aws

Guía del usuario

AWS Batch

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

AWS Batch: Guía del usuario

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es AWS Batch?

AWS Batch le ayuda a ejecutar cargas de trabajo de computación por lotes en Nube de AWS. La informática por lotes es una forma común de acceso a grandes cantidades de recursos informáticos utilizada por desarrolladores, científicos e ingenieros. AWS Batch elimina la ardua tarea de configurar y administrar la infraestructura necesaria, de forma similar al software de informática por lotes tradicional. Este servicio puede aprovisionar recursos de forma eficaz en respuesta a los trabajos enviados para eliminar limitaciones de capacidad, reducir costos informáticos y ofrecer resultados con rapidez.

Al ser un servicio completamente administrado, AWS Batch le ayuda a ejecutar cargas de trabajo informático por lotes de cualquier escala. AWS Batch proporciona automáticamente recursos informáticos y optimiza la distribución de la carga de trabajo en función de la cantidad y la escala. Con AWS Batch, no es necesario instalar ni administrar software de informática por lotes, lo que le permite centrarse en analizar resultados y resolver problemas.

Temas

Componentes de AWS Batch

AWS Batch simplifica los trabajos por lotes en ejecución en múltiples zonas de disponibilidad dentro de una región. Puede crear entornos informáticos de AWS Batch dentro de una VPC nueva o existente. Después de que un entorno informático se ha activado y asociado a una cola de trabajos, puede precisar las definiciones de trabajo que especifican cuáles imágenes de contenedor Docker ejecutarán sus trabajos. Las imágenes de contenedor se almacenan y se extraen desde registros de contenedor, que podrían existir dentro o fuera de la infraestructura de AWS.

Trabajos

Una unidad de trabajo (como un script de shell, un ejecutable en Linux o una imagen de contenedor Docker) que envía a AWS Batch. Tiene un nombre y se ejecuta como una aplicación en contenedor en AWS Fargate o recursos de Amazon EC2 de su entorno informático, utilizando los parámetros que se especifican en una definición de trabajo. Los trabajos pueden hacer referencia a otros

trabajos por nombre o ID, y puede que de ellos dependa la correcta realización de otros trabajos. Para obtener más información, consulte Jobs.

Definiciones de trabajo

Una definición de trabajo especifica cómo se ejecutan los trabajos. Puede considerar una definición de trabajo como un esquema de los recursos en su trabajo. Puede asignar a su trabajo un rol de IAM para proporcionar acceso a otros recursos de AWS. También puede especificar los requisitos de memoria y CPU. La definición de trabajo también puede controlar las propiedades de contenedor, las variables de entorno y los puntos de montaje para un almacenamiento persistente. Muchas de las especificaciones de una definición de trabajo pueden anularse mediante la especificación de nuevos valores al enviar trabajos individuales. Para obtener más información, consultar Definiciones de trabajo

Colas de trabajo

Al enviar un trabajo de AWS Batch, lo envía a una cola de trabajos determinada, donde el trabajo permanecerá hasta que se programe en un entorno informático. Asocie uno o más entornos informáticos con una cola de trabajos. También puede asignar valores de prioridad a estos entornos informáticos e incluso a las propias colas de trabajos. Por ejemplo, puede tener una cola de prioridad alta a la que envía trabajos prioritarios, y una cola de prioridad baja para los trabajos que pueden ejecutarse en cualquier momento, cuando los recursos informáticos son más económicos.

Entorno informático

Un entorno informático es un conjunto de recursos informáticos administrados o no administrados que se utilizan para ejecutar trabajos. Con los entornos de procesamiento gestionados, usted puede especificar el tipo de procesamiento deseado (Fargate o EC2) con varios niveles de detalle. Puede configurar entornos informáticos que utilicen un tipo de instancia EC2 determinado, como `c5.2xlarge` o `m5.10xlarge`. O bien, usted puede elegir especificar únicamente que desea utilizar los tipos de instancias más recientes. También puede especificar el número mínimo, el número deseado y el número máximo de CPU virtuales del entorno, junto con la cantidad que está dispuesto a pagar por una instancia de spot como un porcentaje del precio de las instancias bajo demanda y un conjunto de destino de subredes de la VPC. AWS Batch lanzará, administrará y terminará en forma eficiente los tipos de computación según sea necesario. También puede administrar sus propios entornos informáticos. En este caso, usted es responsable de configurar y escalar las instancias en un clúster de Amazon ECS que AWS Batch crea para usted. Para obtener más información, consulte Entorno de computación.

Introducción

Para comenzar a usar AWS Batch, cree una definición de trabajo, un entorno informático y una cola de trabajo en la consola de AWS Batch.

El asistente de primera ejecución de AWS Batch ofrece la posibilidad de crear un entorno informático y una cola de trabajo, y enviar un trabajo de ejemplo «Hello World». Si, en su lugar, ya tiene una imagen de Docker que desea lanzar en AWS Batch, puede crear una definición de trabajo con esa imagen y enviarla a la cola. Para obtener más información, consulte Cómo empezar con AWS Batch.

Panel

En el panel de AWS Batch, puede supervisar los trabajos recientes, las colas de trabajos y los entornos informáticos. De forma predeterminada, se muestran los siguientes widgets del panel:

- Descripción general de trabajos: para obtener más información sobre trabajos de AWS Batch, consulte Jobs.
- Descripción general de las colas de trabajos: para obtener más información sobre las colas de trabajos de AWS Batch, consulte Colas de trabajo.
- Descripción general del entorno informático: para obtener más información sobre los entornos informáticos de AWS Batch, consulte Entorno de computación.

Puede personalizar los widgets que se muestran en la página del panel de control. En las siguientes secciones se describen los widgets adicionales que puede instalar.

Cola de trabajos única

Este widget muestra información detallada sobre una única cola de trabajos.

Para añadir este widget, siga estos pasos.

1. Abra la consola de AWS Batch.
2. En la barra de navegación, seleccione la Región de AWS que desea utilizar.
3. En el panel de navegación, elija Panel.
4. Elija Añadir widget.
5. En Cola de trabajos única, elija Añadir widget.

6. En Cola de trabajos, elija la cola de trabajos que desee.
7. En Estado del trabajo, elija los estados del trabajo que desee mostrar.
8. (Opcional) Desactive Mostrar entornos informáticos conectados si no desea mostrar las propiedades de los entornos informáticos.
9. En Propiedades del entorno informático, elija las propiedades que desee.
10. Elija Añadir.

Información de contenedores de CloudWatch

Este widget muestra métricas agregadas para trabajos y entornos informáticos de AWS Batch. Para obtener más información sobre los contenedores, consulte Información de contenedores de CloudWatch.

Para añadir este widget, siga estos pasos.

1. Abra la consola de AWS Batch.
2. En la barra de navegación, seleccione el Región de AWS que desea utilizar.
3. En el panel de navegación, elija Panel.
4. Elija Añadir widget.
5. En Información sobre los contenedores, elija Añadir widget.
6. En Entorno informático, elija el entorno informático que desee.
7. Elija Añadir.

Registros de trabajo

Este widget muestra distintos registros desde los trabajos en una ubicación práctica. Para obtener más información acerca de los registros de trabajo, consulte the section called “ Registros de trabajo”.

Para añadir este widget, siga estos pasos.

1. Abra la consola de AWS Batch.
2. En la barra de navegación, seleccione el Región de AWS que desea utilizar.
3. En el panel de navegación, elija Panel.
4. Elija Añadir widget.

5. Para Registros de trabajos, elija Añadir widget.
6. En ID de trabajo, introduzca el ID de trabajo del trabajo que desee.
7. Elija Añadir.

Configuración con AWS Batch

Si ya se ha registrado en Amazon Web Services (AWS) y está utilizando Amazon Elastic Compute Cloud (Amazon EC2) o Amazon Elastic Container Service (Amazon ECS), pronto podrá usar AWS Batch. El proceso de configuración para estos servicios es similar. Esto se debe a que AWS Batch utiliza instancias de contenedor de Amazon ECS en sus entornos informáticos. Para usar la AWS CLI con AWS Batch debe usar una versión de la AWS CLI que admita las últimas características de AWS Batch. Si alguna de las características de AWS Batch no son compatibles con la AWS CLI, debe actualizar a la última versión. Para obtener más información, consulte http://aws.amazon.com/cli/.

> **Note**
>
> Como AWS Batch utiliza componentes de Amazon EC2, se utiliza la consola de Amazon EC2 para muchos de estos pasos.

Lleve a cabo las siguientes tareas para obtener la configuración de AWS Batch. Si ya ha realizado alguno de estos pasos, puede omitirlos y proceder con la instalación de AWS CLI.

Temas

Registro para obtener una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

Cómo registrarse en una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.

2. Siga las instrucciones que se le indiquen.

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

   Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario administrativo y utilice únicamente el usuario raíz para realizar tareas que requieran acceso de usuario raíz.

AWS le enviará un correo electrónico de confirmación luego de completar el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Crear un usuario administrativo

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario raíz en las tareas cotidianas.

Protección de su Usuario raíz de la cuenta de AWS

1. Inicie sesión en la AWS Management Console como propietario de cuenta, elija Usuario raíz e ingrese el email de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In.

2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario raíz de la Cuenta de AWS (consola) en la Guía del usuario de IAM.

Creación de un usuario administrativo

1. Activar IAM Identity Center

   Para conocer las instrucciones, consulte Habilitar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center.

2. En IAM Identity Center, otorga acceso administrativo a un usuario administrativo.

   Para ver un tutorial sobre el uso de Directorio de IAM Identity Center como origen de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada de Directorio de IAM Identity Center en la Guía del usuario de AWS IAM Identity Center.

Cómo iniciar sesión como usuario administrativo

- Para iniciar sesión con el usuario del IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario del IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del IAM Identity Center, consulte Iniciar sesión en el portal de acceso de AWS en la Guía del Usuario de AWS Sign-In.

Creación de roles de IAM para los entornos informáticos e instancias de contenedor

Las instancias de contenedor y los entornos informáticos de AWS Batch requieren credenciales de Cuenta de AWS para realizar llamadas a otras API de AWS en su nombre. Debe crear un rol de IAM que proporcione estas credenciales a los entornos informáticos e instancias de contenedor y, a continuación, asociar el rol a los entornos informáticos.

> **Note**
>
> Los roles de instancia de contenedor y de entorno informático de AWS Batch se crean automáticamente en la experiencia de primera ejecución de la consola. Por lo tanto, si tiene intención de utilizar la consola AWS Batch, puede pasar a la siguiente sección. Si, en cambio, tiene previsto utilizar la AWS CLI, siga el procedimiento descrito en Uso de funciones vinculadas a servicios para AWS Batch y en Función de instancia de Amazon ECS antes de crear el primer entorno informático.

Crear un par de claves

AWS utiliza criptografía de clave pública para proteger la información de inicio de sesión de la instancia. Una instancia de Linux como, por ejemplo, una instancia de contenedor de entorno informático de AWS Batch, no tiene contraseña para acceder a SSH. Utilice un par de claves para

iniciar sesión de forma segura en la instancia. Primero se especifica el nombre del par de claves al crearse el entorno informático y, a continuación, se proporciona la clave privada al iniciar sesión con SSH.

Si aún no ha creado un par de claves, puede crear uno con la consola de Amazon EC2. Tenga en cuenta que si tiene previsto lanzar instancias en varias Regiones de AWS, debe crear un par de claves en cada una de ellas. Para obtener más información acerca de las regiones, consulte Zonas de disponibilidad y regiones en la Guía del usuario de Amazon EC2 para instancias de Linux.

Crear un par de claves

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En la barra de navegación, seleccione una región para el par de claves Región de AWS. Se puede seleccionar cualquier región disponible, independientemente de su ubicación. Sin embargo, cada par de claves corresponde a una región específica. Por ejemplo, si tiene previsto lanzar una instancia en la región Oeste de EE. UU. (Oregón), debe crear un par de claves para la instancia en la misma región.
3. En el panel de navegación, seleccione Key Pairs (Pares de claves), Create Key Pair (Crear par de claves).
4. En el cuadro de diálogo Create Key Pair (Crear par de claves), en Key pair name (Nombre del par de claves), escriba un nombre para el nuevo par de claves y, a continuación, seleccione Create (Crear). Elija un nombre que pueda recordar, como su nombre de usuario, seguido de `-key-pair` y del nombre de la región. Por ejemplo, yopar-de-clavesuswest2.
5. Su navegador descargará el archivo de clave privada automáticamente. El nombre de archivo base es el nombre que especificó como nombre del par de claves y la extensión del archivo es `.pem`. Guarde el archivo de clave privada en un lugar seguro.

   > **Important**
   >
   > Esta es la única oportunidad para guardar el archivo de clave privada. Necesita proporcionar el nombre del par de claves al lanzar una instancia y la clave privada correspondiente cada vez que se conecte a dicha instancia.

6. Si usa un cliente SSH en un equipo Mac o Linux para conectarse a su instancia de Linux, utilice el comando a continuación para establecer los permisos de su archivo de clave privada. De esa forma, solo usted podrá leerlo.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

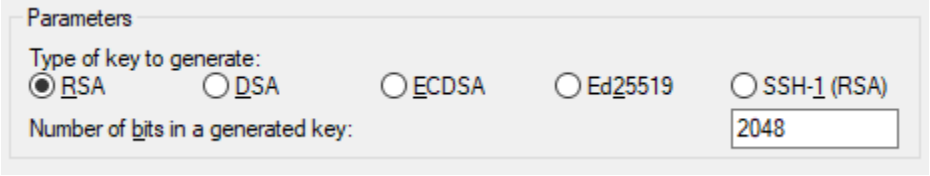
Para obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2 para instancias de Linux.

Para conectarse a la instancia mediante el par de claves

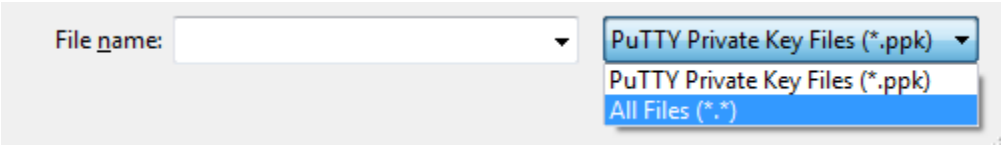
Para conectarse a la instancia de Linux desde un equipo que ejecute Mac OS o Linux, especifique el archivo `.pem` a su cliente SSH con la opción `-i` y la ruta a su clave privada. Para conectarte a tu instancia de Linux desde un ordenador que ejecute Windows, usa PuTTY MindTerm o PuTTY. Si tiene previsto utilizar PuTTY, instálelo y utilice el siguiente procedimiento para convertir el archivo `.pem` en un archivo `.ppk`.

(Opcional) Para prepararse para conectarse a una instancia de Linux desde Windows mediante PuTTY

1. Descargue PuTTY de http://www.chiark.greenend.org.uk/~sgtatham/putty/ e instálelo. Asegúrese de instalar el conjunto completo.
2. Inicie PuTTYgen (por ejemplo, desde el menú Inicio, seleccione Todos los programas, PuTTY, PuTTYgen).
3. En Type of key to generate (Tipo de clave a generar), elija RSA. Si está usando una versión antigua de PuTTYgen, elija SSH-2 RSA.



4. Elija Load (Cargar). De forma predeterminada, PuTTYgen muestra solo archivos con la extensión `.ppk`. Para localizar el archivo `.pem`, seleccione la opción de mostrar todos los tipos de archivo.



5. Seleccione el archivo de clave privada que creó en el procedimiento anterior y elija Open (Abrir). Elija OK (Aceptar) para descartar el cuadro de diálogo de confirmación.

6. Elija Save private key (Guardar clave privada). PuTTYgen mostrará una advertencia acerca de guardar la clave sin una frase de contraseña. Elija Yes (Sí).
7. Especifique para la clave el mismo nombre que utilizó para el par de claves. PuTTY añade la extensión de archivo `.ppk` automáticamente.

Creación de una VPC

Con Amazon Virtual Private Cloud (Amazon VPC), puede lanzar recursos de AWS en una red virtual que haya definido. Recomendamos lanzar las instancias de contenedor en una VPC.

Si dispone de una VPC predeterminada, también puede omitir esta sección y pasar a la siguiente tarea Creación de un grupo de seguridad. Para determinar si dispone de una VPC predeterminada, consulte Plataformas compatibles con la consola de Amazon EC2 en la Guía del usuario de Amazon EC2 para instancias de Linux

Para obtener información acerca de cómo crear una VPC de Amazon, consulte Creación de una VPC en la Guía del usuario de Amazon VPC. Consulte la siguiente tabla para determinar qué opciones seleccionar.

Opción	Valor
Recursos para crear	VPC solo
Nombre	De manera opcional, indique un nombre para su VPC.
IPv4 CIDR block	Entrada manual de IPv4 CIDR El bloque de CIDR debe ser de un tamaño de entre /16 y /28.
IPv6 CIDR block	No hay bloque de CIDR IPv6
Propiedad	Predeterminado

Para obtener más información acerca de Amazon VPC, consulte ¿Qué es Amazon VPC? en la Guía del usuario de Amazon VPC.

Creación de un grupo de seguridad

Los grupos de seguridad actúan como un cortafuegos para las instancias de contenedor de entorno informático asociadas, al controlar el tráfico entrante y saliente en el nivel de instancia del contenedor. El grupo de seguridad solo se puede utilizar en la VPC para la que se creó.

Es posible añadir reglas a un grupo de seguridad que le permita conectarse a la instancia de contenedor desde su dirección IP mediante SSH. También se pueden añadir reglas que permitan HTTP de entrada y salida y acceso HTTPS desde cualquier lugar. Añada reglas para abrir los puertos requeridos por las tareas.

Recuerde que si pretende lanzar instancias de contenedor en varias regiones deberá crear un grupo de seguridad por región. Para obtener más información, consulte Regiones y zonas de disponibilidad en la Guía del usuario de Amazon EC2 para instancias de Linux.

> **Note**
>
> Necesitará la dirección IP pública de su equipo local, que puede obtener mediante un servicio. Por ejemplo, proporcionamos el siguiente servicio: http://checkip.amazonaws.com/ o https://checkip.amazonaws.com/. Para buscar otro servicio que le brinde su dirección IP, utilice la frase de búsqueda "what is my IP address" (cuál es mi dirección IP). Si se conecta a través de un proveedor de servicios de internet (ISP) o protegido por un firewall sin una dirección IP estática, deberá identificar el rango de direcciones IP utilizadas por los equipos cliente.

Para crear un grupo de seguridad con la consola

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Security Groups (Grupos de seguridad).
3. Elija Create Security Group (Crear grupo de seguridad).
4. Ingrese un nombre y una descripción para el grupo de seguridad. No puede cambiar el nombre ni la descripción de un grupo de seguridad después de crearlo.
5. En VPC, elija la VPC.
6. (Opciona) De forma predeterminada, los grupos de seguridad nuevos comienzan con una única regla de salida que permite que todo el tráfico salga del recurso. Debe añadir reglas para permitir el tráfico entrante o restringir el tráfico saliente.

Las instancias de contenedor de AWS Batch no requieren la apertura de ningún puerto de entrada. Sin embargo, es posible que desee agregar una regla SSH. De ese modo, puede iniciar sesión en la instancia de contenedor y examinar los contenedores en los trabajos con comandos de Docker. Si desea que su instancia de contenedor aloje un trabajo que ejecute un servidor web, también puede agregar reglas para HTTP. Siga los pasos a continuación para añadir estas reglas de grupo de seguridad opcionales.

En la pestaña Entrada cree las siguientes reglas y seleccione Crear:

- Seleccione Add Rule (Agregar regla). En Tipo, seleccione HTTP. En Fuente, elija Cualquier lugar (`0.0.0.0/0`).
- Seleccione Add Rule (Agregar regla). En Tipo, seleccione SSH. En Fuente, elija IP personalizada y especifique la dirección IP pública de su computadora o red en notación de enrutamiento entre dominios sin clases (CIDR). Si su empresa asigna direcciones de un rango, especifíquelo; por ejemplo, `203.0.113.0/24`. Para especificar una dirección IP individual en notación CIDR, elija Mi IP. Esto añade el prefijo de enrutamiento `/32` a la dirección IP pública.

> **Note**
>
> Por motivos de seguridad, recomendamos que no permita acceso SSH desde todas las direcciones IP (`0.0.0.0/0`) a su instancia, excepto con fines de prueba y solamente durante un breve periodo.

7. Puede agregar etiquetas ahora o más adelante. Para agregar una etiqueta, elija Add new tag (Agregar nueva etiqueta) y, a continuación, ingrese la clave y el valor de la etiqueta.
8. Elija Create Security Group (Crear grupo de seguridad).

Para crear un grupo de seguridad mediante la línea de comandos, consulta create-security-group() AWS CLI

Para obtener más información acerca de los grupos de seguridad, consulte Trabajar con grupos de seguridad.

Instalar la AWS CLI

Para utilizar la AWS CLI con AWS Batch, instale la última versión de la AWS CLI. Para obtener más información acerca de cómo instalar la AWS CLI o cómo actualizarla a la versión más reciente, consulte Instalación de la interfaz de línea de comandos de AWS en la Guía del usuario de AWS Command Line Interface.

Cómo empezar con AWS Batch

Puede utilizar el asistente AWS Batch de primera ejecución para empezar AWS Batch rápidamente. Tras completar los requisitos previos, puede utilizar el asistente de primera ejecución para crear un entorno de computación, una definición de trabajos y una cola de trabajos.

También puede enviar un ejemplo de trabajo de «Hello World» mediante el asistente de AWS Batch primera ejecución para probar la configuración. Si ya tiene una imagen de Docker con la que quiere lanzarla AWS Batch, puede utilizarla para crear una definición de trabajo.

Requisitos previos

Asegúrese de hacer lo siguiente antes de iniciar el asistente AWS Batch por primera vez:

- Complete los pasos descritos en Configuración con AWS Batch.
- Compruebe que Cuenta de AWS dispone de los permisos necesarios.

Introducción: Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable y segura en Nube de AWS. El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo.

Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento. Amazon EC2 le permite escalar hacia arriba o hacia abajo para controlar los cambios en los requisitos o los picos de popularidad, con lo que se reduce la necesidad de prever el tráfico.

Crear un entorno de computación

Para crear un entorno de computación para una orquestación de Amazon EC2, haga lo siguiente:

1. Abra el Asistente de primer uso de la consola AWS Batch.
2. Para el Selección del tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
3. Elija Siguiente.

4. En la sección de Configuración de entorno de computación de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
5. En Rol de instancia, elija un perfil de instancia existente que tenga asociados los permisos de IAM necesarios. Este perfil de instancia permite a las instancias de contenedor de Amazon ECS de su entorno informático realizar llamadas a las operaciones de AWS API requeridas. Para obtener más información, consulte Función de instancia de Amazon ECS.
6. (Opcional) Una etiqueta es una marca que se asigna a un recurso. Para añadir una etiqueta o una etiqueta Amazon EC2, expanda Etiquetas y, a continuación, seleccione Agregar etiqueta. Introduzca un par clave-valor y, a continuación, vuelva a seleccionar Agregar etiqueta.

   > **Important**
   >
   > Si elige Agregar etiqueta, debe introducir un par clave-valor y volver a elegir Agregar etiqueta o bien elegir Eliminar etiqueta.

7. (Opcional) En la sección Configuración de instancias para Usar instancias de spot de Amazon EC2, active Habilitar el uso de instancias spot.
8. (Solo spot) Para obtener el porcentaje máximo de precio bajo demanda, introduzca el porcentaje máximo del precio bajo demanda que desea pagar por los recursos de spot.
9. (Opcional) (solo Spot) En Rol de la flota de spot, seleccione un rol de IAM para la flota de spot de Amazon EC2 que quiera aplicar a su entorno de computación de spot. Si aún no tiene un rol de IAM para la flota de spot de Amazon EC2, primero debe crear uno. Para obtener más información, consulte Rol de flota de spot de Amazon EC2.

   > **Important**
   >
   > Para etiquetar las instancias puntuales al crearlas, su función de IAM de Amazon EC2 Spot Fleet debe utilizar la política gestionada más reciente de SpotFleetTaggingRoleAmazonEC2. La política SpotFleetRole administrada de AmazonEC2 no tiene los permisos necesarios para etiquetar las instancias puntuales. Para obtener más información, consulte Instancias de spot no etiquetadas en el momento de su creación y the section called “Etiquetado de los recursos de ”.

10. En Mínimo de CPU virtuales, seleccione la cantidad mínima de vCPUs de EC2 que mantienen el entorno de computación, independientemente de la demanda de las colas de trabajos.

