",
        "autoScalingConfigurationRevision": 1
      }
    }
  },
  "requestID": "1a60af60-ecf5-4280-aa8f-64538319ba0a",
  "eventID": "e1a3f623-4d24-4390-a70b-bf08a0e24669",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
```

```
}
```

Establecer las opciones de servicio de App Runner mediante un archivo de configuración

> **Note**
>
> Los archivos de configuración sólo se aplican aservicios basados en el código fuente. No puede usar archivos de configuración conservicios basados en imágenes.

Al crear un un unAWS App RunnerEl servicio de un repositorio de código fuente,AWS App Runnerrequiere información sobre cómo construir e iniciar el servicio. Puede proporcionar esta información cada vez que cree un servicio mediante la consola o la API de App Runner. De forma alternativa, puede definir las opciones de servicio mediante unArchivo de configuración. Las opciones que especifique en un archivo pasan a formar parte del repositorio de origen y cualquier cambio realizado en estas opciones se realiza de manera similar al seguimiento de los cambios realizados en el código fuente. Puede utilizar el archivo de configuración de App Runner para especificar más opciones de las que admite la API. No es necesario que proporcione un archivo de configuración si solo necesita las opciones básicas que admite la API.

El archivo de configuración de App Runner es un archivo YAML denominado`apprunner.yaml`En el directorio raíz del repositorio de su aplicación. Proporciona opciones de compilación y tiempo de ejecución para su servicio. Los valores de este archivo indican a App Runner cómo crear e iniciar el servicio, y proporcionan contexto de tiempo de ejecución, como la configuración de red y las variables de entorno.

El archivo de configuración de App Runner no incluye ajustes operativos, como CPU y memoria.

Para ver ejemplos de archivos de configuración de App Runner, consultethe section called “Ejemplos”. Para obtener una guía de referencia completa, consultethe section called “Referencia”.

Temas

- Ejemplos de archivos de configuración de App Runner
- Referencia del archivo de configuración de App Runner

Ejemplos de archivos de configuración de App Runner

> **Note**
>
> Los archivos de configuración solo se aplican aservicios basados en el código fuente. No puede usar archivos de configuración conservicios basados en imágenes.

En los ejemplos siguientes se ilustraAWS App RunnerArchivos de configuración. Algunos son mínimos y sólo contienen los ajustes necesarios. Otras están completas, incluidas todas las secciones de archivos de configuración. Para obtener información general sobre los archivos de configuración de App Runner, consulteArchivo de configuración de App Runner.

Archivos de configuración

Archivo de configuración mínimo

Con un archivo de configuración mínimo, App Runner hace las siguientes suposiciones:

- No se necesitan variables de entorno personalizadas durante la compilación o ejecución.
- Se utiliza la última versión de tiempo de ejecución.
- Se utilizan el número de puerto predeterminado y la variable de entorno de puerto.

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    build:
      - pip install pipenv
      - pipenv install
run:
  command: python app.py
```

Archivo de configuración completo

Este ejemplo muestra el uso de todas las claves de configuración con un tiempo de ejecución administrado.

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    pre-build:
      - wget -c https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/test-lib.tar.gz -O - | tar
 -xz
    build:
      - pip install pipenv
      - pipenv install
    post-build:
      - python manage.py test
  env:
    - name: DJANGO_SETTINGS_MODULE
      value: "django_apprunner.settings"
    - name: MY_VAR_EXAMPLE
      value: "example"
run:
  runtime-version: 3.7.7
  command: pipenv run gunicorn django_apprunner.wsgi --log-file -
  network:
    port: 8000
    env: MY_APP_PORT
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
```

Para obtener ejemplos de archivos de configuración de tiempo de ejecución administrado específicos, consulte el tema secundario de tiempo de ejecución específico enServicio basado en código de.

Referencia del archivo de configuración de App Runner

> **Note**
>
> Los archivos de configuración sólo se aplican aservicios basados en el código fuente. No puede utilizar archivos de configuración conservicios basados en imágenes.

Este tema es una guía de referencia completa sobre la sintaxis y la semántica de unAWS App RunnerArchivo de configuración. Para obtener información general sobre los archivos de configuración de App Runner, consulteArchivo de configuración de App Runner.

El archivo de configuración de App Runner es un archivo YAML. Nómbraloapprunner.yamlY colóquelo en el directorio raíz del repositorio de la aplicación.

Información general acerca de

El archivo de configuración de App Runner es un archivo YAML. Nómbraloapprunner.yamlY colóquelo en el directorio raíz del repositorio de la aplicación.

El archivo de configuración de App Runner contiene las siguientes partes principales:

- Sección superiorContiene claves de nivel superior
- Sección de compilación— Configura la etapa de compilación
- Sección Ejecutar— Configura la etapa de tiempo de ejecución

Sección superior

Las claves de la parte superior del archivo proporcionan información general sobre el archivo y el tiempo de ejecución del servicio. Están disponibles las siguientes claves:

- `version`–Obligatorio. La versión del archivo de configuración de App Runner. Lo ideal es utilizar la última versión de.

  Sintaxis

  ```
  version: version
  ```

  Example

  ```
  version: 1.0
  ```

- `runtime`–Obligatorio. El nombre del motor de ejecución que utiliza la aplicación. Los siguientes tiempos de ejecución están disponibles actualmente:

  `python3`, `nodejs12`

> **Note**
>
> La convención de nomenclatura de un tiempo de ejecución administrado es*<language-name> <major-version>*.

Sintaxis

```
runtime: runtime-name
```

Example

```
runtime: python3
```

Sección de compilación

La sección de compilación configura la etapa de compilación de la implementación del servicio App Runner. Puede especificar comandos de compilación y variables de entorno. Se requieren comandos de compilación.

La sección comienza con el`build:`y tiene las siguientes subclaves:

- `commands`–Obligatorio. Especifica los comandos que App Runner ejecuta durante varias fases de compilación. Incluye las siguientes subclaves:
  - `pre-build`–Opcional. Los comandos que App Runner ejecuta antes de la compilación. Por ejemplo, instalenpmdependencias o bibliotecas de prueba.
  - `build`–Obligatorio. Los comandos que App Runner ejecuta para crear la aplicación. Por ejemplo, usepipenv.
  - `post-build`–Opcional. Los comandos que App Runner ejecuta después de la compilación. Por ejemplo, use Maven para empaquetar artefactos de la compilación en un archivo JAR o WAR, o ejecute una prueba.

Sintaxis

```
build:
  commands:
```

```
    pre-build:
      - command
      - …
    build:
      - command
      - …
    post-build:
      - command
      - …
```

Example

```
build:
  commands:
    pre-build:
      - yum install openssl
    build:
      - pip install -r requirements.txt
    post-build:
      - python manage.py test
```

- env–Opcional. Especifica variables de entorno personalizadas para la etapa de compilación. Definido como asignaciones escalares de nombre y valor. Puede hacer referencia a estas variables por nombre en los comandos de compilación.

  Sintaxis

```
build:
  env:
    - name: name1
      value: value1
    - name: name2
      value: value2
    - …
```

  Example

```
build:
  env:
    - name: DJANGO_SETTINGS_MODULE
      value: "django_apprunner.settings"
    - name: MY_VAR_EXAMPLE
```

```
      value: "example"
```

Sección Ejecutar

La sección de ejecución configura la etapa de ejecución del contenedor de la implementación de la aplicación App Runner. Puede especificar la versión en tiempo de ejecución, el comando start, el puerto de red y las variables de entorno.

La sección comienza con el`run:`y tiene las siguientes subclaves:

- `runtime-version`–Opcional. Especifica una versión en tiempo de ejecución que desea bloquear para el servicio App Runner.

  De forma predeterminada, sólo la versión principal está bloqueada. App Runner utiliza las últimas versiones secundarias y de parches que están disponibles para el tiempo de ejecución en cada implementación o actualización de servicio. Si especifica versiones principales y secundarias, ambas se bloquean y App Runner solo actualiza las versiones de parches. Si especifica versiones principales, secundarias y de parches, el servicio está bloqueado en una versión de tiempo de ejecución específica y App Runner nunca lo actualiza.

  Sintaxis

  ```
  run:
    runtime-version: major[.minor[.patch]]
  ```

  Example

  ```
  runtime: python3
  run:
    runtime-version: 3.7
  ```

- `command`–Obligatorio. El comando que App Runner utiliza para ejecutar la aplicación después de completar la compilación de la aplicación.

  Sintaxis

  ```
  run:
    command: command
  ```

- `network`–Opcional. Especifica el puerto que escucha la aplicación. Contiene lo siguiente:
  - `port`–Opcional. Si se especifica, este es el número de puerto que su aplicación está a la escucha. El valor predeterminado es `8080`.
  - `env`–Opcional. Si se especifica, App Runner pasa el número de puerto al contenedor de esta variable de entorno, además de pasar (no en lugar de) el mismo número de puerto en la variable de entorno predeterminada,`PORT`. En otras palabras, si especifica`env`, App Runner pasa el número de puerto en dos variables de entorno.

  Sintaxis

  ```
  run:
    network:
      port: port-number
      env: env-variable-name
  ```

  Example

  ```
  run:
    network:
      port: 8000
      env: MY_APP_PORT
  ```

- `env`–Opcional. Definición de variables de entorno personalizadas para la etapa de ejecución. Definido como asignaciones escalares de nombre y valor. Puede hacer referencia a estas variables por su nombre en su entorno de tiempo de ejecución.

  Sintaxis

  ```
  run:
    env:
      - name: name1
        value: value1
      - name: name2
        value: value2
      - …
  ```

  Example

  ```
  run:
    env:
  ```

```
    - name: MY_VAR_EXAMPLE
      value: "example"
```

La API de App Runner

LaAWS App Runnerinterfaz de programación de aplicaciones (API) es una API RESTful para realizar solicitudes al servicio App Runner. Puede usar la API para crear, enumerar, describir, actualizar y eliminar recursos de App Runner de enCuenta de AWS.

Puede llamar a la API directamente en el código de su aplicación, o puede usar uno de losAWSSDK. Para las secuencias de comandos de línea de comandos, utilice laAWS CLIpara realizar llamadas al servicio App Runner. Para obtener más información acerca deAWSPara desarrolladores, consulteHerramientas para construirAWS.

Para obtener información de referencia de la API completa, consulte laAWS App RunnerReferencia de la API.

Seguridad en App Runner

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y de centros de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube–AWSes responsable de proteger la infraestructura que ejecutaAWSen elNube de AWS.AWSTambién 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 Programas de conformidad de AWS . Para obtener más información acerca de los programas de conformidad que se aplican aAWS App Runner, consulteAWSServicios de conformidad en el ámbito del programa de conformidad.
- Seguridad en la nube— Su responsabilidad está determinada por elAWSque utilice. 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 ayuda a comprender cómo puede aplicar el modelo de responsabilidad compartida cuando se utiliza App Runner. En los siguientes temas, se le mostrará cómo configurar App Runner para que cumpla sus objetivos de seguridad y conformidad. También aprenderá a utilizar otrosAWSque le ayudan a monitorear y proteger los recursos de App Runner.

Temas

Protección de datos en App Runner

LaAWS Modelo de responsabilidad compartidaSe aplica a la protección de datos enAWS App Runner. Como se describe en este modelo,AWSes responsable de proteger la infraestructura global en la que se ejecutan todos losAWS- Nube. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Este contenido incluye la configuración de seguridad y las tareas de administración deAWSLos servicios de que utilice. Para obtener más información acerca de la privacidad de datos, consulte las Preguntas frecuentes sobre privacidad de datos. Para obtener más información sobre la protección de datos en Europa, consulte laAWSModelo de responsabilidad compartida y RGPDPublicación del blog en laAWSBlog de seguridad.

Para fines de protección de datos, recomendamos protegerAWSy configure cuentas de usuario individuales conAWS Identity and Access Management(IAM). De esta manera,solo se otorgan a cada usuario los permisos necesarios para cumplir con sus obligaciones laborales. También le recomendamos proteger sus datos de las siguientes formas:

- Utilice Multi-Factor Authentication (MFA) con cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Le recomendamos TLS 1.2 o una versión posterior.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de AWS.
- Utilice avanzados servicios de seguridad administrados, como Amazon Macie, que le ayuden a detectar y proteger los datos personales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de enlace de FIPS. Para obtener más información acerca de los puntos de enlace de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

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 Name (Nombre). Esto incluye cuando trabaje con App Runner u otrosAWSutilizando la consola, API,AWS CLI, o bienAWSSDK. Cualquier dato que escriba en App Runner o en otros servicios podría incluirse 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 consultar otros temas de seguridad de App Runner, consulteSeguridad.

Temas

- Protección de datos mediante cifrado
- Privacidad del tráfico entre redes
- Uso de App Runner con puntos de enlace de la VPC

Protección de datos mediante cifrado

AWS App Runnerlee el origen de la aplicación (imagen de origen o código fuente) desde un repositorio que especifique y lo almacena para su implementación en el servicio. Para obtener más información, consulte Arquitectura y conceptos.

La protección de datos se refiere a la protección de datos mientrasEn tránsito(mientras viaja hacia y desde App Runner) yEn reposo(mientras está almacenado enAWScentros de datos).

Para obtener más información sobre la protección de datos, consultethe section called “Protección de los datos”.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Cifrado en tránsito

Puede conseguir la protección de datos en tránsito de dos formas: cifre la conexión mediante Seguridad de capa de transporte (TLS) o use cifrado del lado del cliente (donde el objeto se cifra antes de enviarse). Ambos métodos son válidos para proteger los datos de la aplicación. Para proteger la conexión, cífrela mediante TLS siempre que su aplicación, sus desarrolladores y administradores, y sus usuarios finales envíen o reciban cualquier objeto. App Runner configura tu aplicación para recibir tráfico a través de TLS.

El cifrado del lado del cliente no es un método válido para proteger la imagen de origen o el código que proporciona a App Runner para su implementación. App Runner necesita acceso a la fuente de la aplicación, por lo que no se puede cifrar. Por tanto, asegúrese de proteger la conexión entre su entorno de implementación o desarrollo y App Runner.

Cifrado en reposo y administración de claves

Para proteger los datos de la aplicación en reposo, App Runner cifra todas las copias almacenadas de la imagen de origen de la aplicación o paquete de origen. Al crear un servicio de App Runner,

puede proporcionar una clave maestra de cliente (CMK). Si proporciona una, App Runner utiliza la clave proporcionada para cifrar su fuente. Si no proporcionan una, App Runner utiliza unaAWSCMK administrada por.

Para obtener más información sobre los parámetros de creación del servicio App Runner, consulteCreateService. Para obtener información acerca deAWS Key Management Service(AWS KMS), consulte laAWS Key Management ServiceGuía para desarrolladores.

Privacidad del tráfico entre redes

App Runner usa Amazon Virtual Private Cloud (Amazon VPC) para crear límites entre los recursos de la aplicación de App Runner y controlar el tráfico entre ellos, la red local e Internet. Para obtener más información sobre la seguridad de Amazon VPC, consultePrivacidad del tráfico entre redes en Amazon VPCen laGuía del usuario de Amazon VPC.

Para obtener información acerca de cómo proteger las solicitudes a App Runner mediante un punto de enlace de la VPC, consultethe section called “Puntos de conexión de la VPC”.

Para obtener más información sobre la protección de datos, consultethe section called “Protección de los datos”.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Uso de App Runner con puntos de enlace de la VPC

SusAWSaplicación podría integrarAWS App RunnerServicios con otrosAWSque se ejecutan en unAmazon Virtual Private Cloud (Amazon VPC). Algunas partes de la aplicación pueden realizar solicitudes a App Runner desde la VPC. Por ejemplo, podría usarAWS CodePipelinepara implementar continuamente en su servicio App Runner. Una forma de mejorar la seguridad de su aplicación es enviar estas solicitudes de App Runner (y solicitudes a otrosAWS) a través de un extremo de VPC.

Un punto de enlace de la VPC le permite conectar de forma privada su VPC a los servicios de AWS compatibles y a servicios de punto de enlace de la VPC basados en AWS PrivateLink sin necesidad de una gateway de internet, un dispositivo NAT, una conexión de VPN o una conexión de AWS Direct Connect.

Los recursos de la VPC no utilizan direcciones IP públicas para interactuar con recursos de App Runner. El tráfico entre su VPC y App Runner no sale de la red de Amazon. Para obtener

información detallada sobre los puntos de enlace de la VPC, consultePuntos de conexión de la VPCen laAWS PrivateLinkGuía.

Compatibilidad con App RunnerAWS PrivateLink, que proporciona conectividad privada a App Runner y elimina la exposición del tráfico al internet público. Para permitir que su aplicación envíe solicitudes a App Runner medianteAWS PrivateLink, se configura un tipo de endpoint VPC conocido comopunto de enlace de la VPC de la interfaz(punto final de interfaz). Para obtener más información, consultePuntos de enlace de la VPC de interfaz (AWS PrivateLink)en laAWS PrivateLinkGuía.

> **Note**
>
> La aplicación web del servicio App Runner se ejecuta en una VPC que App Runner proporciona y configura. Esta VPC es pública, está conectada a Internet pública. App Runner no admite la ejecución de la aplicación en una VPC privada ni la creación de un extremo de VPC para ella.

Configuración de un punto de enlace de la VPC para App Runner

Para crear el punto de enlace de la VPC de la interfaz para el servicio de App Runner en la VPC, siga laCreación de un punto de enlace de interfazEn laAWS PrivateLinkGuía. En Service Name (Nombre de servicio), seleccione `com.amazonaws.`*`region`*`.apprunner`,

Consideraciones sobre VPC privacidad de la red de

> **Important**
>
> El uso de un extremo de VPC para App Runner no garantiza que todo el tráfico de su VPC permanezca fuera de Internet. Es posible que la VPC sea pública (conectada a Internet) y que algunas partes de la solución no usen endpoints de VPC para hacerAWSLlamadas a la API. Por ejemplo,AWSLos servicios de pueden llamar a otros servicios mediante sus puntos de enlace públicos. Si se requiere privacidad de tráfico para la solución en su VPC, lea esta sección.

Para garantizar la privacidad del tráfico de red en la VPC, tenga en cuenta lo siguiente:

- Habilitar nombre de DNS— Es posible que algunas partes de su aplicación sigan enviando solicitudes a App Runner a través de Internet mediante

elapprunner.*region*.amazonaws.comPunto de enlace público. Si la VPC está configurada con acceso público a Internet, estas solicitudes se realizan correctamente sin ninguna indicación. Puede evitarlo asegurándose de que Enable DNS name (Habilitar nombre de la DNS) esté habilitado durante la creación del punto de enlace (está establecido como “true” de forma predeterminada). Esto añade una entrada DNS en la VPC que asigna el punto de enlace del servicio público al punto de enlace de la VPC de tipo interfaz.

- Configuración de puntos de enlace de la VPC para servicios adicionales— Su solución podría enviar solicitudes a otrosAWSServicios de . Por ejemplo,AWS CodePipelinepodría enviar solicitudes aAWS CodeBuild. Configure los puntos de enlace de la VPC para estos servicios y habilite los nombres de DNS en estos puntos de enlace.

Uso de políticas de punto de enlace para controlar el acceso con puntos de enlace de la VPC

De forma predeterminada, un punto de enlace de la VPC permite el acceso completo al servicio al que está asociado. Al crear o modificar un punto de enlace de la VPC para App Runner, puede adjuntar unaPolítica de punto de enlace.

Una política de endpoint es unAWS Identity and Access Management(IAM) que controla el acceso desde el punto de enlace al servicio especificado. La política de punto final es específica del punto final. Es independiente de cualquier política de usuario o instancia de IAM que pueda tener su entorno y no las reemplaza ni las sobrescribe. Para obtener información detallada sobre la creación y el uso de políticas de punto de enlace de la VPC, consulteControl del acceso a los servicios con punto de enlace de la VPCen laAWS PrivateLinkGuía.

En el siguiente ejemplo se permite el acceso de sólo lectura desde el extremo de VPC a App Runner. Estos son derechos de acceso mínimos cuando se utiliza el extremo de VPC. Los principales pueden tener permisos adicionales a través de otras directivas.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apprunner:List*",
        "apprunner:Describe*"
      ],
      "Resource": "*"
    }
```

```
  ]
}
```