11. En CPU virtuales deseadas, seleccione la cantidad de vCPU de EC2 con las que el entorno de computación realiza lanzamientos. A medida que aumenta la demanda de colas de trabajos, AWS Batch aumenta la cantidad deseada de vCPU y agrega instancias EC2. La cantidad de vCPU puede aumentar hasta la cantidad máxima de vCPU. A medida que disminuya la demanda, AWS Batch disminuya la cantidad deseada de vCPU y elimine instancias. El número de vCPU se reduce completamente hasta el número mínimo de vCPU.
12. En Máximo de CPU virtuales, seleccione la cantidad máxima de vCPUs de EC2 que su entorno de computación puede escalar horizontalmente, independientemente de la demanda de las colas de trabajos.
13. En Tipos de instancias permitidos, elija los tipos de instancia de Amazon EC2 que se pueden lanzar. Se pueden especificar familias de instancias para lanzar cualquier tipo de instancia en esas familias (por ejemplo, `c5`, `c5n` o `p3`). O bien puede especificar tamaños específicos dentro de una familia (por ejemplo, `c5.8xlarge`). Los tipos de instancias metálicas no están en las familias de instancias. Por ejemplo, `c5` no incluye `c5.metal`. También puede seleccionar `optimal` para elegir tipos de instancias (de las familias de instancias C4, M4 y R4) que se correspondan con la demanda de las colas de trabajos.

> **Note**
>
> Cuando se crea un entorno de computación, los tipos de instancias que se seleccionen para dicho entorno de computación deben compartir la misma arquitectura. Por ejemplo, no se puede mezclar instancias x86 y ARM en el mismo entorno de computación.

> **Note**
>
> AWS Batch escala las GPU en función de la cantidad requerida en las colas de trabajos. Para utilizar la programación de GPU, el entorno de computación debe incluir tipos de instancia de las familias `p2`, `p3`, `p4`, `p5`, `g3`, `g3s`, `g4` o `g5`.

> **Note**
>
> Actualmente, `optimal` utiliza tipos de instancia de las familias de instancias C4, M4 y R4. Si Regiones de AWS no hay tipos de instancias de esas familias de instancias, se utilizan los tipos de instancia de C5M5, y las familias de R5 instancias.

14. Expanda Configuración adicional.
15. (Opcional) En Grupo de ubicación, introduzca un nombre de grupo de ubicación para agrupar los recursos en el entorno de computación.
16. (Opcional) En Par de claves EC2, elija un par de claves pública y privada como credenciales de seguridad cuando se conecte a la instancia. Para obtener más información sobre pares de claves de Amazon EC2, consulte pares de claves de Amazon EC2 e instancias de Linux.
17. Para Allocation strategy (Estrategia de asignación), elija la estrategia de asignación que se utilizará al seleccionar los tipos de instancia de la lista de tipos de instancia permitidos. BEST_FIT_PROGRESSIVE suele ser la mejor opción para los entornos de computación bajo demanda de EC2 y SPOT_CAPACITY_OPTIMIZED para los entornos de computación Spot de EC2. Para obtener más información, consulte the section called “Estrategias de asignación”.
18. (Opcional) En Configuración de EC2, seleccione Agregar configuración de EC2. Elija los valores de anulación de tipo de imagen e ID de imagen AWS Batch para proporcionar información sobre la selección de Amazon Machine Images (AMI) para las instancias del entorno informático. Si no se especifica la anulación del ID de imagen para cada tipo de imagen, AWS Batch selecciona una AMI reciente optimizada para Amazon ECS. Si no se especifica ningún tipo de imagen, el valor predeterminado es una instancia de Amazon Linux 2 para instancias que no sean de GPU ni de AWS Graviton.

    > **Important**
    >
    > Para usar una AMI personalizada, elija el tipo de imagen y, a continuación, introduzca el ID de AMI personalizado en el cuadro de Cambio de ID de imagen.

    Amazon Linux 2

    Es el valor predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo,, `C6g` `M6gR6g`, `yT4g`) y se puede usar para todos los tipos de instancias que no sean de GPU.

    Amazon Linux 2 (GPU)

    Es el valor predeterminado para todas las familias de instancias de GPU (por ejemplo, `P4` `yG4`) y se puede usar para todos los tipos de instancias que no estén basadas en AWS Graviton.

Amazon Linux

Se puede usar para familias de instancias que no utilizan GPU ni AWS Graviton. El soporte estándar para Amazon Linux ha finalizado. Para obtener más información, consulte AMI de Amazon Linux.

> **Note**
>
> La AMI que elija para un entorno de computación debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno de computación. Por ejemplo, si su entorno de computación utiliza tipos de instancias A1, la AMI de recursos de computación que elija debe admitir instancias Arm. Amazon ECS ofrece versiones x86 y Arm de la AMI Amazon Linux 2 optimizada para Amazon ECS. Para obtener más información, consulte la sección sobre AMI Amazon Linux 2 optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

19. (Opcional) En Plantilla de lanzamiento, seleccione una plantilla de lanzamiento de Amazon EC2 existente para configurar sus recursos de computación. La versión predeterminada de la plantilla se rellena automáticamente. Para obtener más información, consulte Compatibilidad con las plantillas de lanzamiento.

> **Note**
>
> En una plantilla de lanzamiento, puede especificar una AMI personalizada que haya creado.

20. (Opcional) En Launch template version (Versión de la plantilla de lanzamiento), introduzca `$Default`, `$Latest` o el número de versión específico que desea utilizar.

> **Important**
>
> Una vez creado el entorno de computación, la versión de la plantilla de lanzamiento utilizada no cambia, incluso aunque se actualice la versión `$Default` o `$Latest` de la plantilla de lanzamiento. Para utilizar una nueva versión de plantilla de lanzamiento, cree primero un nuevo entorno de computación y añádalo a la cola de trabajos existente. A continuación, quite el entorno de computación antiguo de la cola de trabajos y elimínelo.

21. En la sección Configuración de red:

    a. En Nube privada virtual (VPC), seleccione una Amazon VPC.

    b. En Subredes, se muestran las subredes de sus Cuenta de AWS . Si desea crear un conjunto personalizado de subredes, elija Borrar subredes y, a continuación, elija las subredes que desee.

    > **Important**
    >
    > Los recursos de computación deben comunicarse con el punto de conexión de VPC de Amazon ECS a través de un punto de conexión de VPC o de varias direcciones IP públicas. Para obtener más información, consulte Puntos de conexión de VPC de tipo interfaz de Amazon ECR (AWS PrivateLink). Si su instancia no tiene un punto de conexión de VPC configurado ni una dirección IP pública, puede usar la traducción de direcciones de red (NAT). Para obtener más información acerca de NAT, consulte Puertas de enlace de NAT y Creación de una nube virtual privada (VPC) .

    c. Para los Grupos de seguridad, elija los grupos de seguridad de Amazon EC2 que desee asociar a la instancia. Si desea crear un conjunto personalizado de grupos de seguridad, elija Borrar grupos de seguridad. Seleccione los grupos de seguridad que desea.

22. Elija Siguiente.

Crear una cola de trabajos

Una cola de trabajos almacena los trabajos enviados hasta que el AWS Batch programador ejecute el trabajo en un recurso de su entorno informático. Para obtener más información, consulte Colas de trabajo

Para crear una cola de trabajos para una orquestación de Amazon EC2, haga lo siguiente:

1. En la sección de Configuración de cola de trabajo de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. En Prioridad, introduzca un número entero entre 0 y 100 para la cola de trabajos.

> **Important**
>
> El programador AWS Batch asigna una prioridad mayor a los valores enteros más altos.

3. Elija Siguiente.

Creación de una definición de trabajo

AWS Batch las definiciones de trabajos especifican cómo se van a ejecutar los trabajos. Si bien cada trabajo debe hacer referencia a una definición de trabajo, muchos de los parámetros especificados en dicha definición pueden ser ignorados en tiempo de ejecución.

Para crear la definición de trabajo:

1. En la sección de Configuración general:

   a. En la sección de Configuración general de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. El nombre puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

   b. (Opcional) En Tiempo de espera de la ejecución, introduzca la cantidad de tiempo (en segundos) que tarda en finalizar un trabajo pendiente.

   > **Important**
   >
   > El tiempo de espera mínimo es de 60 segundos.

   c. (Opcional) Una etiqueta es una marca que se asigna a un recurso. Para añadir una etiqueta, expanda Etiquetas y, a continuación, seleccione Agregar etiqueta. Introduzca un par clave-valor y, a continuación, vuelva a seleccionar Agregar etiqueta.

   > **Important**
   >
   > Si elige Agregar etiqueta, debe introducir un par clave-valor y volver a elegir Agregar etiqueta o bien elegir Eliminar etiqueta.

d. (Opcional) Active Propagar etiquetas para propagar las etiquetas a la tarea de Amazon Elastic Container Service.

2. En la sección Configuración del contenedor:

a. En Imagen, introduzca el nombre de la imagen que se utiliza para lanzar el contenedor. Por defecto, todas las imágenes del registro de Docker Hub están disponibles. También puede especificar otros repositorios en formato repository-url/image:tag. El parámetro puede tener 255 caracteres como máximo. El parámetro puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), puntos (.), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a `Image` en la sección Crear un contenedor de la API remota de Docker y el parámetro `IMAGE` de docker run.

> **Note**
>
> La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos de computación en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en Arm solo pueden ejecutarse en recursos de computación basados en Arm.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository:tag` (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se identifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

b. En Comando, especifique los comandos que desea transmitir al contenedor. Este parámetro se asigna a `Cmd` en la sección Crear un contenedor de la API remota de Docker y el

parámetro `COMMAND` se corresponde con docker run. Para obtener más información sobre el parámetro `CMD` de Docker, consulte https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> También puede usar valores predeterminados de sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte Parámetros.

c. (Opcional) En Rol de ejecución, especifique un rol de IAM que conceda permiso a los agentes de contenedor de Amazon ECS para realizar llamadas a la API de AWS en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para las tareas. Para obtener más información, consulte Roles de IAM de ejecución de tareas de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

d. (Opcional) Para configurar el rol de trabajo, elija un rol de IAM que tenga permisos para las AWS API. Esta característica utiliza roles de IAM de Amazon ECS para las tareas. Para obtener más información, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

> **Note**
>
> Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para sus AWS Batch trabajos, consulte Creación de un rol y una política de IAM para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

e. (Opcional) Puede añadir parámetros a la definición del trabajo como asignaciones de clave-valor para anular los valores predeterminados de la definición del trabajo. Para añadir un parámetro:

- En Parámetros, elija Agregar parámetro. Introduzca un par clave-valor y, a continuación, vuelva a seleccionar Agregar parámetro.

> **Important**
>
> Si elige Agregar parámetro, debe configurar al menos un parámetro o elegir Eliminar parámetro.

f. En la sección Configuración de entorno para vCPU, especifique la cantidad de vCPU que quiera reservar para el contenedor. Este parámetro se asigna a `CpuShares` en la sección Crear un contenedor de la API remota de Docker y con la opción `--cpu-shares` de docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU.

g. En Memoria, especifique límite máximo (en MiB) de memoria que quiera presentarle al contenedor del trabajo. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se asigna a `Memory` en la sección Crear un contenedor de la API remota de Docker y con la opción `--memory` de docker run.

h. En Número de unidades GPU, seleccione el número de unidades GPU que desea reservar para el contenedor.

i. (Opcional) En configuración de variables de entorno, seleccione Agregar variables de entorno para añadir variables de entorno y pasarlas al contenedor. Este parámetro se asigna a `Env` en la sección Crear un contenedor de la API remota de Docker y con la opción `--env` de docker run.

j. (Opcional) En Secretos, seleccione Agregar secreto para añadir los secretos como pares de nombre-valor. Estos secretos están expuestos en el contenedor. Para obtener más información, consulte SecretOptions en Parámetros de definición de trabajo para ContainerProperties.

k. (Opcional) En la sección de Configuración de Linux:

   i. En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor. Este parámetro se asigna a `User` en la sección Crear un contenedor de la API remota de Docker y con la opción `--user` de docker run.

   ii. Para otorgar al contenedor de su trabajo permisos elevados en la instancia host (similares a los del usuario de `root`), arrastre el control deslizante Privilegiado hacia la derecha. Este parámetro se asigna a `Privileged` en la sección Crear un contenedor de la API remota de Docker y con la opción `--privileged` de docker run.

   iii. Active la opción Habilitar el proceso para ejecutar un proceso `init` dentro del contenedor. Este proceso reenvía señales y recoge procesos.

l. (Opcional) En la sección de Configuración de Filesystem:

i. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.

ii. En Tamaño de memoria compartida, introduzca el tamaño (en MiB) del `/dev/shm` volumen de .

iii. En Tamaño de intercambio máximo, introduzca la cantidad total de memoria de intercambio (en MiB) que puede utilizar el contenedor.

iv. En Intercambio, introduzca un valor entre 0 y 100 para indicar el comportamiento de intercambio del contenedor. Si no especifica un valor y el intercambio está activado, el valor predeterminado es 60. Para obtener más información, consulte Intercambio en Parámetros de definición de trabajo para ContainerProperties.

v. (Opcional) Expandir Configuración adicional.

vi. En el caso de Tmpfs, seleccione Agregar tmpfs para añadir una montura `tmpfs`.

vii. En el caso de los Dispositivos, seleccione Agregar dispositivo para añadir un dispositivo:

A. En Container path (Ruta del contenedor), especifique la ruta de la instancia del contenedor que va a exponer el dispositivo asignado a la instancia del host. Si lo deja en blanco, se utiliza la ruta del host en el contenedor.

B. En Host path (Ruta de host), especifique la ruta de un dispositivo de la instancia del host.

C. En la página Permisos, haga clic en uno o varios permisos para aplicarlos al dispositivo. Los permisos disponibles son READ, WRITE y MKNOD.

viii. (Opcional) En Configuración de Ulimits, seleccione Agregar ulimit para agregar un `ulimits` valor al contenedor. Introduzca los valores de Nombre, Límite flexible y Límite invariable y, a continuación, elija Agregar límite máximo.

3. Elija Siguiente.

Creación de un trabajo

Para crear un trabajo, haga lo siguiente:

1. En Configuración de trabajo, especifique un Nombre único para el trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

2. Elija Siguiente.

Revisar y crear

En la página Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de recursos.

Introducción: Fargate

AWS Fargate lanza y escala el cómputo para que se ajuste perfectamente a los requisitos de recursos que especifique para el contenedor. Con Fargate, no es necesario aprovisionar en exceso los servidores ni pagar por ellos. Para obtener más información, consulte Fargate.

Crear un entorno de computación

Para crear un entorno de computación para una orquestación de Fargate, haga lo siguiente:

1. Abra el Asistente de primer uso de la consola AWS Batch.
2. En Seleccione el tipo de orquestación, elija Fargate.
3. Elija Siguiente.
4. En la sección de Configuración de entorno de computación de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
5. (Opcional) Una etiqueta es una marca que se asigna a un recurso. Para añadir una etiqueta, expanda Etiquetas y, a continuación, seleccione Agregar etiqueta. Introduzca un par clave-valor y, a continuación, vuelva a seleccionar Agregar etiqueta.

   > **Important**
   >
   > Si elige Agregar etiqueta, debe introducir un par clave-valor y volver a elegir Agregar etiqueta o bien elegir Eliminar etiqueta.

6. (Opcional) En la sección Configuración de instancias para Usar capacidades de Fargate Spot, active Habilitar el uso de instancias de spot.
7. En Máximo de vCPU, introduzca la cantidad máxima de vCPU que puede usar la instancia.
8. En la sección Configuración de red:

a. En Nube privada virtual (VPC), seleccione una Amazon VPC.

b. En Subredes, se muestran las subredes de sus Cuenta de AWS . Si desea crear un conjunto personalizado de subredes, elija Borrar subredes y, a continuación, elija las subredes que desee.

> **Important**
>
> Los recursos de computación deben comunicarse con el punto de conexión de VPC de Amazon ECS a través de un punto de conexión de VPC o de varias direcciones IP públicas. Para obtener más información, consulte Puntos de conexión de VPC de tipo interfaz de Amazon ECR (AWS PrivateLink). Si su instancia no tiene un punto de conexión de VPC configurado ni una dirección IP pública, puede usar la traducción de direcciones de red (NAT). Para obtener más información acerca de NAT, consulte Puertas de enlace de NAT y Creación de una nube virtual privada (VPC) .

c. Para los Grupos de seguridad, elija los grupos de seguridad de Amazon EC2 que desee asociar a la instancia. Si desea crear un conjunto personalizado de grupos de seguridad, elija Borrar grupos de seguridad. Seleccione los grupos de seguridad que desea.

9. Elija Siguiente.

Crear una cola de trabajos

Una cola de trabajos almacena los trabajos enviados hasta que el AWS Batch programador ejecute el trabajo en un recurso de su entorno informático. Para crear una cola de trabajos:

Para crear una cola de trabajos para una orquestación de Fargate, haga lo siguiente:

1. En la sección de Configuración de cola de trabajo de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. En Prioridad, introduzca un número entero entre 0 y 100 para la cola de trabajos.

> **Important**
>
> El programador AWS Batch asigna una prioridad mayor a los valores enteros más altos.

3. Elija Siguiente.

Creación de una definición de trabajo

Para crear la definición de trabajo:

1. En la sección de Configuración general:

   a. En Nombre, introduzca un nombre de definición de trabajo personalizado.

      En la sección de Configuración general de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

   b. (Opcional) En Tiempo de espera de la ejecución, introduzca la cantidad de tiempo (en segundos) que tarda en finalizar un trabajo pendiente.

      > **Important**
      >
      > El tiempo de espera mínimo es de 60 segundos.

   c. (Opcional) Una etiqueta es una marca que se asigna a un recurso. Para añadir una etiqueta, expanda Etiquetas y, a continuación, seleccione Agregar etiqueta. Introduzca un par clave-valor y, a continuación, vuelva a seleccionar Agregar etiqueta.

      > **Important**
      >
      > Si elige Agregar etiqueta, debe introducir un par clave-valor y volver a elegir Agregar etiqueta o bien elegir Eliminar etiqueta.

   d. (Opcional) Active Propagar etiquetas para propagar las etiquetas a la tarea de Amazon Elastic Container Service.

2. En la sección de Configuración de la plataforma Fargate:

   a. (Opcional) En Versión de la plataforma Fargate, introduzca el tiempo de ejecución específico que desee.

   b. Para la Plataforma de tiempo de ejecución, seleccione LINUX o Windows.

   c. (Solo Windows) Para la Familia de sistemas operativos, seleccione un sistema operativo.

d. Para Arquitectura de CPU, seleccione la arquitectura de CPU que desee.

e. (Opcional) Active Asignar IP pública para asignar una dirección IP pública.

f. En Almacenamiento efímero, introduzca la cantidad de almacenamiento efímero que desee.

> **Note**
>
> De forma predeterminada, se utilizan 20 GiB de almacenamiento efímero. Para usar almacenamiento efímero adicional, introduzca un valor entre 21 GiB y 100 GiB.

g. Para la función de ejecución, elija una función de ejecución de tareas que permita a los agentes de Amazon Elastic Container Service (Amazon ECS) AWS realizar llamadas en su nombre. Por ejemplo, puede elegir ecsTaskExecutionRol.

3. En la sección Configuración del contenedor:

a. En Imagen, introduzca el nombre de la imagen que se utiliza para lanzar el contenedor. Por defecto, todas las imágenes del registro de Docker Hub están disponibles. También puede especificar otros repositorios en formato repository-url/image:tag. El parámetro puede tener 255 caracteres como máximo. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), puntos (.), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a `Image` en la sección Crear un contenedor de la API remota de Docker y el parámetro `IMAGE` de docker run.

> **Note**
>
> La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos de computación en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en Arm solo pueden ejecutarse en recursos de computación basados en Arm.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository:tag` (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).

- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se identifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

b. En Comando, especifique los comandos que desea transmitir al contenedor. Este parámetro se asigna a `Cmd` en la sección Crear un contenedor de la API remota de Docker y el parámetro `COMMAND` se corresponde con docker run. Para obtener más información sobre el parámetro `CMD` de Docker, consulte https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> También puede usar valores predeterminados de sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte Parámetros.

> **Tip**
>
> Seleccione Información para revisar los ejemplos de código de Bash y JSON.

c. (Opcional) Puede añadir parámetros a la definición del trabajo como asignaciones de clave-valor para anular los valores predeterminados de la definición del trabajo. Para añadir un parámetro:

- En Parámetros, elija Agregar parámetro. Introduzca un par clave-valor y, a continuación, vuelva a seleccionar Agregar parámetro.

> **Important**
>
> Si elige Agregar parámetro, debe configurar al menos un parámetro o elegir Eliminar parámetro.

d. (Opcional) En la sección Configuración del entorno para la configuración del rol de Job, elija un rol de IAM que otorgue permiso para usar las AWS API.

e. En la sección Configuración de entorno para vCPU, especifique la cantidad de vCPU que quiera reservar para el contenedor. Este parámetro se asigna a `CpuShares` en la sección Crear un contenedor de la API remota de Docker y con la opción `--cpu-shares` de docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU.

f. En Memoria, especifique límite máximo (en MiB) de memoria que quiera presentarle al contenedor del trabajo. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se asigna a `Memory` en la sección Crear un contenedor de la API remota de Docker y con la opción `--memory` de docker run.

g. (Opcional) En Variables de entorno, seleccione Agregar variables de entorno para añadir variables de entorno y pasarlas al contenedor. Este parámetro se asigna a `Env` en la sección Crear un contenedor de la API remota de Docker y con la opción `--env` de docker run.

4. Elija Siguiente.

Creación de un trabajo

Para crear un trabajo de Fargate, haga lo siguiente:

1. En Configuración de trabajo, especifique un Nombre único para el trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. Elija Siguiente.

Revisar y crear

En la página Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de recursos.

Cómo empezar con AWS Batch Amazon EKS

AWS Batch on Amazon EKS es un servicio gestionado para programar y escalar cargas de trabajo por lotes en clústeres de Amazon EKS existentes. AWS Batch no crea, administra ni realiza operaciones del ciclo de vida de sus clústeres de Amazon EKS en su nombre. AWS Batch la orquestación amplía y reduce los nodos administrados por dichos nodos AWS Batch y los ejecuta en ellos.

AWS Batch no afecta a los nodos, ni a los grupos de nodos con escalado automático ni a los ciclos de vida de los pods que no estén asociados a los entornos AWS Batch informáticos de su clúster de Amazon EKS. AWS Batch Para funcionar de forma eficaz, su función vinculada a un servicio necesita permisos de control Kubernetes de acceso basado en funciones (RBAC) en su clúster Amazon EKS existente. Para obtener más información, consulte Utilización de la autorización de RBAC en la documentación Kubernetes.

AWS Batch requiere un espacio de Kubernetes nombres en el que pueda clasificar los pods como trabajos. AWS Batch Recomendamos un espacio de nombres dedicado para aislar los AWS Batch pods de las demás cargas de trabajo del clúster.

Una vez AWS Batch que se le haya otorgado acceso a RBAC y se haya establecido un espacio de nombres, puede asociar ese clúster de Amazon EKS a un entorno AWS Batch informático mediante la operación de API. CreateComputeEnvironment Se puede asociar una cola de trabajos a este nuevo entorno informático de Amazon EKS. AWS Batch los trabajos se envían a la cola de trabajos en función de una definición de trabajo de Amazon EKS mediante la operación de SubmitJobAPI. AWS Batch a continuación, lanza los nodos AWS Batch gestionados y coloca los trabajos de la cola de trabajos en forma de Kubernetes pods en el clúster EKS asociado a un entorno AWS Batch informático.

En las siguientes secciones se explica cómo configurar AWS Batch Amazon EKS.

Contenido

- Paso 5: presentar un trabajo
- (Opcional) Envíe un trabajo con cambios
- Solución de problemas

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesita para crear y administrar tanto AWS Batch los recursos de Amazon EKS.

- AWS CLI: una herramienta de línea de comandos para trabajar con servicios de AWS , incluido Amazon EKS. Esta guía requiere que utilices la versión 2.8.6 o posterior o la 1.26.0 o posterior. Para obtener más información, consulte Instalar, actualizar y desinstalar la AWS CLI en la Guía del usuario de AWS Command Line Interface . Tras instalarlo AWS CLI, le recomendamos que también lo configure. Para obtener más información, consulte Configuración rápida con `aws configure` en la Guía del usuario de AWS Command Line Interface .
- **`kubectl`**: una herramienta de línea de comandos para trabajar con clústeres de Kubernetes. Esta guía requiere que utilice la versión `1.23` o una posterior. Para obtener más información, consulte Instalación o actualización de `kubectl` en la Guía del usuario de Amazon EKS.
- **`eksctl`**— Una herramienta de línea de comandos para trabajar con clústeres de Amazon EKS que automatiza muchas tareas individuales. Esta guía requiere que utilice la versión `0.115.0` o una posterior. Para obtener más información, consulte Instalación o actualización de `eksctl` en la Guía del usuario de Amazon EKS.
- Permisos de IAM necesarios: el responsable de seguridad de IAM que utilice debe tener permisos para trabajar con las funciones de IAM de Amazon EKS y las funciones vinculadas a servicios AWS CloudFormation, así como con una VPC y los recursos relacionados. Para obtener más información, consulte Acciones, recursos y claves de condición para Amazon Elastic Kubernetes Service y Uso de roles vinculados a servicios en la Guía del usuario de IAM. Debe completar todos los pasos de esta guía como el mismo usuario.
- Creación de un clúster de Amazon EKS: para obtener más información, consulte Introducción a Amazon EKS `eksctl` en la Guía del usuario de Amazon EKS.