Administración de identidades y accesos para App Runner

AWS Identity and Access Management(IAM) es unaAWSque ayuda a un administrador a controlar de forma segura el acceso aAWSde AWS. Los administradores de IAM controlan quién puede serAutenticado(iniciado sesión) yAutorizado(tienen permisos) para usar los recursos de App Runner. IAM es unAWSServicio que puede utilizar sin cargo adicional.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Temas

- Audience
- Autenticación con identidades
- Administración de acceso mediante políticas
- Cómo funciona App Runner con IAM
- Ejemplos de políticas basadas en identidad de App Runner
- Uso de roles vinculados a servicios en App Runner
- Políticas administradas por AWS para AWS App Runner
- Solución de problemas de identidad y acceso de App Runner

Audience

Cómo se utilizaAWS Identity and Access Management(IAM) difiere en función del trabajo que realice en App Runner.

Usuario del servicio: si utiliza el servicio App Runner para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de App Runner para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos a su administrador. Si no puede acceder a una característica de App Runner, consulteSolución de problemas de identidad y acceso de App Runner.

Administrador de serviciosSi está a cargo de los recursos de App Runner en su empresa, probablemente tenga acceso completo a App Runner. Su trabajo consiste en determinar a qué

características y recursos de App Runner pueden acceder sus empleados. A continuación, 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 sobre cómo su empresa puede utilizar IAM con App Runner, consulteCómo funciona App Runner con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más información sobre cómo escribir políticas para administrar el acceso a App Runner. Para ver ejemplos de políticas basadas en identidades de App Runner que puede utilizar en IAM, consulteEjemplos de políticas basadas en identidad de App Runner.

Autenticación con identidades

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Para obtener más información acerca de cómo iniciar sesión con laAWS Management Console, consulteInicie sesión enAWS Management ConsoleComo usuario raíz o usuario de IAMen laGuía del usuario de IAM.

Debe serAutenticado(iniciado sesión enAWS) como elAWSUn usuario raíz de la cuenta de, un usuario de IAM o asumiendo un rol de IAM. También puede utilizar la autenticación de inicio de sesión único de la empresa o incluso iniciar sesión con Google o Facebook. En estos casos, su administrador habrá configurado previamente la federación de identidad mediante roles de IAM. Cuando obtiene acceso a AWS mediante credenciales de otra empresa, asume un rol indirectamente.

Para iniciar sesión directamente en elAWS Management Console, utilice su contraseña con su dirección de correo electrónico del usuario raíz o su nombre de usuario de IAM. Puede acceder aAWSMediante programación con las claves de acceso del usuario raíz o del usuario de IAM.AWSproporciona SDK y herramientas de línea de comandos para firmar criptográficamente su solicitud con sus credenciales. Si no utiliza las herramientas de AWS, debe firmar usted mismo la solicitud. Para ello, utilice Signature Version 4, un protocolo para autenticar solicitudes de API de entrada. Para obtener más información acerca de la autenticación de solicitudes, consulteProceso de firma de Signature Version 4en laAWSReferencia general de.

Independientemente del método de autenticación que utilice, es posible que también deba proporcionar información de seguridad adicional. Por ejemplo, AWS le recomienda el uso de la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulteUso de la autenticación multifactor (MFA) enAWSen laGuía del usuario de IAM.

Usuario raíz de la cuenta de AWS

Cuando se crea por primera vez una cuenta de AWS, se comienza con una única identidad de inicio de sesión que tiene acceso completo a todos los servicios y recursos de AWS de la cuenta. Esta identidad se denominaAWSaccountUsuario raízPara obtener acceso a ella, inicia sesión con la dirección de correo electrónico y la contraseña que utilizó para crear la cuenta. Le recomendamos que no utilice el usuario raíz en sus tareas cotidianas, incluso las tareas administrativas. En lugar de ello, es mejor ceñirse a la práctica recomendada de utilizar el usuario final exclusivamente para crear al primer usuario de IAM. A continuación, guarde las credenciales del usuario raíz en un lugar seguro y utilícelas tan solo para algunas tareas de administración de cuentas y servicios.

Usuarios y grupos de IAM

UnaUsuario de IAMes una identidad dentro de suAWSUna cuenta de que dispone de permisos específicos para una sola persona o aplicación. Un usuario de IAM puede tener credenciales a largo plazo, como un nombre de usuario y una contraseña o un conjunto de claves de acceso. Para obtener información sobre cómo generar claves de acceso, consulte Administración de claves de acceso de los usuarios de IAM en la Guía del usuario de IAM. Al generar claves de acceso para un usuario de IAM, asegúrese de ver y guardar de forma segura el par de claves. No puede recuperar la clave de acceso secreta en el futuro. En su lugar, debe generar un nuevo par de claves de acceso.

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 obtener más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

IAM roles

UnaRol de IAMes una identidad dentro de suAWSLa cuenta de que dispone de permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente un rol de IAM en laAWS Management Consolepor[cambiar roles](). Puede asumir un rol llamando a una operación de la AWS CLI o de la API de AWS, o utilizando una URL personalizada.

Para obtener más información acerca de 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:

- 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 de usuarios federados: en lugar de crear un usuario de IAM, puede utilizar identidades existentes deAWS Directory Service, el directorio de usuarios de la compañía o un proveedor de identidades web. A estas identidades se les llama usuarios federados. AWS asigna una función a un usuario federado cuando se solicita acceso a través de un proveedor de identidad. Para obtener más información acerca de los usuarios federados, consulte Usuarios federados y roles en la Guía del usuario de IAM.
- 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 algunos servicios de AWS, puede asociar una política directamente a un recurso (en lugar de utilizar 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— AlgunosAWSutilizan características en otrosAWSServicios de . 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 a servicios.
  - Permisos principales: cuando utiliza un usuario o un rol de IAM para realizar acciones enAWS, se le considera un director. Las políticas conceden permisos a una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. En este caso, debe tener permisos para realizar ambas acciones. Para ver si una acción requiere acciones dependientes adicionales en una política, consulteAcciones, recursos y claves de condiciones paraAWS App Runneren laReferencia de autorizaciones de servicio.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Los roles de servicio ofrecen acceso solo dentro de su cuenta y no se pueden utilizar para otorgar acceso a servicios en otras cuentas. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información,

consulteCreación de un rol para delegar permisos a unAWSServicioen laGuía del usuario de IAM.

- Función vinculada al servicio— Un rol vinculado a un servicio es un tipo de rol de servicio vinculado a unAWSServicio El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM 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 utilizar un rol de IAM para administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia EC2 y realizanAWS CLIorAWSSolicitudes de la API. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar un rol de AWS a una instancia EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia asociado a la misma. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para obtener 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

Puede controlar el acceso enAWSAl crear políticas y adjuntándolas a identidades de IAM oAWSde AWS. Una política es un objeto de AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. Puede iniciar sesión como usuario raíz o usuario de IAM o puede asumir un rol de IAM. Cuando haga una solicitud,AWSevalúa las políticas relacionadas basadas en identidades o en recursos. Los permisos en las políticas determinan si la solicitud se permite o se deniega. Las mayoría de las políticas se almacenan en 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 usarAWSPolíticas JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y bajo qué condiciones.

Cada entidad de IAM (usuario o rol) comienza sin permisos. En otras palabras, de forma predeterminada, los usuarios no pueden hacer nada, ni siquiera cambiar sus propias contraseñas. Para conceder permiso a un usuario para hacer algo, el administrador debe asociarle una política

de permisos. O bien el administrador puede añadir al usuario a un grupo que tenga los permisos necesarios. Cuando el administrador concede permisos a un grupo, todos los usuarios de ese grupo obtienen los permisos.

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 dicha política puede obtener información del usuario de la AWS Management Console, la AWS CLI o la API de AWS.

Políticas basadas en identidad

Las políticas basadas en identidades 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 bajo qué condiciones. Para obtener más información sobre cómo crear una política basada en identidades, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidad 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 asociar a varios usuarios, grupos y roles de su cuenta de AWS. Las políticas administradas incluyen las políticas administradas por AWS y las políticas administradas por el cliente. Para obtener más información acerca de 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 basada en recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados oAWSServicios de .

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puede usarAWSPolíticas administradas 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 política JSON.

Amazon S3,AWS WAFy Amazon VPC son ejemplos de servicios compatibles con las ACL. 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 otros tipos de políticas menos frecuentes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le otorgan.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidades puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una identidad. Los permisos resultantes son la intersección de las políticas basadas en identidades de la entidad y los límites de sus permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, 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 JSON que especifican los permisos máximos para una organización o unidad organizativa (OU) enAWS Organizations.AWS Organizationses un servicio que le permite agrupar y administrar de forma centralizadaAWSLas cuentas de que es propietario de su negocio. Si habilita todas las funciones en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o todas sus cuentas.
  Una SCP limita los permisos para las entidades de las cuentas de miembros, incluido cadaAWSUsuario raíz de la cuenta de. Para obtener más información acerca de Organizations y SCP, consulteFuncionamiento de las políticas SCPen laAWS OrganizationsGuía del usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política basada en recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener 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 cómoAWSdetermina si permitir una solicitud cuando hay varios tipos de políticas implicados, consulteLógica de evaluación de políticasen laGuía del usuario de IAM.

Cómo funciona App Runner con IAM

Antes de utilizar IAM para administrar el acceso aAWS App Runner, debe saber qué características de IAM están disponibles para su uso con App Runner. Para obtener una perspectiva general de cómo App Runner y otrosAWSfuncionan con IAM, consulteAWSServicios que funcionan con IAMen laGuía del usuario de IAM.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Temas

- Políticas basadas en identidad de App Runner
- Políticas basadas en recursos de App Runner
- Autorización basada en etiquetas de App Runner
- Permisos de usuario de App Runner
- Roles de IAM de aplicación

Políticas basadas en identidad de App Runner

Con las políticas basadas en identidades de IAM, puede especificar las acciones y recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. App Runner admite acciones, claves de condiciones y recursos específicos. Para obtener información sobre todos los elementos que utiliza en una política JSON, consulte Referencia de los elementos de las políticas JSON de IAM en la Guía del usuario de IAM.

Actions

Los administradores pueden usarAWSPolíticas JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y bajo qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo

nombre que la operación de API de AWS 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 para realizar la operación asociada.