> **Note**
>
> AWS Batch solo admite clústeres de Amazon EKS con puntos de enlace de servidor de API que tengan acceso público y sean accesibles desde la Internet pública. De forma predeterminada, los puntos finales del servidor de API de clústeres de Amazon EKS tienen

acceso público. Para obtener más información, consulte Control de acceso al punto de conexión del clúster de Amazon EKS en la Guía del usuario de Amazon EKS.

> **Note**
>
> AWS Batch no proporciona una organización de nodos gestionados para CoredNS u otros pods de implementación. Si necesita CoreDNS, consulte Agregar el complemento CoreDNS para Amazon EKS en la Guía del usuario de Amazon EKS. O bien, utilice `eksctl create cluster create` para crear el clúster, que incluye CoredNS de forma predeterminada.

- Permisos: los usuarios que llamen a la operación de CreateComputeEnvironmentAPI para crear un entorno de cómputo que utilice los recursos de Amazon EKS necesitan permisos para la operación de `eks:DescribeCluster` API. El uso de AWS Management Console para crear un recurso informático con los recursos de Amazon EKS requiere permisos `eks:DescribeCluster` tanto para como para`eks:ListClusters`.

Paso 1: Preparar el clúster de Amazon EKS para AWS Batch

Todos los pasos son obligatorios.

1. Cree un espacio de nombres dedicado a los trabajos AWS Batch

   Se utiliza `kubectl` para crear un nuevo espacio de nombres.

```
$ namespace=my-aws-batch-namespace
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Salida:

```
namespace/my-aws-batch-namespace created
```

2. Habilite el acceso a través del control de acceso basado en roles (RBAC)

   `kubectl`Utilícelo para crear un Kubernetes rol para el clúster que AWS Batch permita vigilar los nodos y los pods y vincular el rol. Debe hacerlo una vez para cada clúster de EKS.

   > **Note**
   >
   > Para obtener más información sobre el uso de la autorización RBAC, consulte Uso de la autorización RBAC en la Guía del usuario. Kubernetes

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
```

```
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Salida:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

Cree una Kubernetes función que abarque el espacio de nombres para gestionar los pods y su ciclo de vida y vincúlelos AWS Batch . Debe hacerlo una vez para cada espacio de nombres único.

```
$ namespace=my-aws-batch-namespace
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
```

```
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Salida:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
 created
```

Actualice el Kubernetes `aws-auth` mapa de configuración para asignar los permisos RBAC anteriores al rol vinculado al servicio. AWS Batch

```
$ eksctl create iamidentitymapping \
    --cluster my-cluster-name \
    --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
    --username aws-batch
```

Salida:

```
2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

> **Note**
>
> La ruta `aws-service-role/batch.amazonaws.com/` se ha eliminado del ARN del rol vinculado a un servicio. Esto se debe a un problema con el mapa de configuración de `aws-auth`. Para obtener más información, consulte Los roles con rutas no funcionan cuando la ruta está incluida en su ARN en. aws-authconfigmap

Paso 2: Creación de un entorno de computación de Amazon EKS

AWS Batch los entornos informáticos definen los parámetros de los recursos informáticos para satisfacer sus necesidades de carga de trabajo por lotes. En un entorno informático gestionado, le AWS Batch ayuda a gestionar la capacidad y los tipos de instancia de los recursos informáticos (Kubernetesnodos) de su clúster de Amazon EKS. Se basa en la especificación de recursos de computación que se define al crear el entorno de computación. Puede utilizar instancias bajo demanda EC2 o instancias de spot EC2.

Ahora que el rol AWSServiceRoleForBatchvinculado al servicio tiene acceso a su clúster de Amazon EKS, puede crear AWS Batch recursos. En primer lugar, cree un entorno de computación que apunte a su clúster de Amazon EKS.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
        "m5"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF
$ aws batch create-compute-environment --cli-input-json file://./batch-eks-compute-
environment.json
```

Notas

- No se debe especificar el `serviceRole` parámetro; en ese caso, se utilizará el rol AWS Batch vinculado al servicio. AWS Batch en Amazon EKS solo admite la función AWS Batch vinculada al servicio.
- Solo `BEST_FIT_PROGRESSIVE` se `SPOT_CAPACITY_OPTIMIZED` admiten estrategias de `SPOT_PRICE_CAPACITY_OPTIMIZED` asignación para los entornos informáticos de Amazon EKS.

> **Note**
>
> Le recomendamos que utilice `SPOT_PRICE_CAPACITY_OPTIMIZED` en vez de `SPOT_CAPACITY_OPTIMIZED` la mayoría de los casos.

- Para `instanceRole`, consulte Crear el rol de IAM del nodo de Amazon EKS y Habilitar el acceso principal de IAM a su clúster en la Guía del usuario de Amazon EKS. Si utiliza redes de pod, consulte Configuración del complemento de CNI de Amazon VPC para Kubernetes para utilizar roles de IAM en las cuentas de servicio en la Guía del usuario de Amazon EKS.
- Una forma de hacer que las subredes funcionen para el parámetro `subnets` consiste en utilizar las subredes públicas de los grupos de nodo administrados por Amazon EKS que creó `eksctl` al crear un clúster de Amazon EKS. De lo contrario, utilice subredes que tengan una ruta de red que permita extraer imágenes.
- El parámetro `securityGroupIds` puede utilizar el mismo grupo de seguridad que utiliza el clúster de Amazon EKS. Este comando recupera el ID del grupo de seguridad del clúster.

```
$ eks describe-cluster \
    --name <cluster-name> \
    --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- El mantenimiento de un entorno informático Amazon EKS es una responsabilidad compartida. Para obtener más información, consulte Responsabilidad compartida de los nodos Kubernetes.

> **Important**
>
> Es importante confirmar que el entorno de computación está en buen estado antes de continuar. Para ello, se puede utilizar la operación de la DescribeComputeEnvironmentsAPI.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Confirme que el parámetro `status` no es `INVALID`. Si es así, busque la causa en el parámetro `statusReason`. Para obtener más información, consulte Solución de problemas AWS Batch.

Paso 3: cree una cola de trabajos y adjunte el entorno de computación

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Los trabajos enviados a esta nueva cola de trabajos se ejecutan como pods en los nodos AWS Batch gestionados que se unieron al clúster de Amazon EKS asociado a su entorno informático.

```
$ cat <<EOF > ./batch-eks-job-queue.json
 {
    "jobQueueName": "My-Eks-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-CE1"
      }
    ]
  }
EOF
$ aws batch create-job-queue --cli-input-json file://./batch-eks-job-queue.json
```

Paso 4: crear una definición de trabajo

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
```

```
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-
definition.json
```

Notas

- Solo se admiten trabajos de un solo contenedor.
- Hay que tener en cuenta los parámetros `cpu` y `memory`. Para obtener más información, consulte Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS.

Paso 5: presentar un trabajo

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
    --job-definition MyJobOnEks_Sleep --job-name My-Eks-Job1
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Notas

- Solo se admiten trabajos de un solo contenedor.

- Asegúrese de estar familiarizado con todas las consideraciones relevantes para los parámetros `cpu` y `memory`. Para obtener más información, consulte Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS.
- Para obtener más información sobre la ejecución de trabajos en los recursos de Amazon EKS, consulteTrabajos de Amazon EKS.

(Opcional) Envíe un trabajo con cambios

Este trabajo anula el comando transferido al contenedor.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
$ aws batch submit-job --cli-input-json file://./submit-job-override.json
```

Notas

- AWS Batch limpia agresivamente las cápsulas una vez finalizadas las tareas para reducir la carga. Kubernetes Para examinar los detalles de un trabajo, se debe configurar el registro. Para obtener más información, consulte Utilice CloudWatch Logs para supervisar AWS Batch en los trabajos de Amazon EKS.

- Para mejorar la visibilidad de los detalles de las operaciones, habilite el registro del plano de control de Amazon EKS. Para obtener más información, consulte Registros del plano de control del clúster de Amazon EKS en la Guía del usuario de Amazon EKS.
- La sobrecarga Daemonsets y kubelets afectan a los recursos de vCPU y de memoria disponibles, específicamente al escalado y la colocación de trabajos. Para obtener más información, consulte Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS.

Introducción a AWS Batch Amazon EKS Private Clusters

AWS Batch es un servicio gestionado que organiza las cargas de trabajo por lotes en los clústeres de Amazon Elastic Kubernetes Service (Amazon EKS). Esto incluye la creación de colas, el seguimiento de las dependencias, la gestión de los reintentos y las prioridades de los trabajos, la administración de los pods y el escalado de los nodos. Esta función conecta su clúster privado de Amazon EKS existente con el AWS Batch fin de ejecutar sus trabajos a escala. Puede utilizar `eksctl`(una interfaz de línea de comandos para Amazon EKS), la AWS consola o la AWS Command Line Interfacepara crear un clúster privado de Amazon EKS con todos los demás recursos necesarios. El soporte para clústeres privados de Amazon EKS AWS Batch está generalmente disponible en formato comercial, Regiones de AWS donde AWS Batch esté disponible.

Los clústeres exclusivos privados de Amazon EKS no tienen acceso a Internet entrante ni saliente y solo tienen subredes privadas. Los puntos de enlace de Amazon VPC se utilizan para permitir el acceso privado a otros servicios. AWS `eksctl`admite la creación de clústeres totalmente privados mediante una Amazon VPC y subredes preexistentes. `eksctl`también crea puntos de enlace de Amazon VPC en la Amazon VPC suministrada y modifica las tablas de enrutamiento de las subredes suministradas.

Cada subred debe tener asociada una tabla de enrutamiento explícita, ya que `eksctl` no modifica la tabla de enrutamiento principal. El clúster debe extraer imágenes de un registro de contenedores que se encuentre en su Amazon VPC. Además, puede crear un Amazon Elastic Container Registry en su Amazon VPC y copiar en él las imágenes del contenedor para que las extraigan sus nodos. Para obtener más información, consulte Copiar una imagen de contenedor de un repositorio a otro repositorio. Para empezar a utilizar los repositorios privados de Amazon ECR, consulte Repositorios privados de Amazon ECR.

Si lo desea, puede crear una regla de extracción de caché con Amazon ECR. Una vez creada una regla de extracción de caché para un registro público externo, puede extraer una imagen de ese registro público externo mediante el identificador de recursos (URI) uriforme del registro privado de

Amazon ECR. A continuación, Amazon ECR crea un repositorio y almacena la imagen en caché. Cuando se extrae una imagen en caché mediante el URI del registro privado de Amazon ECR, Amazon ECR comprueba el registro remoto para comprobar si hay una nueva versión de la imagen y actualiza el registro privado hasta una vez cada 24 horas.

Contenido

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesita para crear y administrar tanto AWS Batch los recursos de Amazon EKS. También debe crear todos los recursos necesarios, como la VPC, las subredes, las tablas de rutas, los puntos de enlace de la VPC y el clúster de Amazon EKS. Debe usar el. AWS CLI

- AWS CLI— Una herramienta de línea de comandos para trabajar con AWS servicios, incluido Amazon EKS. Esta guía requiere que utilices la versión 2.8.6 o posterior o la 1.26.0 o posterior. Para obtener más información, consulte Instalar, actualizar y desinstalar la AWS CLI en la Guía del usuario de AWS Command Line Interface .

  Tras instalarlo AWS CLI, le recomendamos que lo configure. Para obtener más información, consulte Configuración rápida con `aws configure` en la Guía del usuario de AWS Command Line Interface .
- **`kubectl`**— Una herramienta de línea de comandos para trabajar con Kubernetes clústeres. Esta guía requiere que utilice la versión `1.23` o una posterior. Para obtener más información, consulte Instalación o actualización de `kubectl` en la Guía del usuario de Amazon EKS.
- **`eksctl`**— Una herramienta de línea de comandos para trabajar con clústeres de Amazon EKS que automatiza muchas tareas individuales. Esta guía requiere que utilice la versión `0.115.0` o

una posterior. Para obtener más información, consulte Instalación o actualización de `eksctl` en la Guía del usuario de Amazon EKS.

- Permisos obligatorios AWS Identity and Access Management (IAM): el responsable de seguridad de IAM que utilice debe tener permisos para trabajar con las funciones de IAM de Amazon EKS y las funciones vinculadas a servicios AWS CloudFormation, así como con una VPC y los recursos relacionados. Para obtener más información, consulte Acciones, recursos y claves de condición para Amazon Elastic Kubernetes Service y Uso de roles vinculados a servicios en la Guía del usuario de IAM. Debe completar todos los pasos de esta guía como el mismo usuario.
- Creación de un clúster de Amazon EKS: para obtener más información, consulte Introducción a Amazon EKS `eksctl` en la Guía del usuario de Amazon EKS.

> **Note**
>
> AWS Batch no proporciona una organización de nodos gestionados para CoredNS u otros pods de implementación. Si necesita CoreDNS, consulte Agregar el complemento CoreDNS para Amazon EKS en la Guía del usuario de Amazon EKS. O bien, utilice `eksctl create cluster create` para crear el clúster, que incluye CoredNS de forma predeterminada.

- Permisos: los usuarios que llamen a la operación de CreateComputeEnvironmentAPI para crear un entorno de cómputo que utilice los recursos de Amazon EKS necesitan permisos para la operación de `eks:DescribeCluster` API. El uso de AWS Management Console para crear un recurso informático con recursos de Amazon EKS requiere permisos `eks:DescribeCluster` tanto para como para`eks:ListClusters`.
- Cree un clúster EKS privado en la región us-east-1 con el archivo de `eksctl` configuración de ejemplo.

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
availabilityZones:
    - us-east-1a
    - us-east-1b
    - us-east-1d
managedNodeGroups:
    privateNetworking: true
privateCluster:
    enabled: true
    skipEndpointCreation: false
```

Cree sus recursos con el comando: `eksctl create cluster -f clusterConfig.yaml`

- Los nodos gestionados por lotes se deben implementar en subredes que tengan los puntos finales de interfaz de VPC que necesite. Para obtener más información, consulte Requisitos de los clústeres privados.

Paso 1: Preparar el clúster de EKS para AWS Batch

Todos los pasos son obligatorios.

1. Cree un espacio de nombres dedicado a los trabajos AWS Batch

   Se utiliza `kubectl` para crear un nuevo espacio de nombres.

   ```
   $ namespace=my-aws-batch-namespace
   $ cat - <<EOF | kubectl create -f -
   {
     "apiVersion": "v1",
     "kind": "Namespace",
     "metadata": {
       "name": "${namespace}",
       "labels": {
         "name": "${namespace}"
       }
     }
   }
   EOF
   ```

   Salida:

   ```
   namespace/my-aws-batch-namespace created
   ```

2. Habilite el acceso a través del control de acceso basado en roles (RBAC)

   Se utiliza `kubectl` para crear un rol de Kubernetes para el clúster que permita a AWS Batch vigilar los nodos y los pods y vincular el rol. Debe hacerlo una vez para cada clúster de Amazon EKS.

> **Note**
>
> Para obtener más información sobre el uso de la autorización de RBAC, consulte Uso de la autorización de RBAC en la Documentación de Kubernetes.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
```

```
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Salida:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

Cree un Kubernetes rol dentro del espacio de nombres para gestionar y vincular los pods durante todo AWS Batch el ciclo de vida. Debe hacerlo una vez para cada espacio de nombres único.

```
$ namespace=my-aws-batch-namespace
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
```

```
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Salida:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
 created
```

Actualice el Kubernetes `aws-auth` mapa de configuración para asignar los permisos RBAC anteriores al rol vinculado al servicio. AWS Batch

```
$ eksctl create iamidentitymapping \
    --cluster my-cluster-name \
    --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
    --username aws-batch
```

Salida:

```
2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

> **Note**
>
> La ruta `aws-service-role/batch.amazonaws.com/` se ha eliminado del ARN del rol vinculado a un servicio. Esto se debe a un problema con el mapa de configuración de `aws-auth`. Para obtener más información, consulte Los roles con rutas no funcionan cuando la ruta está incluida en su ARN en. aws-authconfigmap

Paso 2: Creación de un entorno de computación de Amazon EKS

AWS Batch los entornos informáticos definen los parámetros de los recursos informáticos para satisfacer sus necesidades de carga de trabajo por lotes. En un entorno informático gestionado, le AWS Batch ayuda a gestionar la capacidad y los tipos de instancia de los recursos informáticos (Kubernetesnodos) de su clúster de Amazon EKS. Se basa en la especificación de recursos de

computación que se define al crear el entorno de computación. Puede utilizar instancias bajo demanda EC2 o instancias de spot EC2.

Ahora que el rol AWSServiceRoleForBatchvinculado al servicio tiene acceso a su clúster de Amazon EKS, puede crear AWS Batch recursos. En primer lugar, cree un entorno de computación que apunte a su clúster de Amazon EKS.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
        "m5"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-the-image-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF
$ aws batch create-compute-environment --cli-input-json file://./batch-eks-compute-
environment.json
```

Notas

- No se debe especificar el `serviceRole` parámetro; en ese caso, se utilizará el rol AWS Batch vinculado al servicio. AWS Batch en Amazon EKS solo admite la función AWS Batch vinculada al servicio.

- Solo `BEST_FIT_PROGRESSIVE` se `SPOT_CAPACITY_OPTIMIZED` admiten estrategias de `SPOT_PRICE_CAPACITY_OPTIMIZED` asignación para los entornos informáticos de Amazon EKS.

  > **Note**
  >
  > Se recomienda utilizar `SPOT_PRICE_CAPACITY_OPTIMIZED` en lugar de `SPOT_CAPACITY_OPTIMIZED` en la mayoría de los casos.

- Para `instanceRole`, consulte Crear el rol de IAM del nodo de Amazon EKS y Habilitar el acceso principal de IAM a su clúster en la Guía del usuario de Amazon EKS. Si utiliza redes de pod, consulte Configuración del complemento de CNI de Amazon VPC para Kubernetes para utilizar roles de IAM en las cuentas de servicio en la Guía del usuario de Amazon EKS.
- Una forma de hacer que las subredes funcionen para el parámetro `subnets` consiste en utilizar las subredes públicas de los grupos de nodo administrados por Amazon EKS que creó `eksctl` al crear un clúster de Amazon EKS. De lo contrario, utilice subredes que tengan una ruta de red que permita extraer imágenes.
- El parámetro `securityGroupIds` puede utilizar el mismo grupo de seguridad que utiliza el clúster de Amazon EKS. Este comando recupera el ID del grupo de seguridad del clúster.

```
$ eks describe-cluster \
    --name <cluster-name> \
    --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- El mantenimiento de un entorno informático Amazon EKS es una responsabilidad compartida. Para obtener más información, consulte Seguridad en Amazon EKS.

> **Important**
>
> Es importante confirmar que el entorno de computación está en buen estado antes de continuar. Para ello, se puede utilizar la operación DescribeComputeEnvironmentsAPI.
>
> ```
> $ aws batch describe-compute-environments --compute-environments My-Eks-CE1
> ```

Confirme que el parámetro `status` no es `INVALID`. Si es así, busque la causa en el parámetro `statusReason`. Para obtener más información, consulte Solución de problemas AWS Batch.

Paso 3: cree una cola de trabajos y adjunte el entorno de computación

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Los trabajos enviados a esta nueva cola de trabajos se ejecutan como pods en los nodos AWS Batch gestionados que se unieron al clúster de Amazon EKS asociado a su entorno informático.

```
$ cat <<EOF > ./batch-eks-job-queue.json
 {
    "jobQueueName": "My-Eks-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-CE1"
      }
    ]
  }
EOF
$ aws batch create-job-queue --cli-input-json file://./batch-eks-job-queue.json
```

Paso 4: crear una definición de trabajo

En el campo de imagen de la definición del trabajo, en lugar de proporcionar un enlace a la imagen de un repositorio de ECR público, proporcione el enlace a la imagen almacenada en nuestro repositorio de ECR privado. Consulte el siguiente ejemplo de definición de trabajo:

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
```

```
        {
          "image": "account-id.dkr.ecr.region.amazonaws.com/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-
definition.json
```

Para ejecutar los comandos de kubectl, necesitará acceso privado a su clúster de Amazon EKS. Esto significa que todo el tráfico al servidor de API del clúster debe provenir de la VPC del clúster o de una red conectada.

Paso 5: presentar un trabajo

```
$ aws batch submit-job - -job-queue My-Eks-JQ1 \
    - -job-definition MyJobOnEks_Sleep - -job-name My-Eks-Job1
$ aws batch describe-jobs - -job <jobId-from-submit-response>
```

Notas

- Solo se admiten trabajos de un solo contenedor.
- Asegúrese de estar familiarizado con todas las consideraciones relevantes para los parámetros `cpu` y `memory`. Para obtener más información, consulte Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS.

- Para obtener más información sobre la ejecución de trabajos en los recursos de Amazon EKS, consulteTrabajos de Amazon EKS.

(Opcional) Envíe un trabajo con cambios

Este trabajo anula el comando transferido al contenedor.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
$ aws batch submit-job - -cli-input-json file://./submit-job-override.json
```

Notas

- AWS Batch limpia agresivamente las cápsulas una vez finalizadas las tareas para reducir la carga. Kubernetes Para examinar los detalles de un trabajo, se debe configurar el registro. Para obtener más información, consulte Utilice CloudWatch Logs para supervisar AWS Batch en los trabajos de Amazon EKS.
- Para mejorar la visibilidad de los detalles de las operaciones, habilite el registro del plano de control de Amazon EKS. Para obtener más información, consulte Registros del plano de control del clúster de Amazon EKS en la Guía del usuario de Amazon EKS.

- La sobrecarga Daemonsets y kubelets afectan a los recursos de vCPU y de memoria disponibles, específicamente al escalado y la colocación de trabajos. Para obtener más información, consulte Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS.

Solución de problemas

Si los nodos lanzados por AWS Batch no tienen acceso al repositorio de Amazon ECR (o a ningún otro repositorio) que almacene su imagen, sus trabajos podrían permanecer en el estado STARTING. Esto se debe a que el pod no podrá descargar la imagen ni ejecutar su AWS Batch trabajo. Si haces clic en el nombre del pod lanzado por, AWS Batch deberías poder ver el mensaje de error y confirmar el problema. El mensaje de error debería tener un aspecto similar al siguiente:

```
Failed to pull image "public.ecr.aws/amazonlinux/amazonlinux:2": rpc error: code =
Unknown desc = failed to pull and unpack image
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to resolve reference
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to do request: Head
"https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/2": dial tcp: i/o timeout
```

Para ver otros escenarios de solución de problemas comunes, consulte Solución de problemas AWS Batch. Para solucionar problemas relacionados con el estado del pod, consulte ¿Cómo soluciono los problemas del estado del pod en Amazon EKS? .

Jobs

Los trabajos son la unidad de trabajo con la que se empieza. AWS Batch Los trabajos se pueden invocar como aplicaciones en contenedores que se ejecutan en instancias de contenedor de Amazon ECS dentro de un clúster de ECS.

Los trabajos en contenedores pueden hacer referencia a una imagen, comando o parámetros de contenedor. Para obtener más información, consulte Parámetros de definición de trabajo para ContainerProperties.

Puede enviar una gran cantidad de trabajos sencillos e independientes.

Temas

Enviar un trabajo

Después de registrar una definición de trabajo, puede enviarla como trabajo a una cola de AWS Batch trabajos. Muchos de los parámetros que se especifican en la definición de tareas pueden ignorarse en tiempo de ejecución.

Para enviar un trabajo

1. Abra la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, seleccione Trabajos.
4. Seleccione Enviar el trabajo.
5. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
6. En Definición de trabajo, seleccione una definición de trabajo creada anteriormente. Para obtener más información, consulte Creación de una definición de trabajo de un solo nodo .
7. En Cola de trabajos, elija una cola de trabajos existente. Para obtener más información, consulte Cómo crear de una cola de trabajos.
8. En Dependencias de trabajos, elija Añadir dependencias de trabajos.
   - En ID de trabajo, introduzca el ID de trabajo de cualquier dependencia. A continuación, seleccione Añadir dependencias de trabajos. Un trabajo puede tener hasta 20 dependencias. Para obtener más información, consulte Dependencias de trabajos.
9. (Solo trabajos de matrices) En Tamaño de matriz, especifique un tamaño de matriz comprendido entre 2 y 10 000.
10. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Elija Agregar nueva etiqueta e introduzca la clave y el valor opcional.
11. Seleccione Página siguiente.
12. En la sección Anulaciones de trabajos:
    a. (Opcional) En Prioridad de programación, introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad.
    b. (Opcional) En Intentos de trabajo, introduzca el número máximo de veces que AWS Batch intenta mover el trabajo a un estado `RUNNABLE`. Puede especificar un número comprendido entre 1 y 10. Para obtener más información, consulte Reintentos automáticos de trabajo.
    c. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera el tiempo de espera, se detiene y el

estado cambia a `FAILED`. Para obtener más información, consulte Tiempos de espera de trabajo. El valor mínimo es de 60 segundos.