Las acciones de políticas de App Runner utilizan el siguiente prefijo antes de la acción: `apprunner:`. Por ejemplo, para conceder a alguien permiso para ejecutar una instancia de Amazon EC2 con la`RunInstances`, se incluye la operación API`ec2:RunInstances`acción en su política. Las instrucciones de la política deben incluir un elemento `Action` o un elemento `NotAction`. App Runner define su propio conjunto de acciones que describen las tareas que se pueden realizar con este servicio.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": [
   "apprunner:CreateService",
   "apprunner:CreateConnection"
]
```

Puede utilizar caracteres comodín para especificar varias acciones (*). Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "apprunner:Describe*"
```

Para ver una lista de las acciones de App Runner, consulteAcciones definidas porAWS App Runneren laReferencia de autorizaciones de servicio.

Resources

Los administradores pueden usarAWSPolíticas JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y bajo 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": "*"
```

Los recursos de App Runner tienen la siguiente estructura ARN:

```
arn:aws:apprunner:region:account-id:resource-type/resource-name[/resource-id]
```

Para obtener más información acerca del formato de los ARN, consulteNombres de recursos de Amazon (ARN) yAWSEspacios de nombres de servicios deen laAWSReferencia general de.

Por ejemplo, para especificar el`my-service`En su instrucción, utilice el siguiente ARN:

```
"Resource": "arn:aws:apprunner:us-east-1:123456789012:service/my-service"
```

Para especificar todos los servicios que pertenecen a una cuenta específica, utilice el carácter comodín (*):

```
"Resource": "arn:aws:apprunner:us-east-1:123456789012:service/*"
```

Algunas acciones de App Runner, como las empleadas para la creación de recursos, no se pueden llevar a cabo en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

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

Para ver una lista de los tipos de recursos de App Runner y sus ARN, consulteRecursos definidos porAWS App Runneren laReferencia de autorizaciones de servicio. Para obtener más información acerca de con qué acciones puede especificar los ARN de cada recurso, consulteAcciones definidas porAWS App Runner.

Claves de condición

Los administradores pueden usarAWSPolíticas JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y bajo 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 `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica `OR`. 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 obtener más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWSadmite claves de condición globales y claves de condición específicas del servicio. Para ver todas lasAWSClaves de condición globales de, consulteAWSClaves de contexto de condición globales deen laGuía del usuario de IAM.

App Runner admite el uso de algunas claves de condición globales. Para ver todas lasAWSClaves de condición globales de, consulteAWSClaves de contexto de condición globales deen laGuía del usuario de IAM.

App Runner define un conjunto de claves de condición específicas del servicio. Además, App Runner admite el control de acceso basado en etiquetas, que se implementa utilizando claves de condición. Para obtener más información, consulte the section called “Autorización basada en etiquetas de App Runner”.

Para ver una lista de claves de condición de App Runner, consulteClaves de condición deAWS App Runneren laReferencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulteAcciones definidas porAWS App Runner.

Examples

Para ver ejemplos de políticas basadas en identidades de App Runner, consulteEjemplos de políticas basadas en identidad de App Runner.

Políticas basadas en recursos de App Runner

App Runner no admite políticas basadas en recursos.

Autorización basada en etiquetas de App Runner

Puede asociar etiquetas a los recursos de App Runner o transferirlas en una solicitud a App Runner. 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 `apprunner:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Para obtener más información acerca del etiquetado de recursos de App Runner, consultethe section called “Configuración”.

Para consultar un ejemplo de política basada en la identidad para limitar el acceso a un recurso en función de las etiquetas de ese recurso, consulte Control del acceso a los servicios de App Runner basado en etiquetas.

Permisos de usuario de App Runner

Para usar App Runner, los usuarios de IAM necesitan permisos para las acciones de App Runner. Una forma común de conceder permisos a los usuarios es adjuntando una directiva a usuarios o grupos de IAM. Para obtener más información acerca de la administración de permisos de los usuarios, consulteCambio de permisos para un usuario de IAMen laGuía del usuario de IAM.

App Runner proporciona dos políticas administradas por que puede asociar a los usuarios de.

- `AWSAppRunnerReadOnlyAccess`: concede permisos para mostrar y ver detalles sobre los recursos de App Runner.
- `AWSAppRunnerFullAccess`— Otorga permisos a todas las acciones de App Runner.

Para obtener un control más detallado de los permisos de usuario, puede crear una directiva personalizada y adjuntarla a los usuarios. Para obtener más información, consulte .Creación de directivas de IAMen laGuía del usuario de IAM.

Para ver algunos ejemplos de políticas de usuario de, consultethe section called “Políticas de usuario”.

AWSAppRunnerReadOnlyAccess

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

```
      "Effect": "Allow",
      "Action": [
        "apprunner:List*",
        "apprunner:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWSAppRunnerFullAccess

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/apprunner.amazonaws.com/
AWSServiceRoleForAppRunner",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "apprunner.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "apprunner.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:CreateGrant"
      ],
      "Resource": "*"
```

```
    },
    {
      "Sid": "Administrative permission over AppRunner applications",
      "Effect": "Allow",
      "Action": "apprunner:*",
      "Resource": "*"
    }
  ]
}
```

Roles de IAM de aplicación

UnaRol de IAMes una entidad dentro de suCuenta de AWSque tiene permisos específicos.

Roles vinculados a servicios

Los roles vinculados a servicios permiten a los servicios de AWS obtener acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

App Runner admite roles vinculados a servicios Para obtener más información acerca de cómo crear o administrar roles vinculados a un servicio de App Runner, consultethe section called “Uso de roles vinculados a servicios”.

Roles de servicio

Esta característica permite que un servicio asuma un rol de servicio en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su cuenta de IAM y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

App Runner admite algunos roles de servicio.

Rol de acceso

La función de acceso es una función que App Runner utiliza para obtener acceso a imágenes de Amazon Elastic Container Registry (Amazon ECR) de su cuenta. Se requiere para acceder a una imagen en Amazon ECR y no se requiere con Amazon ECR Public. Antes de crear un servicio basado en una imagen en Amazon ECR, utilice IAM para crear un rol de servicio y

utilice la funciónAWSAppRunnerServicePolicyForECRAccesspolítica administrada en ella. A continuación, puede pasar este rol a App Runner cuando llame al métodoCreateServiceAPI de en laAuthenticationConfiguration(un miembro de laSourceConfiguration) o cuando se utiliza la consola de App Runner para crear un servicio.

AWSAppRunnerServicePolicyForECRAccess

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:DescribeImages",
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> Si crea su propia directiva personalizada para su rol de acceso, asegúrese de especificar`"Resource": "*"`para la`ecr:GetAuthorizationToken`action. Puede utilizar los tokens de para acceder a cualquier registro de Amazon ECR al que tenga acceso.

Cuando cree su rol de acceso, asegúrese de agregar una política de confianza que declare el principal de servicio de App Runner`build.apprunner.amazonaws.com`Como entidad de confianza.

Política de confianza para un rol de acceso

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

```
      "Effect": "Allow",
      "Principal": {
        "Service": "build.apprunner.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Si utiliza la consola de App Runner para crear un servicio, la consola puede crear automáticamente un rol de acceso para usted y elegirlo para el nuevo servicio. La consola también enumera otros roles en su cuenta, y puede seleccionar un rol diferente si lo desea.

Rol de instancia

El rol de instancia es un rol opcional que App Runner utiliza para proporcionar permisos aAWSa las que llama el código de la aplicación. Antes de crear un servicio App Runner, utilice IAM para crear un rol de servicio con los permisos que necesita el código de aplicación. A continuación, puede pasar este rol a App Runner en la sección[CreateService](#)o cuando se utiliza la consola de App Runner para crear un servicio.

Cuando cree su rol de instancia, asegúrese de agregar una política de confianza que declare el principal del servicio App Runner`tasks.apprunner.amazonaws.com`Como entidad de confianza.

Política de confianza para un rol de instancia

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

Si utiliza la consola de App Runner para crear un servicio, la consola enumera los roles de su cuenta y puede seleccionar el rol que creó para este fin.

Para obtener información acerca de la creación de un servicio, consultethe section called “Creación”.

> **Note**
>
> Los permisos que debe proporcionar el rol de instancia dependen enteramente de la aplicación. Es posible que su código no llame a ningúnAWS, y en este caso no es necesario proporcionar un rol de instancia a App Runner.
> En caso de que su código llame aAWSAPI, no tenemos forma de anticipar qué llamadas son. Por lo tanto, no proporcionamos una política administrada para los roles de instancia. Debe incluir explícitamente los permisos necesarios en su rol de instancia o crear su propia directiva personalizada y utilizarla en el rol de instancia.

Ejemplos de políticas basadas en identidad de App Runner

De forma predeterminada, los usuarios y roles de IAM no tienen permiso para crear ni modificarAWS App Runnerde AWS. Tampoco pueden realizar tareas mediante la AWS Management Console, la AWS CLI, o la API de AWS. Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM con estos documentos de políticas JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Temas

- Prácticas recomendadas relativas a políticas
- Políticas de usuario
- Control del acceso a los servicios de App Runner basado en etiquetas

Prácticas recomendadas relativas a políticas

Las políticas basadas en identidad son muy eficaces. Determinan si alguien puede crear, acceder o eliminar recursos de App Runner en su cuenta. Estas acciones pueden generar costes adicionales

para su cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidad:

- Introducción al uso deAWSPolíticas administradas por— Para empezar a usar App Runner rápidamente, usaAWSPara proporcionar a los empleados los permisos necesarios. Estas políticas ya están disponibles en su cuenta y las mantiene y actualiza AWS. Para obtener más información, consulteIntroducción sobre el uso de permisos conAWSPolíticas administradas poren laGuía del usuario de IAM.
- Conceder privilegios mínimos: al crear políticas personalizadas, conceda solo los permisos necesarios para llevar a cabo una tarea. Comience con un conjunto mínimo de permisos y conceda permisos adicionales según sea necesario. Por lo general, es más seguro que comenzar con permisos que son demasiado tolerantes e intentar hacerlos más severos más adelante. Para obtener más información, consulte Conceder privilegios mínimos en la Guía del usuario de IAM.
- Habilitar la MFA para operaciones confidenciales: para mayor seguridad, obligue a los usuarios de IAM a utilizar Multi-Factor Authentication (MFA) para acceder a recursos u operaciones de API confidenciales. Para obtener más información, consulteUso de la autenticación multifactor (MFA) enAWSen laGuía del usuario de IAM.
- Utilizar condiciones de política para mayor seguridad: en la medida en que sea práctico, defina las condiciones en las que las políticas basadas en identidades permitan el acceso a un recurso. Por ejemplo, puede escribir condiciones para especificar un rango de direcciones IP permitidas desde el que debe proceder una solicitud. También puede escribir condiciones para permitir solicitudes solo en un intervalo de hora o fecha especificado o para solicitar el uso de SSL o MFA. Para obtener más información, consulteElementos de política JSON de IAM: Condiciónen laGuía del usuario de IAM.

Políticas de usuario

Para acceder a la consola de App Runner, los usuarios de IAM deben tener un conjunto mínimo de permisos. Estos permisos deben permitirle mostrar y consultar los detalles sobre los recursos de App Runner en suCuenta de AWS. Si crea una política basada en identidad que sea más restrictiva que el mínimo de permisos necesarios, la consola de no funcionará del modo esperado para los usuarios con esa política de.

App Runner proporciona dos políticas administradas por que puede asociar a los usuarios de.

- `AWSAppRunnerReadOnlyAccess`: concede permisos para mostrar y ver detalles sobre los recursos de App Runner.

- `AWSAppRunnerFullAccess`— Otorga permisos a todas las acciones de App Runner.

Para asegurarse de que los usuarios pueden usar la consola de App Runner, adjunte, como mínimo, el`AWSAppRunnerReadOnlyAccess`Administrada a los usuarios de. Puede asociar la`AWSAppRunnerFullAccess`en su lugar, o agregar permisos adicionales específicos, para permitir a los usuarios crear, modificar y eliminar recursos. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM.

No es necesario que conceda permisos mínimos para la consola a los usuarios que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que desea permitir que los usuarios realicen.

En los siguientes ejemplos se muestran las políticas de usuario personalizadas. Puede utilizarlos como puntos de partida para definir sus propias directivas de usuario personalizadas. Copie el ejemplo y, o elimine acciones, reduzca el alcance de los recursos y agregue condiciones.

Ejemplo: directiva de usuario de administración de la consola y la conexión

Esta directiva de ejemplo permite el acceso a la consola y permite la creación y administración de conexiones. No permite la creación y administración de servicios de App Runner. Se puede adjuntar a un usuario cuya función es administrar el acceso del servicio App Runner a los activos de código fuente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apprunner:List*",
        "apprunner:Describe*",
        "apprunner:CreateConnection",
        "apprunner:DeleteConnection"
      ],
      "Resource": "*"
    }
  ]
}
```

Ejemplo: directivas de usuario que utilizan claves de condición

Los ejemplos de esta sección muestran permisos condicionales que dependen de algunas propiedades de recursos o parámetros de acción.

Esta política de ejemplo permite crear un servicio App Runner, pero deniega el uso de una conexión llamada`prod`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateAppRunnerServiceWithNonProdConnections",
      "Effect": "Allow",
      "Action": "apprunner:CreateService",
      "Resource": "*",
      "Condition": {
        "ArnNotLike": {
          "apprunner:ConnectionArn": "arn:aws:apprunner:*:*:connection/prod/*"
        }
      }
    }
  ]
}
```

Esta política de ejemplo permite actualizar un servicio de App Runner denominado`preprod`sólo con una configuración de escalado automático llamada`preprod`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUpdatePreProdAppRunnerServiceWithPreProdASConfig",
      "Effect": "Allow",
      "Action": "apprunner:UpdateService",
      "Resource": "arn:aws:apprunner:*:*:service/preprod/*",
      "Condition": {
        "ArnLike": {
          "apprunner:AutoScalingConfigurationArn":
 "arn:aws:apprunner:*:*:autoscalingconfiguration/preprod/*"
        }
      }
```

```
    }
  ]
}
```

Control del acceso a los servicios de App Runner basado en etiquetas

Puede utilizar las condiciones de su política basada en identidad para controlar el acceso a los recursos de App Runner basados en etiquetas. En este ejemplo se muestra cómo crear una política que permita eliminar un servicio de App Runner. Sin embargo, los permisos solo se conceden si la etiqueta de servicio `0wner` tiene el valor del nombre de usuario de dicho usuario. Esta política también proporciona los permisos necesarios para llevar a cabo esta acción en la consola.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListServicesInConsole",
      "Effect": "Allow",
      "Action": "apprunner:ListServices",
      "Resource": "*"
    },
    {
      "Sid": "DeleteServiceIfOwner",
      "Effect": "Allow",
      "Action": "apprunner:DeleteService",
      "Resource": "arn:aws:apprunner:*:*:service/*",
      "Condition": {
        "StringEquals": {"apprunner:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}
```

También puede asociar esta política al usuario de IAM en su cuenta. Si un usuario llamado`richard-roe`intenta eliminar un servicio de App Runner, el servicio debe estar etiquetado`0wner=richard-roe`o`owner=richard-roe`. De lo contrario, se le deniega el acceso. La clave de la etiqueta de condición `0wner` coincide con los nombres de las claves de condición `0wner` y `owner` porque no distinguen entre mayúsculas y minúsculas. Para obtener más información, consulte[Elementos de política JSON de IAM: Condición]en laGuía del usuario de IAM.