> **Important**
>
> No confíe en que los trabajos que se ejecuten con los recursos de Fargate duren más de 14 días. Después de 14 días, es posible que los recursos de Fargate ya no estén disponibles y es probable que el trabajo se finalice.

d. (Opcional) Active Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo a la tarea de Amazon ECS.

13. Expanda Configuración adicional.
14. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:

    - Reintentar: AWS Batch se vuelve a intentar hasta alcanzar el número de intentos de trabajo que especificó.
    - Salir: AWS Batch deja de volver a intentar el trabajo.

    > **Important**
    >
    > Si elige Añadir evaluación al salir, configure al menos un parámetro y elija una Acción o elija Eliminar evaluación al salir.

15. En Parámetros, seleccione Añadir parámetros para añadir marcadores de sustitución de parámetros. Ingrese una clave y un valor opcional.
16. En la sección Anulaciones de contenedores:

    a. En Comando, especifique los comandos que desea transmitir al contenedor. Para comandos sencillos, introdúzcalo del mismo modo que lo haría para una línea de comandos. Para comandos más complicados (por ejemplo, con caracteres especiales), utilice la sintaxis JSON.

> **Note**
>
> Este parámetro no puede contener una cadena vacía.

b. En CPU virtuales, introduzca la cantidad de CPU virtuales que quiera reservar para el contenedor. Este parámetro se asigna a `CpuShares` en la sección Crear un contenedor de la API remota de Docker y con la opción `--cpu-shares` de docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.

c. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se asigna a `Memory` en la sección Crear un contenedor de la API remota de Docker y con la opción `--memory` de docker run. Debe especificar al menos 4 MiB de memoria para un trabajo.

> **Note**
>
> Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte Administración de la memoria de los recursos informáticos de las .

d. (Opcional) En Cantidad de GPU, seleccione la cantidad de GPU que desea reservar para el contenedor.

e. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.

f. Seleccione Página siguiente.

g. En Revisión del trabajo, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Crear definición de trabajo.

Estados de trabajo

Al enviar un trabajo a una cola de AWS Batch trabajos, el trabajo entra en ese estado. `SUBMITTED` A continuación, pasa por los estados siguientes hasta que termina de ejecutarse correctamente (finaliza con el código `0`) o no (finaliza con un código distinto de cero). Los trabajos de AWS Batch pueden tener los siguientes estados:

`SUBMITTED`

Un trabajo que se ha enviado a la cola, y que aún no ha sido evaluado por el programador. El programador evalúa el trabajo para determinar si tiene alguna dependencia pendiente de la correcta finalización de cualquier otro trabajo. Si hay dependencias, el trabajo pasa a `PENDING`. Si no hay dependencias, el trabajo pasa a `RUNNABLE`.

`PENDING`

Un trabajo que está en la cola y que aún no se ha podido ejecutar debido a una dependencia de otro trabajo o recurso. Una vez se cumplan las dependencias, el trabajo pasa a `RUNNABLE`.

`RUNNABLE`

Un trabajo que está en la cola, que no tiene dependencias pendientes y, por tanto, que está listo para ser programado para un host. Los trabajos en este estado se inician tan pronto como haya recursos suficientes disponibles en alguno de los entornos de computación que se asignan a la cola del trabajo. Sin embargo, un trabajo puede permanecer en este estado de forma indefinida si los recursos suficientes no están disponibles.

> **Note**
>
> Si los trabajos no progresan a `STARTING`, consulte Trabajos bloqueados en estado `RUNNABLE` en la sección de resolución de problemas.

`STARTING`

Estos trabajos se han programado para un host y las operaciones de inicio de contenedor pertinentes están en curso. Después de que la imagen de contenedor se extraiga y el contenedor esté en marcha, el trabajo pasa a `RUNNING`.

`RUNNING`

El trabajo se ejecuta como un trabajo de contenedor en una instancia de contenedor de Amazon ECS dentro de un entorno de computación. Cuando el contenedor del trabajo se cierra, el código del proceso de salida determina si el trabajo ha finalizado correctamente o si ha fallado. Un código de salida `0` indica una correcta ejecución y cualquier código de salida distinto de cero indica error. Si al trabajo asociado a un intento fallido le quedan intentos en su configuración de estrategia de reintento opcional, el trabajo vuelve a pasar al estado `RUNNABLE`. Para obtener más información, consulte Reintentos automáticos de trabajo.

> **Note**
>
> Los registros de los `RUNNING` trabajos están disponibles en CloudWatch Registros. El grupo de registros es `/aws/batch/job`, y el formato del nombre del flujo de registro es el siguiente: *`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`*. Sin embargo, esto puede cambiar en el futuro.
> Una vez que un trabajo alcanza el `RUNNING` estado, puedes recuperar mediante programación su nombre de flujo de registro con la operación de la DescribeJobsAPI. Para obtener más información, consulte Ver los datos de registro enviados a CloudWatch los registros en la Guía del usuario de Amazon CloudWatch Logs. De forma predeterminada, estos registros nunca caducan. Puede modificar el periodo de retención de copia de seguridad. Para obtener más información, consulte Cambiar la retención de datos de registro en CloudWatch los registros en la Guía del usuario de Amazon CloudWatch Logs.

`SUCCEEDED`

El trabajo se ha completado correctamente con un código de salida `0`. El estado de los `SUCCEEDED` trabajos se mantiene AWS Batch durante al menos 7 días.

> **Note**
>
> Los registros de los `SUCCEEDED` trabajos están disponibles en CloudWatch Registros. El grupo de registros es `/aws/batch/job`, y el formato del nombre del flujo de registro es el siguiente: *`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`*. Este formato puede cambiar en el futuro.
> Una vez que un trabajo alcanza el `RUNNING` estado, puedes recuperar mediante programación su nombre de flujo de registro con la operación de la DescribeJobsAPI. Para obtener más información, consulte Ver los datos de registro enviados a CloudWatch los registros en la Guía del usuario de Amazon CloudWatch Logs. De forma predeterminada, estos registros nunca caducan. Puede modificar el periodo de retención de copia de seguridad. Para obtener más información, consulte Cambiar la retención de datos de registro en CloudWatch los registros en la Guía del usuario de Amazon CloudWatch Logs.

`FAILED`

El trabajo ha fallado en todos los intentos disponibles. El estado de trabajos `FAILED` persiste en AWS Batch durante al menos 7 días.

> **Note**
>
> Los registros de los `FAILED` trabajos están disponibles en CloudWatch Logs. El grupo de registros es `/aws/batch/job`, y el formato del nombre del flujo de registro es el siguiente: *`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`*. Este formato puede cambiar en el futuro.
> Una vez que un trabajo alcanza el `RUNNING` estado, puedes recuperar su flujo de registros mediante programación con la operación de la DescribeJobsAPI. Para obtener más información, consulte Ver los datos de registro enviados a CloudWatch los registros en la Guía del usuario de Amazon CloudWatch Logs. De forma predeterminada, estos registros nunca caducan. Puede modificar el periodo de retención de copia de seguridad. Para obtener más información, consulte Cambiar la retención de datos de registro en CloudWatch los registros en la Guía del usuario de Amazon CloudWatch Logs.

AWS Batch variables de entorno laboral

AWS Batch establece variables de entorno específicas en los trabajos de contenedores. Estas variables de entorno proporcionan una visión introspectiva de los contenedores que se encuentran dentro de los trabajos. Puede usar los valores de estas variables en la lógica de sus aplicaciones. Todas las variables que AWS Batch se configuran comienzan con el `AWS_BATCH_` prefijo. Se trata de un prefijo de variable de entorno protegido. No puede usar este prefijo para sus propias variables en las definiciones o anulaciones de trabajos.

Las variables de entorno siguientes están disponibles en los contenedores de trabajos:

`AWS_BATCH_CE_NAME`

Esta variable se establece en el nombre del entorno de computación en el que está situado el trabajo.

`AWS_BATCH_JOB_ARRAY_INDEX`

Esta variable solo se establece en los trabajos de matrices secundarios. El índice de trabajo de matriz empieza en 0, y cada trabajo secundario recibe un número de índice único. Por ejemplo,

un trabajo de matriz con 10 elementos secundarios tiene valores de índice comprendidos entre 0 y 9. Puede utilizar este valor de índice para controlar la forma en la que se diferencian los elementos secundarios del trabajo de matriz. Para obtener más información, consulte Tutorial: Uso del índice de trabajo de matriz para controlar la diferenciación de trabajos.

`AWS_BATCH_JOB_ARRAY_SIZE`

Esta variable se establece en el tamaño del trabajo de matriz principal. El tamaño del trabajo de matriz principal se pasa al trabajo de matriz secundario en esta variable.

`AWS_BATCH_JOB_ATTEMPT`

Esta variable se establece en el número de reintentos de trabajo. Al primer intento se le asigna el valor 1. Para obtener más información, consulte Reintentos automáticos de trabajo.

`AWS_BATCH_JOB_ID`

Esta variable se establece en el ID del AWS Batch trabajo.

`AWS_BATCH_JOB_KUBERNETES_NODE_UID`

Esta variable se establece como el Kubernetes UID del objeto de nodo que se encuentra en el clúster de Kubernetes en el que se ejecuta el pod. Esta variable solo se establece para los trabajos que se ejecutan en los recursos de Amazon EKS. Para obtener más información, consulte UIDs en la Kubernetes documentación.

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable se establece en el número de índice del nodo principal del trabajo. El código de la aplicación puede comparar `AWS_BATCH_JOB_MAIN_NODE_INDEX` con `AWS_BATCH_JOB_NODE_INDEX` en un nodo individual para determinar si es el nodo principal.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable no está presente en el nodo principal, pero se establece en la dirección IPv4 privada del nodo principal del trabajo. El código de la aplicación del nodo secundario puede utilizar esta dirección para comunicarse con el nodo principal.

`AWS_BATCH_JOB_NODE_INDEX`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable se establece en el número de índice del nodo. El índice del nodo comienza a partir de 0 y cada

nodo recibe un número de índice único. Por ejemplo, un trabajo paralelo de varios nodos con 10 elementos secundarios tiene valores de índice comprendidos entre 0 y 9.

`AWS_BATCH_JOB_NUM_NODES`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable se establece en el número de nodos solicitados para el trabajo en paralelo de varios nodos.

`AWS_BATCH_JQ_NAME`

Esta variable se establece en el nombre de la cola de trabajos a la que se ha enviado el trabajo.

Reintentos automáticos de trabajo

Puede aplicar una estrategia de reintento a los trabajos y las definiciones de trabajo que permita reintentar automáticamente la ejecución de los trabajos con errores. Entre las posibles situaciones de error se incluyen:

- Cualquier código de salida distinto de cero a partir de un trabajo de contenedor
- Una finalización o error de la instancia de Amazon EC2
- Error o interrupción del servicio interno AWS

El acto de enviar un trabajo a una cola de trabajos y que pase al estado `RUNNING` se considera un intento. De forma predeterminada, a cada trabajo se le concede un intento para pasar al estado `SUCCEEDED` o `FAILED`. Sin embargo, la definición de trabajo y los flujos de envíos de trabajo permiten especificar una estrategia de entre 1 y 10 reintentos. Si evaluateOnExitse especifica, puede contener hasta 5 estrategias de reintentos. Si evaluateOnExitse especifica, pero ninguna de las estrategias de reintento coincide, se vuelve a intentar el trabajo. En el caso de los trabajos que no coincidan con la salida, añada una última entrada que salga por cualquier motivo. Por ejemplo, este objeto `evaluateOnExit` tiene dos entradas con acciones de `RETRY` y una última entrada con una acción de `EXIT`.

```
"evaluateOnExit": [
    {
        "action": "RETRY",
        "onReason": "AGENT"
    },
    {
        "action": "RETRY",
```

```
        "onStatusReason": "Task failed to start"
    },
    {
        "action": "EXIT",
        "onReason": "*"
    }
]
```

En la ejecución, la variable de entorno `AWS_BATCH_JOB_ATTEMPT` se ajusta a la cantidad de reintentos de trabajo de contenedor correspondiente. Al primer intento se le asigna el 1, y los intentos posteriores se enumeran en orden ascendente (por ejemplo, 2, 3, 4, etc.).

Por ejemplo, supongamos que un intento de trabajo falla por cualquier motivo y que el número de intentos especificado en la configuración de reintentos es superior al número de `AWS_BATCH_JOB_ATTEMPT`. A continuación, el trabajo se vuelve a colocar en el estado `RUNNABLE`. Para obtener más información, consulte Estados de trabajo.

> **Note**
>
> Los trabajos que se cancelan o se terminan no se reintentan. De la misma manera, los trabajos que fallan debido a una definición de trabajo no válida tampoco se reintentan.

Para obtener más información, consulte Estrategia de reintento, Creación de una definición de trabajo de un solo nodo , Enviar un trabajo y Códigos de error de tareas detenidas.

Dependencias de trabajos

Al enviar un AWS Batch trabajo, puede especificar los identificadores del trabajo de los que depende el trabajo. Al hacerlo, el programador de AWS Batch garantiza que el trabajo solo se ejecuta después de que las dependencias especificadas hayan finalizado correctamente. Una vez que estas terminen correctamente, el trabajo dependiente pasa del estado `PENDING` al estado `RUNNABLE` y, a continuación, a `STARTING` y a `RUNNING`. Si alguna de las dependencias del trabajo produce un error, el trabajo dependiente pasa automáticamente de `PENDING` a `FAILED`.

Por ejemplo, Job A puede expresar una dependencia de hasta 20 trabajos distintos que deben completarse correctamente para que pueda ejecutarse. A continuación, puede enviar trabajos adicionales que dependan de Job A y de hasta otros 19 trabajos.

En los trabajos de matrices, puede especificar una dependencia de tipo `SEQUENTIAL` sin especificar un ID de trabajo para que cada trabajo de matriz secundario se complete de forma secuencial, comenzando a partir del índice 0. También puede especificar una dependencia de tipo `N_TO_N` con un ID de trabajo. De esta forma, cada índice secundario de este trabajo debe esperar a que se complete el índice secundario correspondiente de cada dependencia antes de comenzar. Para obtener más información, consulte Trabajos de matrices.

Para enviar un AWS Batch trabajo con dependencias, consulteEnviar un trabajo.

Tiempos de espera de trabajo

Puede configurar un tiempo de espera para sus trabajos de modo que, si un trabajo dura más tiempo, AWS Batch finalice el trabajo. Por ejemplo, es posible que tenga un trabajo que sabe que solo debería tardar 15 minutos en completarse. A veces la aplicación se bloquea en un bucle y se ejecuta para siempre, por lo que puede establecer un tiempo de espera de 30 minutos para terminar el trabajo bloqueado.

> **Important**
>
> De forma predeterminada, AWS Batch no tiene un tiempo de espera para el trabajo. Si no define un tiempo de espera para el trabajo, el trabajo se ejecutará hasta que salga el contenedor.

Especifique un parámetro `attemptDurationSeconds`, que debe tener al menos de 60 segundos, en la definición del trabajo o al enviarlo. Cuando haya transcurrido este número de segundos después de la `startedAt` marca de tiempo del intento de trabajo, AWS Batch finaliza el trabajo. En el recurso de computación, el contenedor del trabajo recibe una señal `SIGTERM` para dar a la aplicación la posibilidad de que se apague correctamente. Si el contenedor se sigue ejecutando al cabo de 30 segundos, se envía una señal `SIGKILL` para forzar su cierre.

Las finalizaciones por haberse agotado el tiempo de espera se realizan en la medida que es posible. No espere que se produzcan exactamente en el momento en que se agota el tiempo de espera del intento de trabajo (pueden tardar algunos segundos más). Si su aplicación necesita una ejecución de tiempo de espera precisa, debe implementar esa lógica en la aplicación. Si tiene una gran cantidad de trabajos cuyo tiempo de espera se agota simultáneamente, las terminaciones por tiempo de espera se deben comportar como una cola del tipo "primero en entrar, primero en salir", donde los trabajos se terminen por lotes.

> **Note**
>
> No hay un valor de tiempo de espera máximo para un trabajo. AWS Batch

Si un trabajo se termina por superar la duración del tiempo de espera, no se vuelve a intentar. Si se produce un error en un intento de trabajo, se puede reintentar si se han habilitado los reintentos y la cuenta atrás del tiempo de espera comienza para el nuevo intento.

> **Important**
>
> Los trabajos que se ejecutan con recursos Fargate no pueden esperar funcionar durante más de 14 días. Si el tiempo de espera supera los 14 días, es posible que los recursos de Fargate ya no estén disponibles y el trabajo se cancelará.

En el caso de los trabajos de matriz, los trabajos secundarios tienen la misma configuración de tiempo de espera que el trabajo principal.

Para obtener información sobre el envío de un AWS Batch trabajo con una configuración de tiempo de espera, consulte. Enviar un trabajo

Trabajos de Amazon EKS

Un trabajo es la unidad de trabajo más pequeña de AWS Batch. Un AWS Batch trabajo en Amazon EKS implica un one-to-one mapeo a un Kubernetes pod. Una definición de AWS Batch trabajo es una plantilla para un AWS Batch trabajo. Cuando envía un AWS Batch trabajo, hace referencia a una definición de trabajo, selecciona una cola de trabajos y proporciona un nombre para el trabajo. En la definición de trabajo de un AWS Batch trabajo en Amazon EKS, el parámetro eksProperties define el conjunto de parámetros que admite un trabajo en AWS Batch Amazon EKS. En una SubmitJobsolicitud, el eksPropertiesOverrideparámetro permite anular algunos parámetros comunes. De esta forma, puede utilizar plantillas de definiciones de trabajos para varios trabajos. Cuando se envía un trabajo a su clúster de Amazon EKS, lo AWS Batch transforma en un `podspec` (`Kind: Pod`). `podspec`Utiliza algunos AWS Batch parámetros adicionales para garantizar que los trabajos se escalen y programen correctamente. AWS Batch combina etiquetas e imprecisiones para garantizar que los trabajos se ejecuten únicamente en los nodos AWS Batch gestionados y que otros módulos no se ejecuten en esos nodos.

Important

- Si el `hostNetwork` parámetro no está establecido de forma explícita en una definición de trabajo de Amazon EKS, el modo de red del pod AWS Batch pasa por defecto al modo host. Más específicamente, se aplican los siguientes ajustes: `hostNetwork=true` y `dnsPolicy=ClusterFirstWithHostNet`.
- AWS Batch limpia los módulos de trabajos poco después de que un pod complete su trabajo. Para ver los registros de las aplicaciones del pod, configure un servicio de registro para su clúster. Para obtener más información, consulte Utilice CloudWatch Logs para supervisar AWS Batch en los trabajos de Amazon EKS.

Asigne un trabajo en ejecución a un pod y un nodo

Los `podProperties` de un trabajo en ejecución tienen los parámetros `podName` y `nodeName` establecidos para el intento de trabajo actual. Utilice la operación DescribeJobsde la API para ver estos parámetros.

A continuación, se muestra un ejemplo del resultado.

```
$ aws batch describe-jobs --job 2d044787-c663-4ce6-a6fe-f2baf7e51b04
{
 "jobs": [
  {
   "status": "RUNNING",
   "jobArn": "arn:aws:batch:us-east-1:123456789012:job/2d044787-c663-4ce6-a6fe-
f2baf7e51b04",
   "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/
MyJobOnEks_SleepWithRequestsOnly:1",
   "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/My-Eks-JQ1",
   "jobId": "2d044787-c663-4ce6-a6fe-f2baf7e51b04",
   "eksProperties": {
    "podProperties": {
     "nodeName": "ip-192-168-55-175.ec2.internal",
     "containers": [
      {
       "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
       "resources": {
        "requests": {
         "cpu": "1",
```

```
          "memory": "1024Mi"
        }
       }
      }
     ],
     "podName": "aws-batch.b0aca953-ba8f-3791-83e2-ed13af39428c"
    }
   }
  }
 ]
}
```

En el caso de un trabajo con los reintentos habilitados, el `podName` y `nodeName` de cada intento completado aparece en el parámetro de `eksAttempts` lista de la operación de DescribeJobsAPI. El `podName` y el `nodeName` del intento en ejecución actual se encuentra en el objeto `podProperties`.

¿Cómo hacer que un pod en ejecución vuelva a su función

Un pod tiene etiquetas que indican la dirección `jobId` y el entorno `uuid` de cómputo al que pertenece. AWS Batch inyecta variables de entorno para que el tiempo de ejecución del trabajo pueda hacer referencia a la información del trabajo. Para obtener más información, consulte AWS Batch variables de entorno laboral. Puede ver esta información al ejecutar el siguiente comando. El resultado es el siguiente.

```
$ kubectl describe pod aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1 -n my-aws-batch-
namespace
Name:         aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1
Namespace:    my-aws-batch-namespace
Priority:     0
Node:         ip-192-168-45-88.ec2.internal/192.168.45.88
Start Time:   Wed, 26 Oct 2022 00:30:48 +0000
Labels:       batch.amazonaws.com/compute-environment-uuid=5c19160b-
d450-31c9-8454-86cf5b30548f
              batch.amazonaws.com/job-id=f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
              batch.amazonaws.com/node-uid=a4be5c1d-9881-4524-b967-587789094647
...
Status:       Running
IP:           192.168.45.88
IPs:
  IP:  192.168.45.88
Containers:
  default:
```

```
    Image:         public.ecr.aws/amazonlinux/amazonlinux:2
    ...
    Environment:
      AWS_BATCH_JOB_KUBERNETES_NODE_UID:  a4be5c1d-9881-4524-b967-587789094647
      AWS_BATCH_JOB_ID:                   f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
      AWS_BATCH_JQ_NAME:                  My-Eks-JQ1
      AWS_BATCH_JOB_ATTEMPT:              1
      AWS_BATCH_CE_NAME:                  My-Eks-CE1

...
```

Características compatibles con AWS Batch Amazon EKS Jobs

Estas son las características AWS Batch específicas que también son comunes a los Kubernetes trabajos que se ejecutan en Amazon EKS:

- Dependencias de trabajos
- Trabajos de matrices
- Tiempos de espera de trabajo
- Reintentos automáticos de trabajo
- Programación de participaciones justas

Kubernetes`Secrets` y `ServiceAccounts`

AWS Batch admite referencias Kubernetes `Secrets` y. `ServiceAccounts` Puede configurar los pods para utilizar los roles de IAM de Amazon EKS para las cuentas de servicio. Para obtener más información, consulte Configurar los pods para usar una cuenta de servicio de Kubernetes en la Guía del usuario de Amazon EKS.

Documentos relacionados

- Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS
- Para crear un trabajo basado en GPU en los recursos de Amazon EKS
- Trabajos bloqueados en estado `RUNNABLE`

Trabajos de matrices

Un trabajo de matriz es un trabajo que comparte parámetros comunes, como la definición de trabajo, las vCPU y la memoria. Se ejecuta como un conjunto de trabajos básicos relacionados, pero independientes, que se pueden distribuir en varios hosts y pueden ejecutarse de forma simultánea. Los trabajos de matrices son la manera más eficiente de ejecutar trabajos que ocurren todos en paralelo, como simulaciones Monte Carlo, barridos paramétricos o grandes trabajos de representación.

AWS Batch los trabajos de matriz se envían igual que los trabajos normales. Sin embargo, es necesario especificar un tamaño de matriz (entre 2 y 10 000) para definir la cantidad de trabajos secundarios que deberían ejecutarse en la matriz. Si envía un trabajo con un tamaño de matriz de 1 000, se ejecuta un solo trabajo que genera 1 000 trabajos secundarios. El trabajo de matriz es una referencia o un puntero para administrar todos los trabajos secundarios. Esto permite enviar grandes cargas de trabajo con una sola consulta. El tiempo de espera especificado en el parámetro `attemptDurationSeconds` se aplica a cada trabajo secundario. El trabajo de matriz principal no tiene tiempo de espera.

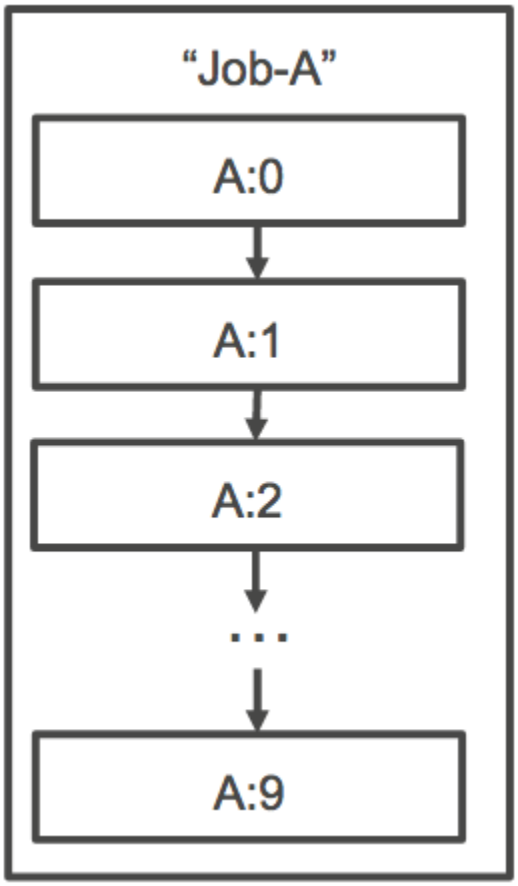
Al enviar un trabajo de matriz, el trabajo de matriz principal recibe un ID de AWS Batch trabajo normal. Cada trabajo secundario tiene el mismo ID base. Cada trabajo secundario tiene el mismo ID de base, pero su índice de matriz se añade al final del ID principal, por ejemplo, *`example_job_ID`*`:0` para el primer trabajo secundario de la matriz.