Uso de roles vinculados a servicios en App Runner

AWS App RunnerutilizaAWS Identity and Access Management(IAM)Roles vinculados a servicios. Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a App Runner. Las funciones vinculadas a servicios están predefinidas por App Runner e incluyen todos los permisos que el servicio requiere para llamar a otrosAWSServicios de en su nombre.

Un rol vinculado a un servicio simplifica la configuración de App Runner porque ya no tendrá que agregar manualmente los permisos necesarios. App Runner define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo App Runner puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se puede asociar a ninguna otra entidad de IAM

Solo puede eliminar una función vinculada a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de App Runner, ya que se evita la eliminación accidental de permisos para obtener acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que tienen Sí en la columna Rol vinculado a servicio. Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Permisos de roles vinculados a servicios para App Runner

App Runner usa el rol vinculado al servicio denominadoAWSServiceRoleForApprunner.

La función permite que App Runner realice las siguientes tareas:

- Insertar registros en grupos de registros de Amazon CloudWatch Logs.
- Cree reglas de Amazon CloudWatch Events para suscribirse a mensajes de imágenes de Amazon Elastic Container Registry (Amazon ECR).

La función vinculada al servicio AWSServiceRoleForoleForAprunner confía en que los siguientes servicios asuman la función:

- `apprunner.amazonaws.com`

La directiva de permisos del rol vinculado al servicio AWSServiceRoleForAprunner contiene todos los permisos que App Runner necesita para completar las acciones en su nombre.

AppRunnerServiceRolePolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:PutRetentionPolicy"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:*:*:log-group:/aws/apprunner/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/apprunner/*:log-stream:*"
      ]
    },
    {
      "Sid": "AllowPutRuleForManagedRules",
      "Effect": "Allow",
      "Action": [
        "events: PutRule",
        "events: PutTargets",
        "events: DeleteRule",
        "events: RemoveTargets",
        "events: DisableRule",
        "events: EnableRule"
      ],
      "Resource": "arn:aws:events:*:*:rule/apprunner-*",
      "Condition": {
        "StringEquals": {
          "events:ManagedBy": "apprunner.amazonaws.com",
          "events:source": "aws.ecr",
          "events:detail-type": "ECR Image Action"
```

```
        }
      }
    }
  ]
}
```

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o función) crear, editar o eliminar la descripción de una función vinculada a un servicio. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM..

Creación de un rol vinculado a un servicio en App Runner

No necesita crear manualmente un rol vinculado a un servicio. Cuando crea un servicio App Runner en elAWS Management Console, elAWS CLI, o elAWSAPI, App Runner crea automáticamente el rol vinculado al servicio.

Si elimina este rol vinculado a servicio y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando se crea un servicio de App Runner, App Runner vuelve a crear el rol vinculado al servicio automáticamente.

Modificación de un rol vinculado a un servicio en App Runner

App Runner no le permite editar el rol vinculado al servicio AWSServiceRoleForAprunner. Después de crear un rol vinculado a un servicio, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia al mismo. Sin embargo, puede editar la descripción del rol utilizando IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM..

Eliminación de un rol vinculado a un servicio de App Runner

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar los recursos del rol vinculado al servicio antes de eliminarlo manualmente.

En App Runner, esto significa eliminar todos los servicios de App Runner en su cuenta de. Para obtener más información sobre cómo eliminar los servicios de App Runner, consultethe section called “Eliminación”.

> **Note**
>
> Si el servicio App Runner está utilizando el rol cuando intenta eliminar los recursos, es posible que no se pueda borrar. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar manualmente el rol vinculado a un servicio mediante IAM

Utilice la consola de IAM, laAWS CLI, o elAWSAPI para eliminar el rol vinculado al servicio de AWSServiceRoleForAprunner. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a un servicio de App Runner

App Runner admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio está disponible. Para obtener más información, consulteAWS App RunnerCuotas y puntos de enlace deen laAWSReferencia general de.

Políticas administradas por AWS para AWS App Runner

Para agregar permisos a usuarios, grupos y roles de, es más fácil utilizarAWSque escribir directivas usted mismo. Lleva tiempo y experienciaCrear políticas administradas por el cliente de IAMque proporcionan a su equipo únicamente los permisos necesarios. Para empezar a trabajar rápidamente, puede utilizar nuestraAWSPolíticas administradas por. Estas directivas cubren casos de uso comunes y están disponibles en suAWSaccount. Para obtener más información acerca deAWSPolíticas administradas de, consulteAWSPolíticas administradas poren laGuía del usuario de IAM.

AWSmantenimiento y actualización de serviciosAWSPolíticas administradas por. No se pueden cambiar los permisos de enAWSPolíticas administradas por. Los servicios ocasionalmente agregan permisos adicionales a unAWSpara admitir nuevas características. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) a los que está asociada la política. Es más probable que los servicios actualicen unAWScuando se inicia una nueva característica o cuando hay nuevas operaciones disponibles. Los servicios no eliminan permisos de unAWS, por lo que las actualizaciones de directivas no romperán los permisos existentes.

Además,AWSes compatible con las políticas administradas por funciones de trabajo que abarcan varios servicios. Por ejemplo, elReadOnlyAccess AWSLa política administrada proporciona acceso de solo lectura a todos losAWSServicios y recursos de. Cuando un servicio inicia una nueva característica,AWSagrega permisos de sólo lectura para nuevas operaciones y recursos. Para obtener una lista y descripciones de políticas de funciones, consulteAWSManaged Policies de para funciones de trabajoen laGuía del usuario de IAM.

App Runner se actualiza aAWSPolíticas administradas por

Ver detalles sobre las actualizaciones deAWSpara App Runner desde que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de App Runner.

Cambio	Descripción	Fecha
ApprunnerServiceReolePolicy Nueva política de política	App Runner agregó una nueva política para permitir que App Runner realice llamadas a los Amazon CloudWatch Logs y a los Amazon CloudWatch Events en nombre de los servicios de App Runner.	1 de marzo de 2021
awsAppRunnerReadOnlyAccessNueva política de política	App Runner agregó una nueva política para permitir a los usuarios enumerar y ver detalles sobre los recursos de App Runner.	1 de marzo de 2021
AWSAPPrunnerFullAccessNueva política de política	App Runner agregó una nueva política para permitir a los usuarios realizar todas las acciones de App Runner.	1 de marzo de 2021
awsAppRunnerServicePolicyForecrAccessNueva política de política	App Runner agregó una política nueva para permitir que App Runner pueda acceder	1 de marzo de 2021

Cambio	Descripción	Fecha
	a imágenes de Amazon Elastic Container Registry (Amazon ECR) en su cuenta.	
App Runner comenzó el seguimien to de los cambios	App Runner comenzó el seguimiento de los cambios para suAWSPolíticas administradas por.	1 de marzo de 2021

Solución de problemas de identidad y acceso de App Runner

Utilice la información siguiente para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje conAWS App RunnerIAM.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Temas

- No tengo autorización para realizar una acción en App Runner
- Soy administrador y deseo permitir que otros obtengan acceso a App Runner
- Quiero permitir a personas externas a miCuenta de AWSpara acceder a mis recursos de App Runner

No tengo autorización para realizar una acción en App Runner

Si la AWS Management Console le indica que no está autorizado para llevar a cabo una acción, póngase en contacto con su administrador para recibir ayuda. Su administrador es la persona que le facilitó suAWSNombre de usuario y contraseña de.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado`marymajor`intenta utilizar la consola para ver detalles sobre un servicio de App Runner pero no tiene`apprunner:DescribeService`Permisos.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 apprunner:DescribeService on resource: my-example-service
```

En este caso, Mary pide a su administrador que actualice sus políticas para que pueda acceder a*my-example-service*Con el recurso`apprunner:DescribeService`action.