El trabajo de matriz principal puede introducir un estado `SUBMITTED`, `PENDING`, `FAILED` o `SUCCEEDED`. El trabajo de matriz principal se actualiza a `PENDING` cuando se actualiza cualquier trabajo secundario a `RUNNABLE`. Para obtener más información acerca de estas dependencias, consulte Dependencias de trabajos.

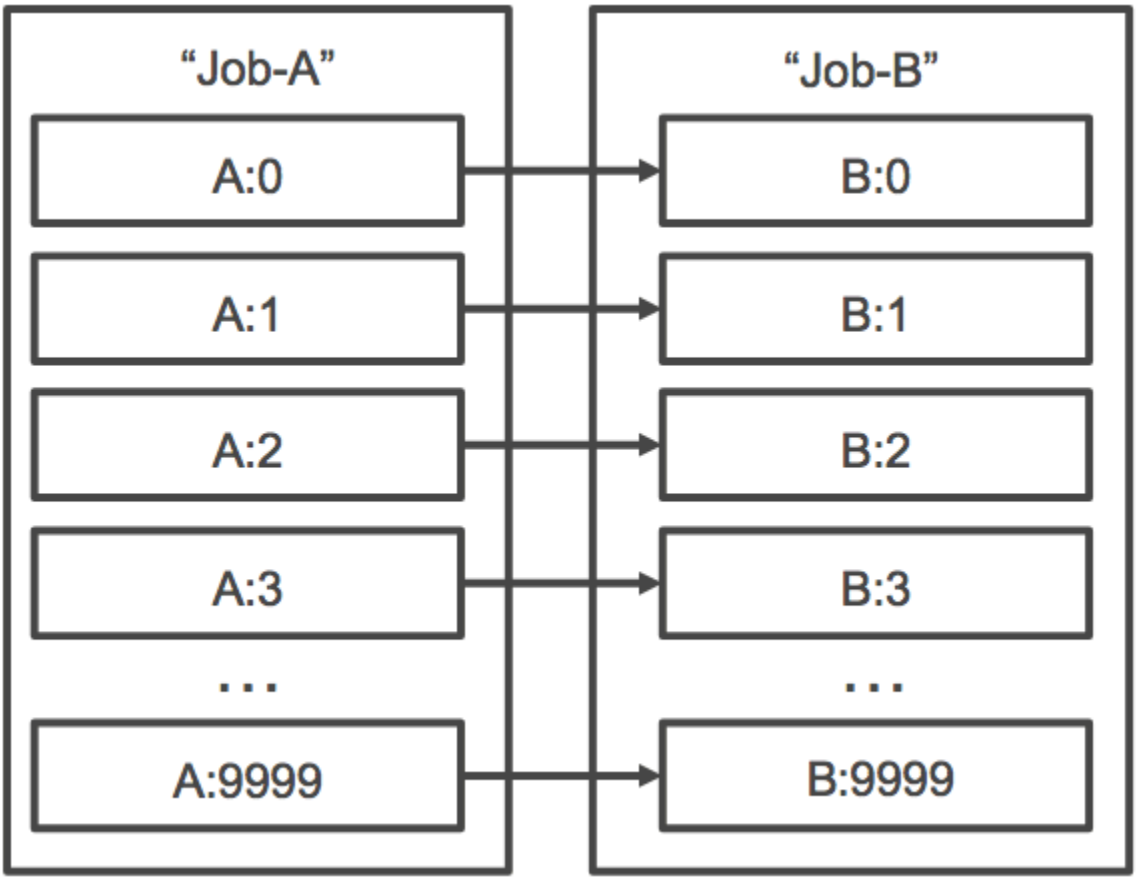
Durante el tiempo de ejecución, la variable de entorno `AWS_BATCH_JOB_ARRAY_INDEX` se establece en el número de índice del trabajo de matriz correspondiente del contenedor. Al primer índice de trabajo de matriz se le asigna el número `0`, y los trabajos secundarios posteriores se numeran en orden ascendente (por ejemplo, 1, 2, 3, etc.). Puede utilizar este valor de índice para controlar la forma en la que se diferencian los elementos secundarios del trabajo de matriz. Para obtener más información, consulte Tutorial: Uso del índice de trabajo de matriz para controlar la diferenciación de trabajos.

En las dependencias de trabajo de matriz, puede especificar un tipo para la dependencia, como, por ejemplo, `SEQUENTIAL` o `N_TO_N`. Puede especificar una dependencia de tipo `SEQUENTIAL`

(sin especificar un ID de trabajo) para que cada trabajo de matriz secundario se complete de forma secuencial, comenzando a partir del índice 0. Por ejemplo, si envía un trabajo de matriz con un tamaño de matriz de 100, y especifica una dependencia de tipo `SEQUENTIAL`, se generan 100 trabajos secundarios de forma secuencial, cada uno de los cuales debe completarse correctamente para que comience el siguiente. En la ilustración siguiente se muestra Job A, un trabajo de matriz con un tamaño de matriz de 10. Cada trabajo del índice secundario de Job A depende del trabajo secundario anterior. Job A:1 no puede comenzar hasta que termine Job A:0.



También puede especificar una dependencia de tipo `N_TO_N` con un ID de trabajo para los trabajos de matriz. De esta forma, cada índice secundario de este trabajo debe esperar a que se complete el índice secundario correspondiente de cada dependencia antes de comenzar. En la ilustración siguiente se muestran Job A y Job B, dos trabajos de matrices con un tamaño de matriz de 10 000. Cada trabajo del índice secundario de Job B depende del índice correspondiente de Job A. Job B:1 no puede comenzar hasta que termine Job A:1.



Si cancela o termina un trabajo de matriz principal, todos los trabajos secundarios se cancelan o terminan con él. Puede cancelar o terminar trabajos secundarios por separado (con lo que adoptarán el estado `FAILED`) sin afectar al resto de trabajos secundarios. Sin embargo, si un trabajo de matriz secundario produce un error (durante su ejecución o por una cancelación/terminación manual), el trabajo principal también producirá un error.

Ejemplo de flujo de trabajo de un trabajo de matriz

Un flujo de trabajo habitual para AWS Batch los clientes consiste en ejecutar un trabajo de configuración previo, ejecutar una serie de comandos en un gran número de tareas de entrada y, a continuación, concluir con un trabajo que agregue los resultados y escriba datos resumidos en Amazon S3, DynamoDB, Amazon Redshift o Aurora.

Por ejemplo:

- `JobA`: un trabajo estándar que no es de matriz, que realiza un listado rápido y una validación de metadatos de los objetos de un bucket `BucketA` de Amazon S3. La sintaxis de SubmitJobJSON es la siguiente.

```
{
    "jobName": "JobA",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobA-list-and-validate:1"
}
```

- `JobB`: un trabajo de matriz con 10 000 copias que depende de `JobA`, que ejecuta comandos con un uso intensivo de la CPU en cada objeto de `BucketA` y que carga los resultados en `BucketB`. La sintaxis de SubmitJobJSON es la siguiente.

```
{
    "jobName": "JobB",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobB-CPU-Intensive-Processing:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "4096"
            },
            {
                "type": "VCPU",
                "value": "32"
            }
        ]
   }
    "arrayProperties": {
        "size": 10000
    },
    "dependsOn": [
        {
            "jobId": "JobA_job_ID"
  }
    ]
}
```

- `JobC`: otro trabajo de matriz de 10 000 copias que depende de `JobB` con un modelo de dependencia `N_TO_N`, que ejecuta comandos con un uso intensivo de la memoria en cada elemento de `BucketB`, escribe los metadatos en DynamoDB y carga la salida obtenida en `BucketC` . La sintaxis de SubmitJobJSON es la siguiente.

```
{
    "jobName": "JobC",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobC-Memory-Intensive-Processing:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "32768"
            },
            {
                "type": "VCPU",
                "value": "1"
            }
        ]
   }
    "arrayProperties": {
        "size": 10000
    },
    "dependsOn": [
        {
            "jobId": "JobB_job_ID",
            "type": "N_TO_N"
        }
    ]
}
```

- JobD: un trabajo de matriz que realiza 10 pasos de validación que requieren consultar DynamoDB y que pueden interactuar con cualquiera de los buckets de Amazon S3 anteriores. Cada uno de los pasos de JobD ejecuta el mismo comando. Sin embargo, el comportamiento es diferente en función del valor de la variable de entorno AWS_BATCH_JOB_ARRAY_INDEX del contenedor del trabajo. Estos pasos de validación se ejecutan de forma secuencial (por ejemplo JobD:0 y, a continuación, JobD:1). La sintaxis de SubmitJobJSON es la siguiente.

```
{
    "jobName": "JobD",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobD-Sequential-Validation:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
```

```
                "type": "MEMORY",
                "value": "32768"
            },
            {
                "type": "VCPU",
                "value": "1"
            }
        ]
   }
    "arrayProperties": {
        "size": 10
    },
    "dependsOn": [
        {
            "jobId": "JobC_job_ID"
        },
        {
            "type": "SEQUENTIAL"
        },

    ]
}
```

- JobE: un trabajo final que no es de matriz, que realiza varias operaciones de limpieza sencillas, y que envía una notificación de Amazon SNS con un mensaje que indica que la canalización se ha completado y que incluye un enlace con la URL de salida. La sintaxis de SubmitJobJSON es la siguiente.

```
{
    "jobName": "JobE",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobE-Cleanup-and-Notification:1",
    "parameters": {
        "SourceBucket": "s3://JobD-Output-Bucket",
        "Recipient": "pipeline-notifications@mycompany.com"
    },
    "dependsOn": [
        {
            "jobId": "JobD_job_ID"
        }
    ]
}
```

Tutorial: Uso del índice de trabajo de matriz para controlar la diferenciación de trabajos

En este tutorial se describe cómo utilizar la variable de entorno de `AWS_BATCH_JOB_ARRAY_INDEX` para diferenciar los trabajos secundarios. Cada trabajo secundario se asigna a esta variable. El ejemplo utiliza el número de índice del trabajo secundario para leer una línea específica de un archivo. A continuación, sustituye el parámetro asociado a ese número de línea por un comando incluido en el contenedor del trabajo. El resultado es que puede tener varios AWS Batch trabajos que ejecuten la misma imagen de Docker y los mismos argumentos de comando. Sin embargo, los resultados son diferentes porque el índice de trabajos de la matriz se usa como modificador.

En este tutorial, creará un archivo de texto que tiene todos los colores del arco iris, cada uno en su propia línea. A continuación, creará un script de punto de entrada para un contenedor de Docker que convierte el índice en un valor que se puede utilizar como número de línea en el archivo de colores. El índice comienza en cero, pero los números de línea comienzan en uno. Creará un Dockerfile que copia los archivos de colores y de índice en la imagen del contenedor y establece el valor `ENTRYPOINT` de la imagen en el script de punto de entrada. El Dockerfile y los recursos se compilarán en una imagen de Docker que se enviará a Amazon ECR. A continuación, registras una definición de trabajo que usa tu nueva imagen de contenedor, envías un trabajo de AWS Batch matriz con esa definición de trabajo y ves los resultados.

Requisitos previos

Este tutorial tiene los requisitos previos siguientes:

- Un entorno AWS Batch informático. Para obtener más información, consulte Cómo crear un entorno de computación.
- Una cola de AWS Batch trabajos y un entorno informático asociado. Para obtener más información, consulte Cómo crear de una cola de trabajos.
- Están AWS CLI instalados en su sistema local. Para obtener más información, consulte Instalar la AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface .
- Docker debe estar instalado en el sistema local. Para obtener más información, consulte About Docker CE en la documentación de Docker.

Paso 1: Crear una imagen de contenedor

Puede utilizar el `AWS_BATCH_JOB_ARRAY_INDEX` en una definición de trabajo en el parámetro de comando. Sin embargo, se recomienda crear una imagen contenedor que utilice la variable en un script de punto de entrada en su lugar. En esta sección, se describe cómo crear dicha imagen de contenedor.

Para compilar la imagen de contenedor de Docker

1. Cree un directorio nuevo para utilizarlo como espacio de trabajo de la imagen de Docker y desplácese a él.
2. Cree un archivo denominado `colors.txt` en el directorio del espacio de trabajo y pegue en él el contenido siguiente.

```
red
orange
yellow
green
blue
indigo
violet
```

3. Cree un archivo denominado `print-color.sh` en el directorio del espacio de trabajo y pegue en él el contenido siguiente.

> **Note**
>
> La variable `LINE` se establece en `AWS_BATCH_JOB_ARRAY_INDEX` + 1 debido a que el índice de matriz empieza en 0, pero los números de línea empiezan en 1. La variable `COLOR` se establece en el color de `colors.txt` que está asociado a su número de línea.

```
#!/bin/sh
LINE=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
COLOR=$(sed -n ${LINE}p /tmp/colors.txt)
echo My favorite color of the rainbow is $COLOR.
```

4. Cree un archivo denominado `Dockerfile` en el directorio del espacio de trabajo y pegue en él el contenido siguiente. Este Dockerfile copia los archivos anteriores en el contenedor y configura el script de punto de entrada para que se ejecute al iniciarse el contenedor.

```
FROM busybox
COPY print-color.sh /tmp/print-color.sh
COPY colors.txt /tmp/colors.txt
RUN chmod +x /tmp/print-color.sh
ENTRYPOINT /tmp/print-color.sh
```

5. Compile la imagen de Docker.

```
$ docker build -t print-color .
```

6. Pruebe el contenedor con el script siguiente. Este script establece la variable `AWS_BATCH_JOB_ARRAY_INDEX` en 0 localmente y, a continuación, incrementa su valor para simular lo que haría un trabajo de matriz con siete elementos secundarios.

```
$ AWS_BATCH_JOB_ARRAY_INDEX=0
while [ $AWS_BATCH_JOB_ARRAY_INDEX -le 6 ]
do
    docker run -e AWS_BATCH_JOB_ARRAY_INDEX=$AWS_BATCH_JOB_ARRAY_INDEX print-color
    AWS_BATCH_JOB_ARRAY_INDEX=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
done
```

Se genera la siguiente salida.

```
My favorite color of the rainbow is red.
My favorite color of the rainbow is orange.
My favorite color of the rainbow is yellow.
My favorite color of the rainbow is green.
My favorite color of the rainbow is blue.
My favorite color of the rainbow is indigo.
My favorite color of the rainbow is violet.
```

Paso 2: Insertar una imagen en Amazon ECR

Ahora que ha compilado y probado el contenedor de Docker, debe enviarlo a un repositorio de imágenes. En este ejemplo se utiliza Amazon ECR, pero puede utilizar otro registro, como DockerHub.

1. Cree un repositorio de imágenes de Amazon ECR para almacenar la imagen de contenedor. En este ejemplo solo se utiliza el AWS CLI, pero también se puede utilizar el AWS Management Console. Para obtener más información, consulte Creación de un repositorio en la Guía del usuario de Amazon Elastic Container Registry.

   ```
   $ aws ecr create-repository --repository-name print-color
   ```

2. Etiquete la imagen `print-color` con el URI del repositorio de Amazon ECR que se obtuvo del paso anterior.

   ```
   $ docker tag print-color aws_account_id.dkr.ecr.region.amazonaws.com/print-color
   ```

3. Inicie sesión en su registro Amazon ECR. Para obtener más información, consulte Autenticación de registros en la Guía del usuario de Amazon Elastic Container Registry.

   ```
   $ aws ecr get-login-password \
       --region region | docker login \
       --username AWS \
       --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
   ```

4. Envíe la imagen a Amazon ECR.

   ```
   $ docker push aws_account_id.dkr.ecr.region.amazonaws.com/print-color
   ```

Paso 3: Crear y registrar una definición de trabajo

Ahora que su imagen de Docker está en un registro de imágenes, puede especificarla en una definición de AWS Batch trabajo. Posteriormente, puede utilizarlo para ejecutar un trabajo de matriz. En este ejemplo se utiliza la API de AWS CLI. Sin embargo, también puede utilizar la AWS Management Console. Para obtener más información, consulte Creación de una definición de trabajo de un solo nodo .

Para crear una definición de trabajo

1. Cree un archivo denominado `print-color-job-def.json` en el directorio del espacio de trabajo y pegue en él el contenido siguiente. Reemplace el URI del repositorio de imágenes por el URI de su propia imagen.

   ```
   {
   ```

```
  "jobDefinitionName": "print-color",
  "type": "container",
  "containerProperties": {
    "image": "aws_account_id.dkr.ecr.region.amazonaws.com/print-color",
    "resourceRequirements": [
        {
            "type": "MEMORY",
            "value": "250"
        },
        {
            "type": "VCPU",
            "value": "1"
        }
    ]
  }
}
```

2. Registre la definición del trabajo con AWS Batch.

```
$ aws batch register-job-definition --cli-input-json file://print-color-job-def.json
```

Paso 4: Envíe un trabajo AWS Batch de matriz

Después de registrar la definición de su trabajo, puede enviar un trabajo de AWS Batch matriz que utilice su nueva imagen de contenedor.

Para enviar un trabajo AWS Batch de matriz

1. Cree un archivo denominado `print-color-job.json` en el directorio del espacio de trabajo y pegue en él el contenido siguiente.

> **Note**
>
> En este ejemplo, se utiliza la cola de trabajos mencionada en la sección the section called “Requisitos previos”.

```
{
  "jobName": "print-color",
```

```
  "jobQueue": "existing-job-queue",
  "arrayProperties": {
    "size": 7
  },
  "jobDefinition": "print-color"
}
```

2. Envíe el trabajo a su lista de AWS Batch trabajos. Anote el ID de trabajo que se devuelve como resultado.

```
$ aws batch submit-job --cli-input-json file://print-color-job.json
```

3. Describa el estado del trabajo y espere a que el trabajo adopte el valor `SUCCEEDED`.

Paso 5: Ver los registros del trabajo de matriz

Cuando su trabajo alcance el `SUCCEEDED` estado, podrá ver los CloudWatch registros desde el contenedor del trabajo.

Para ver los registros de su trabajo en CloudWatch Registros

1. Abre la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En el panel de navegación izquierdo, elija Jobs (Trabajos).
3. En Job queue (Cola de trabajos), seleccione una cola.
4. En la sección Status (Estado), elija succeeded (realizado correctamente).
5. Para mostrar todos los trabajos secundarios del trabajo de matriz, seleccione el ID de trabajo que se ha obtenido en la sección anterior.
6. Para ver los logs del contenedor del trabajo, seleccione uno de los trabajos secundarios y elija View logs (Ver logs).

7. Vea los logs del otro trabajo secundario. Cada trabajo devuelve un color diferente del arco iris.

Trabajos paralelos de varios nodos

Los trabajos paralelos de varios nodos le permiten ejecutar trabajos individuales que abarcan varias instancias de Amazon EC2. Los trabajos paralelos de varios nodos de AWS Batch le permiten ejecutar aplicaciones informáticas de alto rendimiento a gran escala estrechamente acopladas y entrenar modelos de GPU distribuida sin necesidad de lanzar, configurar y administrar los recursos de Amazon EC2 directamente. Un trabajo paralelo de AWS Batch varios nodos es compatible con cualquier marco que admita la comunicación entre nodos basada en IP. Algunos ejemplos son Apache MXNet TensorFlow, Caffe2 o Message Passing Interface (MPI).

Los trabajos paralelos de varios nodos se envían como un único trabajo. Sin embargo, la definición de trabajo (o las anulaciones de nodos de envío de trabajos) especifica el número de nodos que se deben crear para el trabajo y qué grupos de nodos deben crearse. Cada trabajo paralelo de varios nodos contiene un nodo principal, que se lanza en primer lugar. Una vez que el nodo principal está en marcha, se lanzan e inician los nodos secundarios. El trabajo finaliza solo si sale el nodo principal. A continuación, se detienen todos los nodos secundarios. Para obtener más información, consulte Grupos de nodos.

Los nodos de trabajo paralelos de varios nodos son de un solo inquilino. Esto significa que solo se ejecuta un contenedor de trabajos en cada instancia de Amazon EC2.

El estado final del trabajo (`SUCCEEDED` o `FAILED`) lo determina el estado final del trabajo del nodo principal. Para obtener el estado de un trabajo paralelo de varios nodos, puede describir el trabajo utilizando el ID de trabajo obtenido al enviar el trabajo. Si necesita los detalles de los nodos secundarios, deberá describir cada nodo secundario por separado. Puede direccionar los nodos mediante la notación #*N* (empezando por 0). Por ejemplo, para acceder a los detalles del segundo nodo de un trabajo, describa *aws_batch_job_id* #1 mediante la operación de API. AWS Batch DescribeJobs La información `started`, `stoppedAt`, `statusReason` y `exit` de un trabajo paralelo de varios nodos se rellena desde el nodo principal.

Si especifica los reintentos de trabajo, un error en el nodo principal provocará otro intento. Los errores en los nodos secundarios no provocan que se produzcan más intentos. Cada nuevo intento de un trabajo paralelo de varios nodos actualiza el intento correspondiente de sus nodos secundarios asociados.

Para ejecutar trabajos paralelos de varios nodos AWS Batch, el código de la aplicación debe contener los marcos y las bibliotecas necesarios para la comunicación distribuida.

Variables de entorno

En tiempo de ejecución, cada nodo tiene configuradas las variables de entorno estándar que reciben todos los AWS Batch trabajos. Además, los nodos se configuran con las siguientes variables de entorno que son específicas para los trabajos paralelos de varios nodos:

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Esta variable se establece en el número de índice del nodo principal del trabajo. El código de la aplicación puede comparar `AWS_BATCH_JOB_MAIN_NODE_INDEX` con `AWS_BATCH_JOB_NODE_INDEX` en un nodo individual para determinar si es el nodo principal.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable no está presente en el nodo principal. Esta variable se establece en la dirección IPv4 privada del nodo principal del trabajo. El código de la aplicación del nodo secundario puede utilizar esta dirección para comunicarse con el nodo principal.

`AWS_BATCH_JOB_NODE_INDEX`

Esta variable se establece en el número de índice del nodo. El índice del nodo comienza a partir de 0 y cada nodo recibe un número de índice único. Por ejemplo, un trabajo paralelo de varios nodos con 10 elementos secundarios tiene valores de índice comprendidos entre 0 y 9.

`AWS_BATCH_JOB_NUM_NODES`

Esta variable se establece en el número de nodos solicitados para el trabajo en paralelo de varios nodos.

Grupos de nodos

Un grupo de nodos es un conjunto de nodos de trabajo idénticos que comparten las mismas propiedades de contenedor. Se puede utilizar AWS Batch para especificar hasta cinco grupos de nodos distintos para cada trabajo.

Cada grupo puede tener sus propias imágenes de contenedor, comandos, variables de entorno, etc. Por ejemplo, puede enviar un trabajo que requiera una sola instancia `c5.xlarge` para el nodo principal y cinco nodos secundarios de la instancia `c5.xlarge`. Cada uno de estos grupos de nodos

distintos puede especificar diferentes imágenes de contenedor o comandos para ejecutarlos en cada trabajo.

Como alternativa, todos los nodos de su trabajo pueden usar un único grupo de nodos. Además, el código de su aplicación puede diferenciar las funciones de los nodos, como el nodo principal y el nodo secundario. Para ello, compara la variable de entorno `AWS_BATCH_JOB_MAIN_NODE_INDEX` con su propio valor para `AWS_BATCH_JOB_NODE_INDEX`. Puede tener un máximo de 1000 nodos en un solo trabajo. Este es el límite predeterminado de las instancias de un clúster de Amazon ECS. Puede solicitar un aumento de este límite.

> **Note**
>
> En la actualidad, todos los grupos de nodos de un trabajo paralelo de varios nodos deben utilizar el mismo tipo de instancia.

Ciclo de vida del trabajo

Al enviar un trabajo paralelo de varios nodos, el trabajo entra en el estado `SUBMITTED`. A continuación, el trabajo espera a que finalicen todas las dependencias del trabajo. El trabajo también pasa al estado `RUNNABLE`. Por último, AWS Batch aprovisiona la capacidad de instancia necesaria para ejecutar el trabajo y lanza estas instancias.

Cada trabajo paralelo de varios nodos contiene un nodo principal. El nodo principal es una subtarea única que AWS Batch supervisa para determinar el resultado del trabajo de varios nodos enviado. El nodo principal se lanza en primer lugar y pasa a tener el estado `STARTING`. El valor de tiempo de espera especificado en el parámetro `attemptDurationSeconds` se aplica a todo el trabajo y no a los nodos.