Soy administrador y deseo permitir que otros obtengan acceso a App Runner

Para permitir que otros obtengan acceso a App Runner, debe crear una entidad de IAM (usuario o rol) para la persona o aplicación que necesita acceso. Esta persona utilizará las credenciales de la entidad para obtener acceso a AWS. A continuación, debe asociar una política a la entidad que le conceda los permisos correctos en App Runner.

Para comenzar de inmediato, consulte Creación del primer grupo y usuario delegado de IAM en la Guía del usuario de IAM.

Quiero permitir a personas externas a miCuenta de AWSpara acceder a mis recursos de App Runner

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 obtener más información, consulte lo siguiente:

- Para saber si App Runner admite estas características, consulteCómo funciona App Runner con IAM.
- Para obtener información sobre cómo proporcionar acceso a sus recursos a través deAWSCuentas de su propiedad, consulteProporcionar acceso a un usuario de IAM en otroAWSCuenta de la suyaen laGuía del usuario de IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos deAWSCuentas, consulteProporcionar acceso aAWSCuentas de propiedad de tercerosen laGuía del usuario de IAM.
- Para obtener información sobe cómo proporcionar acceso mediante una identidad federada, 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.

Registro y monitoreo en App Runner

La monitorización es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el desempeño de suAWS App RunnerServicio Recopilación de datos de monitoreo de todas las partes de suAWSle permite depurar más fácilmente una falla si ocurre una. App Runner se integra con variosAWSpara monitorear sus servicios de App Runner y responder a posibles incidentes.

Alarmas de Amazon CloudWatch

Con las alarmas de Amazon CloudWatch, puede ver una métrica de servicio durante el período de tiempo que especifique. Si la métrica supera un umbral determinado durante un número determinado de períodos, recibirá una notificación.

App Runner recopila una variedad de métricas sobre el servicio en su conjunto y las instancias (unidades de escala) que ejecutan el servicio web. Para obtener más información, consulte Métricas (CloudWatch).

Logs de aplicaciones

App Runner recopila la salida del código de la aplicación y lo transmite a Amazon CloudWatch Logs. Lo que hay en este resultado depende de usted. Por ejemplo, puede incluir registros detallados de las solicitudes realizadas a su servicio web. Estos registros de pueden resultar útiles en auditorías de acceso y seguridad. Para obtener más información, consulte Registros (registros de CloudWatch Logs).

AWS CloudTrailLog de acciones

App Runner está integrado conAWS CloudTrail, un servicio que proporciona un registro de las medidas adoptadas por un usuario, un rol o unAWSen App Runner. CloudTrail captura como eventos todas las llamadas a la API de App Runner. Puede ver los eventos más recientes en la consola de CloudTrail y crear una pista para permitir la entrega continua de eventos de CloudTrail a un depósito de Amazon Simple Storage Service (Amazon S3). Para obtener más información, consulte Acciones de API (CloudTrail).

Validación de conformidad para App Runner

Auditores externos evalúan la seguridad y la conformidad de AWS App Runner en el marco de varios programas de conformidad de AWS. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para saber si App Runner u otrosAWSse incluyen en un programa de conformidad específico, consulteAWSServicios en el ámbito del programa de conformidad. Para obtener información general, consulte Programas de conformidad de AWS.

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Su responsabilidad de cumplimiento al usar servicios de AWS está determinada por la sensibilidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y regulaciones aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido de seguridad y conformidad: estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia enAWSque se centran en la seguridad y el cumplimiento normativo.
- Documento técnico de seguridad y conformidad de HIPAA— Este documento técnico describe cómo las empresas pueden utilizarAWSpara crear aplicaciones que se ajusten al estándar HIPAA.

  > **Note**
  >
  > No todos los servicios cumplen con la HIPAA.

- AWSRecursos de conformidad de: es posible que este conjunto de guías y libros de ejercicios se apliquen a su sector y ubicación.
- Evaluación de recursos con reglasen laAWS ConfigGuía para desarrolladores— ElAWS Configeste servicio de evalúa en qué medida las configuraciones de los recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- AWS Security Hub— EstoAWSproporciona una visión completa de su estado de seguridad dentro deAWSDe este modo, podrá contraste el grado de conformidad con las prácticas recomendadas y los estándares sectoriales.
- AWS Audit Manager— EstoAWSle ayuda a auditar continuamente suAWSpara simplificar la forma en que gestiona los riesgos y el cumplimiento de las regulaciones y los estándares del sector.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Resiliencia en App Runner

LaAWSLa infraestructura global de está constituida porRegiones de AWSZonas de disponibilidad yRegiones de AWSLas redes proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información acerca deRegiones de AWSZonas de disponibilidad yAWSInfraestructura global.

AWS App Runner administra y automatiza el uso de la infraestructura global de AWS en su nombre. Al usar App Runner, se beneficia de los mecanismos de disponibilidad y tolerancia a fallos que se encuentranAWSOfertas.

Para ver otros temas de seguridad de App Runner, consulteSeguridad.

Seguridad de la infraestructura de AWS App Runner

Como servicio administrado,AWS App Runnerestá protegido por elAWSProcedimientos de seguridad de red globales de que se describen en elAmazon Web Services: Información general sobre procesos de seguridad deDocumento técnico.

Usted usaAWSPara obtener acceso a App Runner a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.0 o una versión posterior. Le recomendamos TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). 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.

Para ver otros temas de seguridad de App Runner, consulteSeguridad.

Configuración y análisis de vulnerabilidades en App Runner

AWS y nuestros clientes comparten la responsabilidad de conseguir un alto nivel de seguridad y conformidad en los componentes del software. Para obtener más información, consulte laAWS Modelo de responsabilidad compartida.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Prácticas recomendadas de seguridad para App Runner

AWS App Runner cuenta con varias 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 el entorno, considérelas como consideraciones útiles, no como normas.

Para consultar otros temas de seguridad de App Runner, consulteSeguridad.

Prácticas recomendadas de seguridad preventiva

Los controles de seguridad preventivos intentan evitar incidentes antes de que ocurran.

Implementación del acceso a los privilegios mínimos

App Runner ofreceAWS Identity and Access ManagementPolíticas administradas de IAM paraUsuarios de IAMy laRol de acceso. Estas políticas administradas especifican todos los permisos que pueden ser necesarios para el correcto funcionamiento de su servicio de App Runner.

Es posible que su aplicación no requiera todos los permisos de nuestras políticas administradas. Puede personalizarlos y conceder solo los permisos necesarios para sus usuarios y su servicio de App Runner para realizar sus tareas. Esto es especialmente importante para las políticas de usuario, donde diferentes roles de usuario pueden tener necesidades de permiso distintas. La implementación del acceso con privilegios mínimos es esencial a la hora de reducir los riesgos de seguridad y el impacto que podrían causar los errores o los intentos malintencionados.

Prácticas recomendadas de detección de seguridad

Los controles de detección de seguridad identifican las infracciones de seguridad tras producirse. Pueden ayudarte a detectar una posible amenaza o incidente de seguridad.

Implementar la monitorización

La supervisión es un aspecto importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de las instancias de las instancias de sus soluciones de App Runner.AWSOfrece varias herramientas y servicios para ayudarle a supervisar suAWSServicios de .

A continuación se muestran algunos ejemplos de elementos que monitorizar:

- Métricas de Amazon CloudWatch para App Runner: establezca alarmas para métricas clave de App Runner y para las métricas personalizadas de su aplicación. Para obtener más información, consulte Métricas (CloudWatch).
- AWS CloudTrailentries: realice un seguimiento de las acciones que pueden afectar a la disponibilidad, como`PauseService`or`DeleteConnection`. Para obtener más información, consulte Acciones de API (CloudTrail).

AWSGlosario

Contiene la más recienteAWSterminología, consulte laAWSGlosarioen laAWSReferencia general de.

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.