Cuando el nodo principal alcanza el estado `RUNNING` (después de que el contenedor del nodo se esté ejecutando), los nodos secundarios se lanzan y pasan al estado `STARTING`. Los nodos secundarios aparecen en orden aleatorio. No hay ninguna garantía sobre la sincronización o el orden del lanzamiento de los nodos secundarios. Para asegurarse de que todos los nodos de los trabajos están en el estado `RUNNING` después de que el contenedor del nodo se está ejecutando, el código de la aplicación puede consultar la API de AWS Batch para obtener información sobre el nodo principal y los nodos secundarios. Como alternativa, el código de la aplicación puede esperar hasta que todos los nodos estén en línea antes de iniciar cualquier tarea de procesamiento distribuido. La dirección IP privada del nodo principal está disponible en la variable de entorno `AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS` de cada nodo secundario. El código de

la aplicación puede utilizar esta información para coordinar y comunicar datos entre cada una de las tareas.

A medida que finalizan los nodos individuales, pasan al estado `SUCCEEDED` o `FAILED`, en función de su código de salida. Si finaliza el nodo principal, se considera que el trabajo ha finalizado y todos los nodos secundarios se detienen. Si un nodo secundario muere, AWS Batch no realiza ninguna acción en los demás nodos del trabajo. Si no desea que el trabajo continúe con una cantidad reducida de nodos, debe tenerlo en cuenta en el código de la aplicación. De este modo, se termina o se cancela el trabajo.

Consideraciones del entorno de computación

Hay varios aspectos que es preciso tener en cuenta al configurar entornos de computación para ejecutar trabajos paralelos de varios nodos con AWS Batch.

- Los trabajos paralelos de varios nodos no se admiten en entornos de computación `UNMANAGED`.
- Si va a enviar trabajos paralelos de varios nodos a un entorno de computación, considere la posibilidad de crear un grupo con ubicación en clúster en una única zona de disponibilidad y asociarlo a los recursos de computación. Esto mantiene los trabajos paralelos de varios nodos en una agrupación lógica de instancias muy próxima al alto potencial de flujo de la red. Para obtener más información, consulte Grupos de ubicación en la Guía del usuario de instancias de Linux de Amazon EC2.
- Los trabajos paralelos de varios nodos no se admiten en entornos de computación que utilicen instancias de spot.
- AWS Batch los trabajos paralelos de varios nodos utilizan el modo de `awsvpc` red Amazon ECS, que proporciona a los contenedores de trabajos paralelos de varios nodos las mismas propiedades de red que las instancias de Amazon EC2. Cada contenedor de trabajos paralelos de varios nodos obtiene su propia interfaz de red elástica, una dirección IP privada principal y un nombre de host DNS interno. La interfaz de red se crea en la misma subred de VPC como su recurso de computación de host. Los grupos de seguridad que se hayan aplicado a los recursos de computación se aplicarán también a ella. Para obtener más información, consulte Integración en red de las tareas con el modo de red awsvpc en la Guía para desarrolladores de Amazon Elastic Container Service.
- El entorno de computación no puede tener asociados más de cinco grupos de seguridad.
- El modo de red `awsvpc` no proporciona las interfaces de red elásticas para los trabajos paralelos de varios nodos con direcciones IP públicas. Para obtener acceso a Internet, los recursos de computación deben lanzarse en una subred privada configurada para utilizar una puerta de enlace

NAT. Para obtener más información, consulte Gateways NAT en la Guía del usuario de Amazon VPC. La comunicación entre nodos debe utilizar la dirección IP privada o el nombre de host DNS para el nodo. Los trabajos paralelos de varios nodos que se ejecutan en recursos de computación dentro de subredes públicas no disponen de acceso de red saliente. Para crear una VPC con subredes privadas y una gateway NAT, consulte Creación de una nube virtual privada (VPC) .

- Las interfaces de red elásticas que se crean y asocian a los recursos de computación no se pueden separar manualmente ni la cuenta puede modificarlas. De este modo, se evita la eliminación accidental de una interfaz de red elástica que esté asociada a un trabajo en ejecución. Para liberar las interfaces de red elásticas de una tarea, termine el trabajo.
- El entorno de computación debe tener un número máximo de CPU virtuales suficiente para admitir el trabajo paralelo de varios nodos.
- Su cuota de instancias de Amazon EC2 incluye la cantidad de instancias necesarias para ejecutar su trabajo. Por ejemplo, si el trabajo requiere 30 instancias, pero la cuenta solo puede ejecutar 20 en una región. Entonces, su trabajo se quedará estancado en el estado `RUNNABLE`.
- Si especifica un tipo de instancia para un grupo de nodos en un trabajo paralelo de varios nodos, el entorno de computación debe ser capaz de lanzar ese tipo de instancia.

Trabajos de GPU

Los trabajos de GPU le permiten ejecutar los trabajos que utilizan las GPU de una instancia.

Se admiten los siguientes tipos de instancias de Amazon EC2 basadas en GPU. Para obtener más información, consulte las Instancias G3 de Amazon EC2, Instancias G4 de Amazon EC2, Instancias G5 de Amazon EC2, Instancias P2 de Amazon EC2, Instancias P3 de Amazon EC2, Instancias P4d de Amazon EC2 e Instancias P5 de Amazon EC2.

Tipo de instancia	GPU	Memoria de GPU	vCPU	Memoria	Ancho de banda de red
g3s.xlarge	1	8 GiB	4	30,5 GiB	10 Gbps
g3.4xlarge	1	8 GiB	16	122 GiB	Hasta 10 Gbps
g3.8xlarge	2	16 GiB	32	244 GiB	10 Gbps
g3.16xlarge	4	32 GiB	64	488 GiB	25 Gbps

Tipo de instancia	GPU	Memoria de GPU	vCPU	Memoria	Ancho de banda de red
g4dn.xlarge	1	16 GiB	4	16 GiB	Hasta 25 Gbps.
g4dn.2xlarge	1	16 GiB	8	32 GiB	Hasta 25 Gbps.
g4dn.4xlarge	1	16 GiB	16	64 GiB	Hasta 25 Gbps.
g4dn.8xlarge	1	16 GiB	32	128 GiB	50 Gbps
g4dn.12xlarge	4	64 GiB	48	192 GiB	50 Gbps
g4dn.16xlarge	1	16 GiB	64	256 GiB	50 Gbps
g5.xlarge	1	24 GiB	4	16 GiB	Hasta 10 Gbps
g5.2xlarge	1	24 GiB	8	32 GiB	Hasta 10 Gbps
g5.4xlarge	1	24 GiB	16	64 GiB	Hasta 25 Gbps.
g5.8xlarge	1	24 GiB	32	128 GiB	25 Gbps
g5.16xlarge	1	24 GiB	64	256 GiB	25 Gbps
g5.12xlarge	4	96 GiB	48	192 GiB	40 Gbps
g5.24xlarge	4	96 GiB	96	384 GiB	50 Gbps
g5.48xlarge	8	192 GiB	192	768 GiB	100 Gbps
p2.xlarge	1	12 GiB	4	61 GiB	Alta
p2.8xlarge	8	96 GiB	32	488 GiB	10 Gbps
p2.16xlarge	16	192 GiB	64	732 GiB	20 Gbps
p3.2xlarge	1	16 GiB	8	61 GiB	Hasta 10 Gbps
p3.8xlarge	4	64 GiB	32	244 GiB	10 Gbps
p3.16xlarge	8	128 GiB	64	488 GiB	25 Gbps

Tipo de instancia	GPU	Memoria de GPU	vCPU	Memoria	Ancho de banda de red
p3dn.24xlarge	8	256 GiB	96	768 GiB	100 Gbps
p4d.24xlarge	8	320 GiB	96	1152 GiB	4x100 Gbps
p5.48xlarge	8	640 GiB	192	2 TiB	32x100 Gbps

> **Note**
>
> Solo los tipos de instancias que admiten una GPU NVIDIA y utilizan una arquitectura x86_64 se admiten para los trabajos de GPU en AWS Batch. Por ejemplo, las familias de instancias de `G4ad` y `G5g` no son compatibles.

El parámetro resourceRequirements de la definición del trabajo especifica el número de GPU que se va a anclar al contenedor. Esta cantidad de GPU no está disponible para ningún otro trabajo que se ejecute en esa instancia mientras dure ese trabajo. Todos los tipos de instancia de un entorno de computación que ejecutarán los trabajos de GPU deben pertenecer a las familias de instancias de `p2`, `p3`, `p4`, `p5`, `g3`, `g3s`, `g4` o `g5`. Si no lo hace así, un trabajo de GPU podría quedar bloqueado en el estado `RUNNABLE`.

Los trabajos que no utilizan las GPU se pueden ejecutar en instancias de GPU. Sin embargo, es posible que su ejecución en las instancias de GPU cueste más que en instancias similares que no sean de GPU. En función de la vCPU específica, la memoria y el tiempo necesario, estos trabajos sin GPU pueden bloquear la ejecución de los trabajos de GPU.

Para crear un trabajo basado en GPU en los recursos de Amazon EKS

En esta sección se explica cómo ejecutar una carga de trabajo de una GPU de Amazon EKS en AWS Batch.

Contenido

- Para crear una definición de trabajo de GPU de Amazon EKS
- Para ejecutar un trabajo de GPU en su clúster de Amazon EKS

Para crear un clúster de Kubernetes basado en GPU en Amazon EKS

Antes de crear un clúster de Kubernetes basado en GPU en Amazon EKS, debe haber completado los pasos que se indican en Cómo empezar con AWS Batch Amazon EKS. Además, tenga en cuenta lo siguiente:

- AWS Batch admite tipos de instancias con GPU NVIDIA.
- De forma predeterminada, AWS Batch selecciona la AMI acelerada de Amazon EKS con la Kubernetes versión que coincide con la versión del plano de control del clúster de Amazon EKS.

```
$ cat <<EOF > ./batch-eks-gpu-ce.json
{
  "computeEnvironmentName": "My-Eks-GPU-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:<account>:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 1024,
    "instanceTypes": [
      "p3dn.24xlarge",
      "p4d.24xlarge"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
```

```
EOF

$ aws batch create-compute-environment --cli-input-json file://./batch-eks-gpu-ce.json
```

AWS Batch no administra el complemento del dispositivo NVIDIA GPU en su nombre. Debe instalar este complemento en su clúster de Amazon EKS y permitir que se dirija a los AWS Batch nodos. Para obtener más información, consulte Habilitar GPU Support en Kubernetes on GitHub.

Para configurar el complemento del NVIDIA dispositivo (`DaemonSet`) para que se dirija a los AWS Batch nodos, ejecute los siguientes comandos.

```
# pull nvidia daemonset spec
$ curl -O https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.12.2/nvidia-
device-plugin.yml
# using your favorite editor, add Batch node toleration
# this will allow the DaemonSet to run on Batch nodes
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"

$ kubectl apply -f nvidia-device-plugin.yml
```

No se recomienda mezclar cargas de trabajo informáticas (CPU y memoria) con cargas de trabajo basadas en GPU en las mismas combinaciones de entorno de computación y cola de tareas. Esto se debe a que las tareas informáticas pueden consumir la capacidad de la GPU.

Para adjuntar colas de trabajos, ejecute los siguientes comandos.

```
$ cat <<EOF > ./batch-eks-gpu-jq.json
 {
    "jobQueueName": "My-Eks-GPU-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-GPU-CE1"
      }
    ]
  }
EOF
```

```
$ aws batch create-job-queue --cli-input-json file://./batch-eks-gpu-jq.json
```

Para crear una definición de trabajo de GPU de Amazon EKS

Solo `nvidia.com/gpu` se admite en este momento y el valor del recurso que establezca debe ser un número entero. No puede usar fracciones de GPU. Para obtener más información, consulte Programación de GPUs en la Kubernetes documentación.

Para registrar una definición de trabajo de GPU para Amazon EKS, ejecute los siguientes comandos.

```
$ cat <<EOF > ./batch-eks-gpu-jd.json
{
    "jobDefinitionName": "MyGPUJobOnEks_Smi",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "hostNetwork": true,
            "containers": [
                {
                    "image": "nvcr.io/nvidia/cuda:10.2-runtime-centos7",
                    "command": ["nvidia-smi"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi",
                            "nvidia.com/gpu": "1"
                        }
                    }
                }
            ]
        }
    }
}
EOF

$ aws batch register-job-definition --cli-input-json file://./batch-eks-gpu-jd.json
```

Para ejecutar un trabajo de GPU en su clúster de Amazon EKS

El recurso de la GPU no se puede comprimir. AWS Batch crea una especificación de módulo para los trabajos de GPU en la que el valor de la solicitud es igual al valor de los límites. Se trata de un requisito de Kubernetes.

Para reiniciar un trabajo de GPU, ejecute los siguientes comandos.

```
$ aws batch submit-job --job-queue My-Eks-GPU-JQ1 --job-definition MyGPUJobOnEks_Smi --
job-name My-Eks-GPU-Job

# locate information that can help debug or find logs (if using Amazon CloudWatch Logs
 with Fluent Bit)
$ aws batch describe-jobs --job <job-id> | jq '.jobs[].eksProperties.podProperties |
 {podName, nodeName}'
{
  "podName": "aws-batch.f3d697c4-3bb5-3955-aa6c-977fcf1cb0ca",
  "nodeName": "ip-192-168-59-101.ec2.internal"
}
```

Busca y filtra trabajos AWS Batch

Puede enumerar los trabajos de una cola de trabajos mediante la AWS Batch consola. Sin embargo, si hay muchos trabajos en la cola de trabajos, puede resultar difícil encontrar uno específico.

Puede utilizar Búsqueda y filtrado para enumerar los trabajos que coinciden con los criterios de búsqueda que especifique.

1. Abra la consola de AWS CloudFormation.
2. Elija Jobs (Trabajos).
3. Active la búsqueda y el filtrado.

> **Note**
>
> Si tiene varios trabajos, este proceso puede tardar unos minutos.

4. En el cuadro Seleccione una cola de trabajos, seleccione la cola de trabajos que desee buscar.
5. En el cuadro Filtrar recursos por propiedad o valor, elija una de las propiedades de la lista.
6. Elija el operador que desee usar. Por ejemplo, elija Estado =.

> **Tip**
>
> Para usar una propiedad o un operador diferentes, cierre los criterios actuales. A continuación, elija la propiedad y el operador que desee.

7. Introduzca o elija un valor de propiedad. Por ejemplo, introduzca la totalidad o parte del nombre de un trabajo o elija Estado = EJECUTABLE.
8. Elija el trabajo que quiera en la lista filtrada.

> **Tip**
>
> Si no ve el trabajo que desea, desplácese por la lista filtrada.

Registros de trabajo

Puede configurar sus AWS Batch trabajos para enviar información de registro a CloudWatch Logs. De esta manera, puede ver diferentes registros de sus trabajos en una ubicación práctica. Para obtener más información, consulte Uso de CloudWatch Logs con AWS Batch.

También puede usar los registros de tareas de la AWS Batch consola para supervisar o solucionar problemas de una AWS Batch tarea.

1. Abra la consola de AWS CloudFormation.
2. Elija Jobs (Trabajos).
3. En Cola de trabajos, elija la cola de trabajos que desee.

> **Tip**
>
> Si hay varios trabajos en la cola de trabajos, puede activar la Búsqueda y filtrado para encontrar un trabajo más rápido. Para obtener más información, consulte Busca y filtra trabajos AWS Batch.

4. En Estado, elija el estado del trabajo que desee.
5. Elija el trabajo que desee.
6. En la página Detalles, desplázate hacia abajo hasta Registros de trabajos.
7. Seleccione Recuperar registros.
8. Si se requiere autorización, introduce y**OK**, a continuación, selecciona Autorizar para aceptar CloudWatch los cargos de Amazon.

> **Note**
>
> Para revocar tu autorización de CloudWatch cobro, sigue estos pasos:
>
> 1. En el panel de navegación de la izquierda, elija Permisos.
> 2. Para Registros de trabajos, seleccione Editar.
> 3. Desactive la CloudWatch casilla Autorizar el uso de Batch.
> 4. Elija Guardar cambios.

9. Revise los datos de registro del AWS Batch trabajo.

> **Tip**
>
> Puede filtrar el registro en función de Palabras clave, Resultados máximos y Clasificación. También puede elegir uno de los intervalos de tiempo predeterminados o crear un intervalo personalizado para personalizar los resultados.

Información de trabajo

Puede revisar la información del trabajo AWS Batch , como el estado, la definición del trabajo y la información del contenedor.

1. Abra la consola de AWS CloudFormation.
2. Elija Jobs (Trabajos).
3. En Cola de trabajos, elija la cola de trabajos que desee.

> **Tip**
>
> Si hay varios trabajos en la cola de trabajos, puede activar la Búsqueda y filtrado para encontrar un trabajo más rápido. Para obtener más información, consulte Busca y filtra trabajos AWS Batch.

4. Elija el trabajo que desee.

> **Note**
>
> También puede usar AWS Command Line Interface (AWS CLI) para ver los detalles de un AWS Batch trabajo. Para obtener más información, consulte describe-jobs en la AWS CLI Referencia de comandos.

Definiciones de trabajo

AWS Batch las definiciones de trabajos especifican cómo se van a ejecutar los trabajos. Si bien cada trabajo debe hacer referencia a una definición de trabajo, muchos de los parámetros especificados en dicha definición pueden ser ignorados en tiempo de ejecución.

Contenido

Algunos de los atributos especificados en una definición de trabajo son:

- La imagen de Docker a utilizar con el contenedor en el trabajo
- La cantidad de vCPU y de memoria a utilizar con el contenedor
- El comando que el contenedor debe ejecutar al iniciarse
- Las variables de entorno que se deben transmitirse al contenedor cuando se inicia, de haberlas
- Los volúmenes de datos que deben utilizarse con el contenedor
- ¿Qué función de IAM (si la hay) debe utilizar su trabajo para obtener permisos AWS

Para obtener una descripción completa de los parámetros disponibles en una definición de trabajo, consulte Parámetros de definición de trabajo para ContainerProperties.

Creación de una definición de trabajo de un solo nodo

Antes de ejecutar trabajos en AWS Batch, es necesario crear una definición de trabajo. Este proceso varía ligeramente para los trabajos paralelos entre un solo nodo y de varios nodos. Este tema cubre

específicamente cómo crear una definición de trabajo para un trabajo AWS Batch que no es un trabajo paralelo multinodo.

Puede crear una definición de trabajo paralelo de varios nodos en los recursos de Amazon Elastic Container Service. Para obtener más información, consulte the section called “Creación de una definición de trabajo paralelo de varios nodos”.

Temas

Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EC2

Para crear una nueva definición de trabajo en los recursos de Amazon EC2:

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS a utilizar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione Create (Crear).
5. Para el Tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
6. Para la Configuración de la plataforma EC2, desactive Habilitar el procesamiento paralelo de varios nodos.
7. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
8. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado `FAILED`. Para obtener más información, consulte Tiempos de espera de trabajo. El valor mínimo es de 60 segundos.
9. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad.

10. (Opcional) En Intentos de trabajo, introduzca el número de veces que AWS Batch intenta mover el trabajo al estado `RUNNABLE`. Ingrese un número entero entre 1 y 10.
11. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:

    - Reintentar: AWS Batch vuelve a intentarlo hasta alcanzar el número de intentos de trabajo que especificó.
    - Salir: AWS Batch deja de volver a intentar el trabajo.

    > **Important**
    >
    > Si elige Agregar evaluar al salir, debe configurar al menos un parámetro y elegir una Acción o Eliminar evaluar al salir.

12. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Elija Agregar nueva etiqueta e introduzca la clave y el valor opcional.
13. (Opcional) Active Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo a la tarea de Amazon ECS.
14. Seleccione Página siguiente.
15. En la sección Configuración del contenedor:

    a. En Imagen, elija la imagen Docker que desea utilizar para su trabajo. De manera predeterminada, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *`repository-url`*`/`*`image`*`:`*`tag`*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a `Image` en la sección Create a container (Crear un contenedor) de la API remota de Docker y el parámetro `IMAGE` de docker run.

    > **Note**
    >
    > La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por

ejemplo, las imágenes de Docker basadas en Arm solo pueden ejecutarse en recursos informáticos basados en Arm.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]` (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

b. Para la Sintaxis de comandos, elija Bash o JSON.

c. En Comando, especifique los comandos que desea transmitir al contenedor. Para comandos más sencillos, introduzca el comando como lo haría para una línea de comandos. A continuación, compruebe que el resultado JSON es correcto y se ha transferido a Docker daemon. Para comandos más complicados (por ejemplo, con caracteres especiales), utilice la sintaxis JSON.

> **Tip**
>
> Seleccione Información para ver los códigos de ejemplo Bash y JSON.

Este parámetro se asigna a `Cmd` en la sección Crear un contenedor de la API remota de Docker y el parámetro `COMMAND` se corresponde con docker run. Para obtener más información sobre el parámetro `CMD` de Docker, consulte https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> También puede usar valores predeterminados para la sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte Parámetros.

d. (Opcional) En Rol de ejecución, especifique un rol de IAM que conceda permiso a los agentes de contenedor de Amazon ECS para realizar llamadas a la API de AWS en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para las tareas. Para obtener más información, consulte Roles de IAM de ejecución de tareas de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

e. Para la configuración del rol de trabajo, elija un rol de IAM que tenga permisos para las API AWS. Esta característica utiliza roles de IAM de Amazon ECS para las tareas. Para obtener más información, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

> **Note**
>
> Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para trabajos deAWS Batch, consulte Creación de un rol de IAM y una política para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

16. En Parámetros, elija Agregar parámetros para añadir marcadores de sustitución de parámetros como pares Clave y Valores opcionales.
17. En la sección de Configuración del entorno:

    a. En CPU virtuales, introduzca la cantidad de CPU virtuales que quiera reservar para el contenedor. Este parámetro se asigna a `CpuShares` en la sección Crear un contenedor de la API remota de Docker y con la opción `--cpu-shares` de docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.

    b. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la cantidad de memoria que ha especificado aquí, se detiene el contenedor. Este parámetro se asigna a `Memory` en la sección Crear un contenedor de la

API remota de Docker y con la opción `--memory` de docker run. Debe especificar al menos 4 MiB de memoria para un trabajo.

> **Note**
>
> Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte Administración de la memoria de los recursos informáticos de las .

c. En Número de unidades GPU, seleccione el número de unidades GPU que desea reservar para el contenedor.

d. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.

e. (Opcional) En Secretos, seleccione Agregar secreto para añadir los secretos como pares de nombre-valor. Estos secretos están expuestos en el contenedor. Para obtener más información, consulte SecretOptions en Parámetros de definición de trabajo para ContainerProperties.

18. Seleccione Página siguiente.

19. En la sección de Configuración de Linux:

a. En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor. Este parámetro se asigna a `User` en la sección Crear un contenedor de la API remota de Docker y con la opción `--user` de docker run.

b. (Opcional) Para otorgar al contenedor de su trabajo permisos elevados en la instancia host (similares a los del usuario de `root`), arrastre el control deslizante Privilegiado hacia la derecha. Este parámetro se asigna a `Privileged` en la sección Crear un contenedor de la API remota de Docker y con la opción `--privileged` de docker run.

c. (Opcional) Active la opción Habilitar el proceso init para ejecutar un proceso `init` dentro del contenedor. Este proceso reenvía señales y recoge procesos.

20. (Opcional) En la sección de Configuración de Fylesystem:

a. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.

b. En Tamaño de memoria compartida, introduzca el tamaño (en MiB) del `/dev/shm` volumen de .

c. En Tamaño de intercambio máximo, introduzca la cantidad total de memoria de intercambio (en MiB) que puede utilizar el contenedor.

d. En Intercambio, introduzca un valor entre 0 y 100 para indicar el comportamiento de intercambio del contenedor. Si no especifica un valor y el intercambio está activado, el valor predeterminado es 60. Para obtener más información, consulte Intercambio en Parámetros de definición de trabajo para ContainerProperties.

e. (Opcional) Expandir Configuración adicional.

f. (Opcional) En el caso de Tmpfs, seleccione Agregar tmpfs para añadir una montura `tmpfs`.

g. (Opcional) En el caso de los Dispositivos, seleccione Agregar dispositivo para añadir un dispositivo:

   i. En Container path (Ruta del contenedor), especifique la ruta de la instancia del contenedor que va a exponer el dispositivo asignado a la instancia del host. Si lo deja en blanco, se utiliza la ruta del host en el contenedor.

   ii. En Host path (Ruta de host), especifique la ruta de un dispositivo de la instancia del host.

   iii. En la página Permisos, haga clic en uno o varios permisos para aplicarlos al dispositivo. Los permisos disponibles son READ, WRITE y MKNOD.

h. (Opcional) En Configuración de volúmenes, seleccione Agregar volumen para crear una lista de volúmenes que se transferirán al contenedor. Introduzca el Nombre y la Ruta de origen del volumen y, a continuación, seleccione Agregar volumen. También puede optar por activar Activar EFS.

i. (Opcional) En Puntos de montaje, elija la configuración Agregar puntos de montaje para agregar puntos de montaje a los volúmenes de datos. Debe especificar el volumen de origen y la ruta del contenedor. Estos puntos de montaje se transfieren al Docker daemon de una instancia de contenedor. También puede elegir que el volumen sea de Solo lectura.

j. (Opcional) En Configuración de Ulimits, seleccione Agregar ulimit para agregar un `ulimits` valor al contenedor. Introduzca los valores de Nombre, Límite flexible y Límite invariable y, a continuación, elija Agregar límite máximo.

21. (Opcional) En la sección de Configuración de registro:

    a. En el Controlador de registro, elija el controlador de registro que desee utilizar. Para obtener más información sobre los controladores de registro disponibles, consulte Controlador de registro en Parámetros de definición de trabajo para ContainerProperties.

> **Note**
>
> De forma predeterminada, se utiliza el controlador de registro `awslogs`.

b. En Opciones, elija Agregar opción para agregar una opción. Introduzca un par nombre-valor y, a continuación, elija Agregar opción.

c. En Secretos, seleccione Agregar secreto. Introduzca un par nombre-valor y, a continuación, seleccione Agregar secreto para añadir un secreto.

> **Tip**
>
> Para obtener más información, consulte SecretOptions en Parámetros de definición de trabajo para ContainerProperties.

22. Seleccione Página siguiente.
23. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, seleccione Edit (Editar). Cuando haya terminado, seleccione Crear definición de trabajo.

Creación de una definición de trabajo de un solo nodo en los recursos de AWS Fargate

Para crear una nueva definición de trabajo en los recursos de AWS Fargate:

1. Abra la consola AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, elija la Región de AWS a utilizar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione Crear.
5. En Tipo de orquestación, elija Fargate. Para obtener más información, consulte AWS Batch en AWS Fargate.
6. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

7. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado `FAILED`. Para obtener más información, consulte Tiempos de espera de trabajo. El valor mínimo es de 60 segundos.
8. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad respecto a los valores más bajos.
9. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Active Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo.
10. En la sección de Configuración de la plataforma Fargate:
    a. Para la Plataforma de tiempo de ejecución, elija la arquitectura del entorno informático.
    b. En Familia de sistemas operativos, seleccione el sistema operativo para el entorno informático.
    c. En Arquitectura CPU, elija la arquitectura vCPU.
    d. Para la versión de la plataforma Fargate, introduzca `LATEST` o una versión específica del entorno del tiempo de ejecución.
    e. (Opcional) Active Asignar IP pública para asignar una dirección IP a una interfaz de redes de trabajo Fargate. Para que un trabajo que se ejecuta en una subred privada envíe tráfico saliente a Internet, la subred privada requiere que se conecte una puerta de enlace NAT para enrutar las solicitudes a Internet. Es posible que desee hacer esto para poder extraer imágenes de contenedores. Para obtener más información, consulte Amazon ECS task networking (Integración en red de las tareas de Amazon ECS) en la Guía para desarrolladores de Amazon Elastic Container Service.
    f. (Opcional) En Almacenamiento efímero, introduzca la cantidad de almacenamiento efímero que se va a asignar a la tarea. La cantidad de almacenamiento efímero debe estar entre 21 GiB y 200 GiB. De forma predeterminada, se asignan 20 GiB de almacenamiento efímero si no ingresa un valor.

> **Note**
>
> El almacenamiento efímero requiere la versión de la plataforma 1.4 Fargate o una posterior.

g. Para el Rol de ejecución, especifique un rol de IAM que conceda permiso a los agentes de contenedor de Amazon ECS y Fargate para realizar llamadas a la API de AWS en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, incluidos los requisitos previos de configuración, consulte Roles de IAM para tareas de ejecución de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

h. Para los Intentos de trabajo, introduzca el número de veces que AWS Batch intenta mover el trabajo al estado `RUNNABLE`. Ingrese un número entero entre 1 y 10.

i. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:

- Reintentar: AWS Batch vuelve a intentarlo hasta alcanzar el número de intentos de trabajo que especificó.
- Salir: AWS Batch deja de volver a intentar el trabajo.

> **Important**
>
> Si elige Agregar evaluar al salir, debe configurar al menos un parámetro y elegir una Acción o Eliminar evaluar al salir.

11. Seleccione Página siguiente.
12. En la sección Configuración del contenedor:

a. En Imagen, elija la imagen de Docker que desea utilizar para su trabajo. Por defecto, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *`repository-url`*`/`*`image`*`:`*`tag`*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), puntos (.), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a `Image` en la sección Crear un contenedor de la API remota de Docker y el parámetro `IMAGE` de docker run.

> **Note**
>
> La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en Arm solo pueden ejecutarse en recursos informáticos basados en Arm.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]` (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

b. Para la Sintaxis de comandos, elija Bash o JSON.

c. En Comando, especifique los comandos que desea transmitir al contenedor. Para comandos sencillos, introdúzcalo como lo haría para una línea de comandos y, a continuación, compruebe que el resultado JSON es correcto. Se pasa al daemon Docker . Para comandos más complicados (por ejemplo, con caracteres especiales), utilice la sintaxis JSON.

> **Tip**
>
> Seleccione Información para ver los códigos de ejemplo Bash y JSON.

Este parámetro se asigna a `Cmd` en la sección Crear un contenedor de la API remota de Docker y el parámetro `COMMAND` se corresponde con docker run. Para obtener más

información sobre el parámetro `CMD` de Docker, consulte https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> También puede usar valores predeterminados para la sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte Parámetros.

d. (Opcional) Añada parámetros a la definición del trabajo como asignaciones de nombre-valor para anular los valores predeterminados de la definición del trabajo. Para añadir un parámetro:

   - En Parámetros, elija Agregar parámetros, introduzca un par de nombre-valor y, a continuación, elija Agregar parámetro.

> **Important**
>
> Si elige Agregar parámetro, debe configurar al menos un parámetro o elegir Eliminar parámetro

e. En la sección de Configuración del entorno:

   i. Para la configuración del rol de trabajo, elija un rol de IAM que tenga permisos para las API AWS. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

> **Note**
>
> Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para trabajos de AWS Batch, consulte Creación de un rol de IAM y una política para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

   ii. En CPU virtuales, introduzca la cantidad de CPU virtuales que quiera reservar para el contenedor. Este parámetro se asigna a `CpuShares` en la sección Crear un

contenedor de la API remota de Docker y con la opción `--cpu-shares` de docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.

iii. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se asigna a `Memory` en la sección Crear un contenedor de la API remota de Docker y con la opción `--memory` de docker run. Debe especificar al menos 4 MiB de memoria para un trabajo.

Si usa GuardDuty Runtime Monitoring, hay una ligera sobrecarga de memoria para el agente GuardDuty de seguridad. Por lo tanto, el límite de memoria debe incluir el tamaño del agente de GuardDuty seguridad. Para obtener información sobre los límites de memoria del agente de GuardDuty seguridad, consulte los límites de CPU y memoria en la Guía del GuardDuty usuario. Para obtener información sobre las prácticas recomendadas, consulte Cómo corregir los errores de falta de memoria en mis tareas de Fargate después de activar Runtime Monitoring en la Guía para desarrolladores de Amazon ECS.

> **Note**
>
> Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte Administración de la memoria de los recursos informáticos de las .

f. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.

g. (Opcional) En Secretos, seleccione Agregar secreto para añadir los secretos como pares de nombre-valor. Estos secretos están expuestos en el contenedor. Para obtener más información, consulte SecretOptions en Parámetros de definición de trabajo para ContainerProperties.

h. Seleccione Página siguiente.

13. (Opcional) En la sección de Configuración de Linux:

a. En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor.

b. Active la opción Habilitar el proceso para ejecutar un proceso dentro del contenedor. Este proceso reenvía señales y recoge procesos.

c. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.

d. (Opcional) Expandir Configuración adicional.

e. Para la Configuración de puntos de montaje, elija la configuración Agregar puntos de montaje para agregar puntos de montaje a los volúmenes de datos. Debe especificar el volumen de origen y la ruta del contenedor. Estos puntos de montaje se transfieren al Docker daemon de una instancia de contenedor.

f. Para la Configuración de volúmenes, seleccione Agregar volumen para crear una lista de volúmenes que se transferirán al contenedor. Introduzca el nombre y la ruta de origen del volumen y, a continuación, seleccione Agregar volumen.

g. En la sección de Configuración de registro:

i. (Opcional) En el Controlador de registro, elija el controlador de registro que desee utilizar. Para obtener más información sobre los controladores de registro disponibles, consulte Controlador de registro en Parámetros de definición de trabajo para ContainerProperties.

> **Note**
>
> De forma predeterminada, se utiliza el controlador de registro `awslogs`.

ii. (Opcional) En Opciones, elija Agregar opción para agregar una opción. Introduzca un par nombre-valor y, a continuación, elija Agregar opción.

iii. (Opcional) En Secretos, seleccione Agregar secreto para añadir un secreto. A continuación, introduzca un par nombre-valor y seleccione Agregar secreto.

> **Tip**
>
> Para obtener más información, consulte SecretOptions en Parámetros de definición de trabajo para ContainerProperties.

14. Seleccione Página siguiente.

15. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Crear definición de trabajo.

Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EKS

Para crear una nueva definición de trabajo en los recursos de Amazon Elastic Kubernetes Service:

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, elija la Región de AWS a utilizar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione Create (Crear).
5. Para el tipo de orquestación, elija Elastic Kubernetes Service (EKS).
6. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
7. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado `FAILED`. Para obtener más información, consulte Tiempos de espera de trabajo. El valor mínimo es de 60 segundos.
8. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad respecto a los valores más bajos.
9. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso.
10. Seleccione Página siguiente.
11. En la sección de propiedades de pod EKS:

    a. En Nombre de cuenta de servicio, introduzca una cuenta que proporcione una identidad para los procesos que se ejecutan en un pod.

    b. Active la Red de host para usar el modelo de red Kubernetes pod y abra un puerto de escucha para las conexiones entrantes. Desactive esta configuración solo para las comunicaciones salientes.

    c. Para la política DNS, elija una de las siguientes opciones:

    - Sin valor (nulo): pod ignora la configuración de DNS del entorno Kubernetes.

- Predeterminado: pod hereda la configuración de resolución de nombres del nodo en el que se ejecuta.

> **Note**
>
> Si no se especifica una política de DNS, la política de DNS Valor predeterminado no es la política de DNS predeterminada. En su lugar, se utiliza ClusterFirst.

- ClusterFirst: cualquier consulta de DNS que no coincida con el sufijo de dominio del clúster configurado se reenvía al servidor de nombres ascendente heredado del nodo.
- ClusterFirstWithHostNet: se usa si la Red de host está activada.

d. (Opcional) En Etiquetas de los pods, seleccione Agregar etiquetas de pod y, a continuación, introduzca un par de nombre-valor.

> **Important**
>
> El prefijo de una etiqueta de pod no puede contener`kubernetes.io/`, `k8s.io/` o `batch.amazonaws.com/`.

e. Seleccione Página siguiente.

f. En la sección Configuración del contenedor:

i. Para el Nombre, escriba un nombre único para el contenedor. El nombre debe empezar por una letra o un número y puede tener un máximo de 63 caracteres. Puede contener letras mayúsculas y minúsculas, números y guiones (-).

ii. En Imagen, elija la imagen Docker que desea utilizar para su trabajo. Por defecto, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *`repository-url/image:tag`*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), puntos (.), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a `Image` en la sección Crear un contenedor de la API remota de Docker y el parámetro `IMAGE` de docker run

> **Note**
>
> La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por

ejemplo, las imágenes de Docker basadas en Arm solo pueden ejecutarse en recursos informáticos basados en Arm.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]` (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

iii. (Opcional) En Política de extracción de imágenes, elija cuándo se extraerán las imágenes.

iv. (Opcional) En Comando, ingrese un comando Bash o JSON para pasarlo al contenedor.

v. (Opcional) En Argumentos, introduzca los argumentos para pasar al contenedor. Si no se proporciona un argumento, se utiliza el comando contenedor de imágenes.

g. (Opcional) Puede añadir parámetros a la definición del trabajo como asignaciones de nombre-valor para anular los valores predeterminados de la definición del trabajo. Para añadir un parámetro:

- En Parámetros, introduzca un par de nombre-valor y, a continuación, elija Agregar parámetro.

> **Important**
>
> Si elige Agregar parámetro, debe configurar al menos un parámetro o elegir Eliminar parámetro

h. En la sección de Configuración del entorno:

i. En CPU virtuales, introduzca la cantidad de CPU virtuales que quiera reservar para el contenedor. Este parámetro se asigna a `CpuShares` en la sección Crear un contenedor de la API remota de Docker y con la opción `--cpu-shares` de docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.

ii. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se asigna a `Memory` en la sección Crear un contenedor de la API remota de Docker y con la opción `--memory` de docker run. Debe especificar al menos 4 MiB de memoria para un trabajo.

> **Note**
>
> Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte Administración de la memoria de los recursos informáticos de las .

i. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.

j. (Opcional) En Montaje del volumen:

i. Seleccione Agregar montaje del volumen.

ii. Introduzca un Nombre y, a continuación, introduzca una Ruta de montaje en el contenedor donde está montado el volumen.

iii. Seleccione Solo lectura para eliminar los permisos de escritura del volumen.

iv. Seleccione Agregar montaje del volumen.

k. (Opcional) En Ejecutar como usuario, introduzca un ID de usuario para ejecutar el proceso del contenedor.

> **Note**
>
> El ID de usuario debe existir en la imagen para que se ejecute el contenedor.

l. (Opcional) En Ejecutar como grupo, introduzca un ID de grupo para ejecutar el proceso del contenedor en tiempo de ejecución.

> **Note**
>
> El ID de grupo debe existir en la imagen para que se ejecute el contenedor.

m. (Opcional) Para otorgar al contenedor de su trabajo permisos elevados en la instancia host (similares a los del usuario de `root`), arrastre el control deslizante Privilegiado hacia la derecha. Este parámetro se asigna a `Privileged` en la sección Crear un contenedor de la API remota de Docker y con la opción `--privileged` de docker run.

n. (Opcional) Active el sistema de Archivos raíz de solo lectura para eliminar el acceso de escritura al sistema de archivos raíz.

o. (Opcional) Active la opción Ejecutar como usuario no raíz para ejecutar los contenedores como usuario no raíz. pod

> **Note**
>
> Si la opción Ejecutar como no raíz está activada, kubelet valida la imagen en tiempo de ejecución para comprobar que la imagen no se ejecuta como UID 0.

p. Seleccione Página siguiente.

12. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, seleccione Edit (Editar). Cuando haya terminado, seleccione Crear definición de trabajo.

Creación de una definición de trabajo paralelo de varios nodos

Antes de ejecutar trabajos en AWS Batch, es necesario crear una definición de trabajo. Este proceso varía ligeramente para los trabajos paralelos entre un solo nodo y de varios nodos. En este tema, se explica específicamente cómo crear una definición de trabajo para un trabajo paralelo de varios nodos de AWS Batch. Para obtener más información, consulte Trabajos paralelos de varios nodos.

> **Note**
>
> AWS Fargate no admite trabajos paralelos de varios nodos.

Creación de una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2

Para crear una definición de trabajo paralelo de un solo nodo, consulte Creación de una definición de trabajo de un solo nodo .

Para crear una definición de trabajo paralelo de varios nodos en recursos de Amazon Elastic Compute Cloud:

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS a utilizar.
3. En el panel de navegación, seleccione Definiciones de trabajo.
4. Seleccione Create (Crear).
5. Para el Tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
6. En Habilitar el paralelo de varios nodos, active el paralelo de varios nodos.
7. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
8. (Opcional) En Tiempo de espera de ejecución, especifique el número máximo de segundos que desea permitir que el trabajo intente ejecutarse. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado `FAILED`. Para obtener más información, consulte Tiempos de espera de trabajo.
9. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad respecto a los valores más bajos.
10. (Opcional) En Intentos de trabajo, introduzca el número de veces que AWS Batch intenta mover el trabajo al estado `RUNNABLE`. Ingrese un número entero entre 1 y 10.
11. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:

    - Reintentar: AWS Batch vuelve a intentarlo hasta alcanzar el número de intentos de trabajo que especificó.
    - Salir: AWS Batch deja de volver a intentar el trabajo.

> **Important**
>
> Si elige Agregar evaluar al salir, debe configurar al menos un parámetro y elegir una Acción o Eliminar evaluar al salir.

12. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Elija Agregar nueva etiqueta e introduzca la clave y el valor opcional. También puede activar Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo a la tarea de Amazon ECS.
13. Seleccione Página siguiente.
14. En Number of nodes (Número de nodos), introduzca el número total de nodos que desea utilizar en el trabajo.
15. En Main node (Nodo principal), introduzca el índice de nodo que desea utilizar para el nodo principal. El índice de nodo principal predeterminado es `0`.
16. En Tipo de instancia, elija un tipo de instancia.

> **Note**
>
> El tipo de instancia que elija se aplica a todos los nodos.

17. En Parámetros, elija Agregar parámetros para añadir marcadores de sustitución de parámetros como pares Clave y Valores opcionales.
18. En la sección Rangos de nodos:

    a. Seleccione Agregar rango de nodos. Esto crea una sección de Rango de nodos.

    b. En Target nodes (Nodos de destino), especifique el rango del grupo de nodos utilizando la notación *`range_start`*`:`*`range_end`*.

    Puede crear hasta cinco rangos de nodos para los nodos que ha especificado para el trabajo. Los rangos de nodos utilizan el valor de índice para un nodo, y el índice de nodo comienza a partir de 0. Asegúrese de que el valor del índice final del rango del grupo de nodos final sea uno menos que el número de nodos que especificó. Por ejemplo, supongamos que ha especificado 10 nodos y desea utilizar un único grupo de nodos. Entonces, el rango final será 9.

c. En Imagen, elija la imagen Docker que desea utilizar para su trabajo. De manera predeterminada, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *repository-url*/*image*:*tag*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a `Image` en la sección Create a container (Crear un contenedor) de la API remota de Docker y el parámetro `IMAGE` de docker run.

> **Note**
>
> La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en Arm solo pueden ejecutarse en recursos informáticos basados en Arm.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/`*registry_alias*`/`*my-web-app*`:`*latest*).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]`. Por ejemplo, *aws_account_id*`.dkr.ecr.`*region*`.amazonaws.com/`*my-web-app*`:`*latest* .
- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
- Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

d. Para la Sintaxis de comandos, elija Bash o JSON.

e. En Comando, especifique los comandos que desea transmitir al contenedor. Los comandos sencillos pueden introducirse en la pestaña Delimitado por espacios tal como lo hace en un símbolo del sistema. A continuación, compruebe que el resultado JSON es correcto. El resultado de JSON se pasa a Docker daemon. Si los comandos son más complejos

(con caracteres especiales, por ejemplo), acceda a la pestaña JSON e introduzca allí el equivalente a la matriz de cadenas.

Este parámetro se asigna a `Cmd` en la sección Crear un contenedor de la API remota de Docker y el parámetro `COMMAND` se corresponde con docker run. Para obtener más información sobre el parámetro `CMD` de Docker, consulte https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> También puede usar valores predeterminados para la sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte Parámetros.

f. En CPU virtuales, especifique la cantidad de CPU virtuales que quiera reservar para el contenedor. Este parámetro se asigna a `CpuShares` en la sección Crear un contenedor de la API remota de Docker y con la opción `--cpu-shares` de docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.

g. En Memoria, especifique límite máximo (en MiB) de memoria que quiera presentarle al contenedor del trabajo. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se asigna a `Memory` en la sección Crear un contenedor de la API remota de Docker y con la opción `--memory` de docker run. Debe especificar al menos 4 MiB de memoria para un trabajo.

> **Note**
>
> Para maximizar la utilización de los recursos, puede proporcionar a los trabajos la mayor cantidad de memoria posible para un tipo de instancia determinado. Para obtener más información, consulte Administración de la memoria de los recursos informáticos de las .

h. (Opcional) En Número de GPU, especifique el número de GPU que utiliza su trabajo. El trabajo se ejecuta en un contenedor con el número especificado de GPU anclado a dicho contenedor.

i. (Opcional) En Rol de trabajo, especifique un rol de IAM que le otorgue al contenedor en su trabajo permisos para usar las API de AWS. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información,

incluidos los requisitos previos de configuración, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

> **Note**
>
> Para trabajos que se ejecutan en recursos de Fargate, se requiere un puesto de trabajo.

> **Note**
>
> Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para trabajos deAWS Batch, consulte Creación de un rol de IAM y una política para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

j. (Opcional) En Rol de ejecución, especifique un rol de IAM que conceda permiso a los agentes de contenedor de Amazon ECS para realizar llamadas a la API de AWS en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte Roles de IAM de ejecución de tareas de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

19. (Opcional) Expandir Configuración adicional:

    a. En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.

    b. En Configuración de rol de trabajo, especifique un rol de IAM que le otorgue al contenedor en su trabajo permisos para usar las API de AWS. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, incluidos los requisitos previos de configuración, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

> **Note**
>
> Para trabajos que se ejecutan en recursos de Fargate, se requiere un puesto de trabajo.

> **Note**
>
> Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para trabajos de AWS Batch, consulte Creación de un rol de IAM y una política para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

c. Para el Rol de ejecución, especifique un rol de IAM que conceda permiso a los agentes de contenedor de Amazon ECS para realizar llamadas a la API de AWS en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte Roles de IAM de ejecución de tareas de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

20. En la sección Configuración de seguridad:

    a. (Opcional) Para concederle privilegios elevados al contenedor del trabajo en la instancia del host (similares a los de un usuario `root`), active Privilegiado. Este parámetro se asigna a `Privileged` en la sección Crear un contenedor de la API remota de Docker y con la opción `--privileged` de docker run.

    b. (Opcional) En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor. Este parámetro se asigna a `User` en la sección Crear un contenedor de la API remota de Docker y con la opción `--user` de docker run.

    c. (Opcional) En Secretos, seleccione Agregar secreto para añadir los secretos como pares de nombre-valor. Estos secretos están expuestos en el contenedor. Para obtener más información, consulte SecretOptions en Parámetros de definición de trabajo para ContainerProperties.

21. En la sección de Configuración de Linux:

    a. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.

b. (Opcional) Active la opción Habilitar el proceso init para ejecutar un proceso `init` dentro del contenedor. Este proceso reenvía señales y recoge procesos.

c. En Tamaño de memoria compartida, introduzca el tamaño (en MiB) del `/dev/shm` volumen de .

d. En Tamaño de intercambio máximo, introduzca la cantidad total de memoria de intercambio (en MiB) que puede utilizar el contenedor.

e. En Intercambio, introduzca un valor entre 0 y 100 para indicar el comportamiento de intercambio del contenedor. Si no especifica un valor y el intercambio está activado, el valor predeterminado es 60. Para obtener más información, consulte Intercambio en Parámetros de definición de trabajo para ContainerProperties.

f. (Opcional) En el caso de los Dispositivos, seleccione Agregar dispositivo para añadir un dispositivo:

   i. En Container path (Ruta del contenedor), especifique la ruta de la instancia del contenedor que va a exponer el dispositivo asignado a la instancia del host. Si lo deja en blanco, se utiliza la ruta del host en el contenedor.

   ii. En Host path (Ruta de host), especifique la ruta de un dispositivo de la instancia del host.

   iii. En la página Permisos, haga clic en uno o varios permisos para aplicarlos al dispositivo. Los permisos disponibles son READ, WRITE y MKNOD.

22. (Opcional) En Puntos de montaje, elija la configuración Agregar puntos de montaje para agregar puntos de montaje a los volúmenes de datos. Debe especificar el volumen de origen y la ruta del contenedor. Estos puntos de montaje se transfieren al daemon Docker de una instancia de contenedor. También puede elegir que el volumen sea de Solo lectura.

23. (Opcional) En Configuración de Ulimits, seleccione Agregar ulimit para agregar un `ulimits` valor al contenedor. Introduzca los valores de Nombre, Límite flexible y Límite invariable y, a continuación, elija Agregar límite máximo.

24. (Opcional) En Configuración de volúmenes, seleccione Agregar volumen para crear una lista de volúmenes que se transferirán al contenedor. Introduzca el Nombre y la Ruta de origen del volumen y, a continuación, seleccione Agregar volumen. También puede optar por activar Activar EFS.

25. (Opcional) En el caso de Tmpfs, seleccione Agregar tmpfs para añadir una montura `tmpfs`.

26. (Opcional) En la sección de Configuración de registro:

a. En el Controlador de registro, elija el controlador de registro que desee utilizar. Para obtener más información sobre los controladores de registro disponibles, consulte Controlador de registro en Parámetros de definición de trabajo para ContainerProperties.

> **Note**
>
> De forma predeterminada, se utiliza el controlador de registro `awslogs`.

b. En Opciones, elija Agregar opción para agregar una opción. Introduzca un par nombre-valor y, a continuación, elija Agregar opción.

c. En Secretos, seleccione Agregar secreto. Introduzca un par nombre-valor y, a continuación, seleccione Agregar secreto para añadir un secreto.

> **Tip**
>
> Para obtener más información, consulte SecretOptions en Parámetros de definición de trabajo para ContainerProperties.

27. Seleccione Página siguiente.
28. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, seleccione Edit (Editar). Cuando haya terminado, seleccione Crear definición de trabajo.

Creación de definiciones de trabajo mediante ContainerProperties

La siguiente es una plantilla de definición de trabajo vacía que incluye un único contenedor. Puede usar esta plantilla para crear su definición de trabajo, que luego se puede guardar en un archivo y usar con la AWS CLI `--cli-input-json` opción. Para obtener más información sobre estos parámetros, consulte Parámetros de definición de trabajo para ContainerProperties.

```
{
    "jobDefinitionName": "",
    "type": "container",
    "parameters": {
        "KeyName": ""
    },
    "schedulingPriority": 0,
```

```
    "containerProperties": {
        "image": "",
        "vcpus": 0,
        "memory": 0,
        "command": [
            ""
        ],
        "jobRoleArn": "",
        "executionRoleArn": "",
        "volumes": [
            {
                "host": {
                    "sourcePath": ""
                },
                "name": "",
                "efsVolumeConfiguration": {
                    "fileSystemId": "",
                    "rootDirectory": "",
                    "transitEncryption": "ENABLED",
                    "transitEncryptionPort": 0,
                    "authorizationConfig": {
                        "accessPointId": "",
                        "iam": "DISABLED"
                    }
                }
            }
        ],
        "environment": [
            {
                "name": "",
                "value": ""
            }
        ],
        "mountPoints": [
            {
                "containerPath": "",
                "readOnly": true,
                "sourceVolume": ""
            }
        ],
        "readonlyRootFilesystem": true,
        "privileged": true,
        "ulimits": [
            {
```

```
                "hardLimit": 0,
                "name": "",
                "softLimit": 0
            }
        ],
        "user": "",
        "instanceType": "",
        "resourceRequirements": [
            {
                "value": "",
                "type": "MEMORY"
            }
        ],
        "linuxParameters": {
            "devices": [
                {
                    "hostPath": "",
                    "containerPath": "",
                    "permissions": [
                        "WRITE"
                    ]
                }
            ],
            "initProcessEnabled": true,
            "sharedMemorySize": 0,
            "tmpfs": [
                {
                    "containerPath": "",
                    "size": 0,
                    "mountOptions": [
                        ""
                    ]
                }
            ],
            "maxSwap": 0,
            "swappiness": 0
        },
        "logConfiguration": {
            "logDriver": "syslog",
            "options": {
                "KeyName": ""
            },
            "secretOptions": [
                {
```

```
                    "name": "",
                    "valueFrom": ""
                }
            ]
        },
        "secrets": [
            {
                "name": "",
                "valueFrom": ""
            }
        ],
        "networkConfiguration": {
            "assignPublicIp": "DISABLED"
        },
        "fargatePlatformConfiguration": {
            "platformVersion": ""
        }
    },
    "nodeProperties": {
        "numNodes": 0,
        "mainNode": 0,
        "nodeRangeProperties": [
            {
                "targetNodes": "",
                "container": {
                    "image": "",
                    "vcpus": 0,
                    "memory": 0,
                    "command": [
                        ""
                    ],
                    "jobRoleArn": "",
                    "executionRoleArn": "",
                    "volumes": [
                        {
                            "host": {
                                "sourcePath": ""
                            },
                            "name": "",
                            "efsVolumeConfiguration": {
                                "fileSystemId": "",
                                "rootDirectory": "",
                                "transitEncryption": "DISABLED",
                                "transitEncryptionPort": 0,
```

```
                "authorizationConfig": {
                    "accessPointId": "",
                    "iam": "ENABLED"
                }
            }
        }
    ],
    "environment": [
        {
            "name": "",
            "value": ""
        }
    ],
    "mountPoints": [
        {
            "containerPath": "",
            "readOnly": true,
            "sourceVolume": ""
        }
    ],
    "readonlyRootFilesystem": true,
    "privileged": true,
    "ulimits": [
        {
            "hardLimit": 0,
            "name": "",
            "softLimit": 0
        }
    ],
    "user": "",
    "instanceType": "",
    "resourceRequirements": [
        {
            "value": "",
            "type": "MEMORY"
        }
    ],
    "linuxParameters": {
        "devices": [
            {
                "hostPath": "",
                "containerPath": "",
                "permissions": [
                    "WRITE"
```

```
                            ]
                        }
                    ],
                    "initProcessEnabled": true,
                    "sharedMemorySize": 0,
                    "tmpfs": [
                        {
                            "containerPath": "",
                            "size": 0,
                            "mountOptions": [
                                ""
                            ]
                        }
                    ],
                    "maxSwap": 0,
                    "swappiness": 0
                },
                "logConfiguration": {
                    "logDriver": "awslogs",
                    "options": {
                        "KeyName": ""
                    },
                    "secretOptions": [
                        {
                            "name": "",
                            "valueFrom": ""
                        }
                    ]
                },
                "secrets": [
                    {
                        "name": "",
                        "valueFrom": ""
                    }
                ],
                "networkConfiguration": {
                    "assignPublicIp": "DISABLED"
                },
                "fargatePlatformConfiguration": {
                    "platformVersion": ""
                }
            }
        }
    ]
```

```
    },
    "retryStrategy": {
        "attempts": 0,
        "evaluateOnExit": [
            {
                "onStatusReason": "",
                "onReason": "",
                "onExitCode": "",
                "action": "RETRY"
            }
        ]
    },
    "propagateTags": true,
    "timeout": {
        "attemptDurationSeconds": 0
    },
    "tags": {
        "KeyName": ""
    },
    "platformCapabilities": [
        "EC2"
    ],
    "eksProperties": {
        "podProperties": {
            "serviceAccountName": "",
            "hostNetwork": true,
            "dnsPolicy": "",
            "containers": [
                {
                    "name": "",
                    "image": "",
                    "imagePullPolicy": "",
                    "command": [
                        ""
                    ],
                    "args": [
                        ""
                    ],
                    "env": [
                        {
                            "name": "",
                            "value": ""
                        }
                    ],
```

```
                    "resources": {
                        "limits": {
                            "KeyName": ""
                        },
                        "requests": {
                            "KeyName": ""
                        }
                    },
                    "volumeMounts": [
                        {
                            "name": "",
                            "mountPath": "",
                            "readOnly": true
                        }
                    ],
                    "securityContext": {
                        "runAsUser": 0,
                        "runAsGroup": 0,
                        "privileged": true,
                        "readOnlyRootFilesystem": true,
                        "runAsNonRoot": true
                    }
                }
            ],
            "volumes": [
                {
                    "name": "",
                    "hostPath": {
                        "path": ""
                    },
                    "emptyDir": {
                        "medium": "",
                        "sizeLimit": ""
                    },
                    "secret": {
                        "secretName": "",
                        "optional": true
                    }
                }
            ]
        }
    }
}
```

> **Note**
>
> Puede generar una plantilla de definición de trabajo en un solo contenedor con el siguiente comando: AWS CLI
>
> ```
> $ aws batch register-job-definition --generate-cli-skeleton
> ```

Parámetros de definición de trabajo para ContainerProperties

Las definiciones de trabajo que `ContainerProperties`se utilizan se dividen en varias partes:

- nombre de la definición de trabajo
- el tipo de definición de trabajo
- los valores predeterminados de marcador de sustitución de parámetros
- el contenedor de propiedades del trabajo
- las propiedades de Amazon EKS para la definición del trabajo que son necesarias para los trabajos que se ejecutan en los recursos de Amazon EKS
- las propiedades de nodo necesarias para un trabajo paralelo de varios nodos
- las capacidades de la plataforma que son necesarias para los trabajos que se ejecutan en los recursos de Fargate
- los detalles de propagación de etiquetas predeterminados de la definición del trabajo
- la estrategia de reintento predeterminada para la definición del trabajo
- la prioridad de programación predeterminada para la definición del trabajo
- las etiquetas predeterminadas para la definición del trabajo
- el tiempo de espera predeterminado de la definición del trabajo

Contenido

- Propiedades de Amazon EKS
- Capacidades de plataforma
- Propagar etiquetas
- Propiedades del nodo
- Estrategia de reintento
- Prioridad de programación
- Etiquetas
- Tiempo de espera

Nombre de la definición de trabajo

`jobDefinitionName`

Al registrar una definición de trabajo, es necesario especificar un nombre. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_). A la primera definición de trabajo registrado con ese nombre se le da asigna la revisión 1. A las definiciones de trabajo posteriores registradas con el mismo nombre se les asignan números de revisión en orden ascendente.

Tipo: cadena

Obligatorio: sí

Tipo

`type`

Al registrar una definición de trabajo, es necesario especificar el tipo de trabajo. Si el trabajo se ejecuta en recursos de Fargate, entonces `multinode` no es compatible. Para obtener más información sobre los trabajos paralelos de varios nodos, consulte the section called “Creación de una definición de trabajo paralelo de varios nodos”.

Tipo: cadena

Valores válidos: `container` | `multinode`

Obligatorio: sí

Parámetros

`parameters`

Al enviar un trabajo, es posible especificar parámetros que sustituyan los marcadores de posición o ignoren los parámetros de definición de trabajo predeterminados. Los parámetros de las solicitudes de envío de trabajo prevalecen sobre los predeterminados de la definición de trabajo. Esto significa que puede usar la misma definición de trabajo para varios trabajos que usen el mismo formato. También puede cambiar mediante programación los valores del comando en el momento del envío.

Tipo: mapa de cadena a cadena

Obligatorio: no

Al registrar una definición de trabajo, es posible utilizar marcadores de posición de sustitución de parámetros en el campo `command` de las propiedades de un contenedor de trabajo. La sintaxis es la siguiente.

```
"command": [
    "ffmpeg",
    "-i",
    "Ref::inputfile",
    "-c",
    "Ref::codec",
    "-o",
    "Ref::outputfile"
]
```

En el ejemplo anterior, hay marcadores de posición de sustitución de parámetros *`Ref::inputfile`*, *`Ref::codec`* y *`Ref::outputfile`* en el comando. Puede usar el objeto `parameters` de la definición de trabajo para establecer valores predeterminados para estos marcadores de posición. Por ejemplo, para definir un valor predeterminado para el marcador de posición *`Ref::codec`*, especifique lo siguiente en la definición del trabajo:

```
"parameters" : {"codec" : "mp4"}
```

Cuando esta definición de trabajo se envíe para su ejecución, el argumento *`Ref::codec`* del comando para el contenedor se sustituirá por el valor predeterminado, `mp4`.

Propiedades de contenedor

Al registrar una definición de trabajo, especifique una lista de las propiedades de contenedor que se transmiten al daemon de Docker en una instancia de contenedor cuando se coloca el trabajo. Las siguientes propiedades de contenedor se permiten en una definición de trabajo. Para los trabajos de un solo nodo, estas propiedades de contenedor se establecen en el nivel de definición de trabajo. En los trabajos paralelos de varios nodos, las propiedades de contenedor se establecen en el nivel Propiedades del nodo para cada grupo de nodos.

`command`

El comando que se transfiere al contenedor. Este parámetro se asigna a `Cmd` en la sección Crear un contenedor de la API remota de Docker y el parámetro `COMMAND` se corresponde con docker run. Para obtener más información sobre el parámetro `CMD` de Docker, consulte https://docs.docker.com/engine/reference/builder/#cmd.

```
"command": ["string", ...]
```

Tipo: matriz de cadenas

Obligatorio: no

`environment`

Las variables de entorno a transferir a un contenedor. Este parámetro se asigna a `Env` en la sección Crear un contenedor de la API remota de Docker y con la opción `--env` de docker run.

> **Important**
>
> No es recomendable usar variables del entorno en texto sin formato para información confidencial, como los datos de la credencial.

> **Note**
>
> Las variables de entorno no pueden empezar por `AWS_BATCH`. Esta convención de nomenclatura está reservada para las variables que establece el AWS Batch servicio.

Tipo: matriz de pares clave-valor

Obligatorio: no

`name`

El nombre de la variable de entorno.

Tipo: cadena

Obligatorio: sí, si se utiliza `environment`.

`value`

El valor de la variable de entorno.

Tipo: cadena

Obligatorio: sí, si se utiliza `environment`.

```
"environment" : [
    { "name" : "envName1", "value" : "envValue1" },
    { "name" : "envName2", "value" : "envValue2" }
]
```

`executionRoleArn`

Al registrar una definición de trabajo, es posible especificar un rol de IAM. El rol le concede al agente de contenedor de Amazon ECS permisos para llamar a las acciones de la API que se especifican en sus políticas asociadas, en nombre de quien registra la definición de trabajo. Para los trabajos que se ejecutan en recursos de Fargate, debe proporcionar un rol de ejecución. Para obtener más información, consulte AWS Batch función de IAM de ejecución.

Tipo: string

Obligatorio: no

`fargatePlatformConfiguration`

La configuración de la plataforma para trabajos que se ejecutan en recursos de Fargate. Los trabajos que se ejecutan en los recursos de EC2 no deben especificar este parámetro.

Tipo: objeto FargatePlatformConfiguration

Obligatorio: no

`platformVersion`

La versión de la plataforma AWS Fargate utilizada para los trabajos o `LATEST` para usar una versión reciente y aprobada de la plataforma Fargate AWS .

Tipo: cadena

Valor predeterminado: `LATEST`

Obligatorio: no

`image`

La imagen que se utiliza para iniciar un trabajo. Esta cadena se transfiere directamente al daemon de Docker. Las imágenes del registro de Docker Hub están disponibles de forma predeterminada. También es posible especificar otros repositorios con *`repository-url`*`/`*`image`*`:`*`tag`*. Se permiten hasta 255 letras (mayúsculas y minúsculas), números, guiones, caracteres de subrayado, comas, puntos, barras diagonales y signos numéricos. Este parámetro se asigna a `Image` en la sección Crear un contenedor de la API remota de Docker y el parámetro `IMAGE` de docker run.

> **Note**
>
> La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos de computación en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en Arm solo pueden ejecutarse en recursos de computación basados en Arm.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository:[tag]`. Por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*.
- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).

- Las imágenes de otros repositorios de Docker Hub se identifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).

Tipo: cadena

Obligatorio: sí

`instanceType`

El tipo de instancia que se va a utilizar para un trabajo en paralelo de varios nodos. Todos los grupos de nodos de un trabajo paralelo de varios nodos deben utilizar el mismo tipo de instancia. Este parámetro no es aplicable a trabajos de contenedor de un solo nodo o para trabajos que se ejecutan en recursos de Fargate.

Tipo: cadena

Obligatorio: no

`jobRoleArn`

Al registrar una definición de trabajo, es posible especificar un rol de IAM. El rol le concede al contenedor de trabajo permisos para llamar a las acciones de la API que se especifican en sus políticas asociadas, en nombre de quien registra la definición de trabajo. Para obtener más información, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

Tipo: cadena

Obligatorio: no

`linuxParameters`

Modificaciones específicas de Linux que se aplican al contenedor, como los detalles de asignación de dispositivos.

```
"linuxParameters": {
    "devices": [
        {
            "hostPath": "string",
            "containerPath": "string",
            "permissions": [
```

```
                "READ", "WRITE", "MKNOD"
            ]
        }
    ],
    "initProcessEnabled": true|false,
    "sharedMemorySize": 0,
    "tmpfs": [
        {
            "containerPath": "string",
            "size": integer,
            "mountOptions": [
                "string"
            ]
        }
    ],
    "maxSwap": integer,
    "swappiness": integer
}
```

Tipo: objeto LinuxParameters

Obligatorio: no

`devices`

Lista de dispositivos asignados en el contenedor. Este parámetro se corresponde con `Devices` en la sección Crear un contenedor de la API remota de Docker y la opción `--device` se corresponde con docker run.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: matriz de objetos Dispositivo

Obligatorio: no

`hostPath`

Ruta en la que se encuentra el dispositivo disponible en la instancia de contenedor del host.

Tipo: cadena

Obligatorio: sí

`containerPath`

Ruta en la que queda expuesto el dispositivo en el contenedor. Si no se especifica, el dispositivo se expone en la misma ruta que la ruta del host.

Tipo: cadena

Obligatorio: no

`permissions`

Permisos del dispositivo en el contenedor. Si no se especifica, los permisos se establecen en `READ`, `WRITE` y `MKNOD`.

Tipo: matriz de cadenas

Obligatorio: no

Valores válidos: `READ`| `WRITE`| `MKNOD`

`initProcessEnabled`

Si es “true” (verdadero), ejecute un proceso de `init` dentro del contenedor que reenvíe señales y aproveche procesos. Este parámetro se corresponde con la opción `--init` de docker run. Este parámetro requiere la versión 1.25 de la API remota de Docker o superior en su instancia de contenedor. Para comprobar la versión de la API remota de Docker en su instancia de contenedor, inicie sesión en su instancia de contenedor y ejecute el comando siguiente: `sudo docker version | grep "Server API version"`

Tipo: Booleano

Obligatorio: no

`maxSwap`

La cantidad total de memoria de intercambio (en MiB) que puede utilizar un trabajo. Este parámetro se traduce en la opción `--memory-swap` de docker run donde el valor es la suma de la memoria del contenedor más el valor de `maxSwap`. Para obtener más información, consulte Detalles de `--memory-swap` en la documentación de Docker.

Si se especifica un valor `maxSwap` para `0`, el contenedor no utiliza el intercambio. Los valores aceptados son `0` o cualquier entero positivo. Si se omite el parámetro `maxSwap`, el

contenedor utiliza la configuración de intercambio de la instancia de contenedor en la que se está ejecutando. Debe establecerse un valor de `maxSwap` para el parámetro `swappiness`.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`sharedMemorySize`

El valor del tamaño (en MiB) del volumen `/dev/shm`. Este parámetro se corresponde con la opción `--shm-size` de docker run.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`swappiness`

Puede utilizar esta opción para ajustar el comportamiento de intercambio de memoria de un contenedor. Un valor `swappiness` de `0` impide que ocurra el intercambio a menos que sea absolutamente necesario. Un valor `swappiness` de `100` aumenta al máximo el intercambio de páginas. Los valores aceptados son números enteros comprendidos entre `0` y `100`. Si no se especifica el parámetro `swappiness`, se utiliza el valor predeterminado de `60`. Si no se especifica ningún valor para `maxSwap`, este parámetro se omite. Si `maxSwap` se establece en 0, el contenedor no utiliza intercambio. Este parámetro se corresponde con la opción `--memory-swappiness` de docker run.

Tenga en cuenta lo siguiente cuando utilice una configuración de intercambio por contenedor.

- El espacio de intercambio debe estar habilitado y asignado a la instancia de contenedor para que los contenedores lo utilicen.

> **Note**
>
> Las AMI optimizadas de Amazon ECS no tienen habilitado el intercambio de forma predeterminada. Debe habilitar el intercambio en la instancia para utilizar esta característica. Para obtener más información, consulte Volúmenes de intercambio de almacenes de instancias en la Guía del usuario de Amazon EC2 para instancias de Linux o ¿Cómo asignar memoria para que funcione como espacio de intercambio en una instancia de Amazon EC2 mediante un archivo de intercambio?

- Los parámetros de espacio de intercambio solo se admiten para definiciones de trabajos que utilizan recursos de EC2.
- Si en una definición de trabajo se omiten los parámetros `maxSwap` y `swappiness`, cada contenedor tendrá un valor `swappiness` predeterminado de 60. El uso total de intercambio se limita al doble de la reserva de memoria del contenedor.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`tmpfs`

La ruta del contenedor, las opciones de montaje y el tamaño del montaje tmpfs.

Tipo: matriz de objetos Tmpfs

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Obligatorio: no

`containerPath`

La ruta de archivo absoluta en el contenedor donde se monta el volumen tmpfs.

Tipo: cadena

Obligatorio: sí

`mountOptions`

La lista de opciones de montaje del volumen tmpfs.

Valores válidos: `"defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"`

Tipo: matriz de cadenas

Obligatorio: no

`size`

El tamaño (en MiB) del volumen tmpfs.

Tipo: entero

Obligatorio: sí

`logConfiguration`

La especificación de configuración de registro para el trabajo.

Este parámetro se corresponde con `LogConfig` en la sección Crear un contenedor de la API remota de Docker y la opción `--log-driver` se corresponde con docker run. De forma predeterminada, los contenedores usan el mismo controlador de registro que el daemon de Docker. No obstante, el contenedor puede usar un controlador de registro distinto del daemon de Docker especificando un controlador de registro con este parámetro en la definición de contenedor. Para utilizar un controlador de registro distinto para un contenedor, el sistema de registro se debe configurar correctamente en la instancia de contenedor o en un servidor de registro distinto para opciones de registro remotas. Para obtener más información acerca de las opciones para los distintos controladores de registro admitidos, consulte Configurar controladores de registro en la documentación de Docker.

> **Note**
>
> AWS Batch actualmente admite un subconjunto de los controladores de registro disponibles para el daemon de Docker (que se muestran en el tipo de datos). LogConfiguration

Este parámetro requiere la versión 1.18 de la API remota de Docker o superior en su instancia de contenedor. Para comprobar la versión de la API remota de Docker en su instancia de contenedor, inicie sesión en su instancia de contenedor y ejecute el comando siguiente: `sudo docker version | grep "Server API version"`

```
"logConfiguration": {
    "devices": [
        {
            "logDriver": "string",
            "options": {
                "optionName1" : "optionValue1",
                "optionName2" : "optionValue2"
            }
            "secretOptions": [
              {
                  "name" : "secretOptionName1",
                  "valueFrom" : "secretOptionArn1"
              },
              {
                  "name" : "secretOptionName2",
                  "valueFrom" : "secretOptionArn2"
              }
            ]
        }
    ]
}
```

Tipo: objeto LogConfiguration

Obligatorio: no

`logDriver`

El controlador de registro que utilizar para el trabajo. De forma predeterminada, AWS Batch habilita el controlador de registro`awslogs`. Los valores válidos que se enumeran para

este parámetro son controladores de registro con los que el agente del contenedor de Amazon ECS se puede comunicar de forma predeterminada.

Este parámetro se corresponde con `LogConfig` en la sección Crear un contenedor de la API remota de Docker y la opción `--log-driver` se corresponde con docker run. De forma predeterminada, los trabajos usan el mismo controlador de registro que el daemon de Docker. No obstante, el trabajo puede usar un controlador de registro distinto del daemon de Docker especificando un controlador de registro con este parámetro en la definición de contenedor. Si desea especificar otro controlador de registro para un trabajo, el sistema de registro se debe configurar en la instancia de contenedor en el entorno de computación. O bien, configúrelo en otro servidor de registro para proporcionar opciones de registro remoto. Para obtener más información acerca de las opciones para los distintos controladores de registro admitidos, consulte Configurar controladores de registro en la documentación de Docker.

> **Note**
>
> AWS Batch actualmente admite un subconjunto de los controladores de registro que están disponibles para el daemon de Docker. Es posible que haya controladores de registro adicionales en versiones futuras del agente de contenedor de Amazon ECS.

Los controladores de registro admitidos son `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `syslog` y `splunk`.

> **Note**
>
> Los trabajos que se ejecutan en recursos de Fargate están restringidos a los controladores de registro `awslogs` y `splunk`.

Este parámetro requiere la versión 1.18 de la API remota de Docker o superior en su instancia de contenedor. Para comprobar la versión de la API remota de Docker en su instancia de contenedor, inicie sesión en su instancia de contenedor y ejecute el comando siguiente: `sudo docker version | grep "Server API version"`

> **Note**
>
> El agente de contenedor de Amazon ECS que se ejecuta en una instancia de contenedor debe registrar los controladores de registro disponibles en dicha instancia

con la variable de entorno `ECS_AVAILABLE_LOGGING_DRIVERS`. De lo contrario, los contenedores ubicados en esa instancia no pueden usar estas opciones de configuración de registro. Para obtener más información, consulte Configuración del agente de contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

`awslogs`

Especifica el controlador de CloudWatch registro de Amazon Logs. Para obtener más información, consulte Uso del controlador de registros awslogs el controlador de registro de Amazon CloudWatch Logs en la documentación de Docker.

`fluentd`

Especifica el controlador de registro Fluentd. Para obtener más información, incluidos el uso y las opciones, consulte Controlador de registro Fluentd en la documentación de Docker.

`gelf`

Especifica el controlador de registro Graylog Extended Format (GELF). Para obtener más información, incluidos el uso y las opciones, consulte Controlador de registro Graylog Extended Format en la documentación de Docker.

`journald`

Especifica el controlador de registro journald. Para obtener más información, incluidos el uso y las opciones, consulte Controlador de registro Journald en la documentación de Docker.

`json-file`

Especifica el controlador de registro de archivos JSON. Para obtener más información, incluidos el uso y las opciones, consulte Controlador de registro de archivos JSON en la documentación de Docker.

`splunk`

Especifica el controlador de registro Splunk. Para obtener más información, incluidos el uso y las opciones, consulte (Controlador de registro Splunk en la documentación de Docker.

`syslog`

Especifica el controlador de registro syslog. Para obtener más información, incluidos el uso y las opciones, consulte Controlador de registro Syslog en la documentación de Docker.

Tipo: cadena

Obligatorio: sí

Valores válidos: `awslogs` | `fluentd` | `gelf` | `journald` | `json-file` | `splunk` | `syslog`

> **Note**
>
> Si tiene un controlador personalizado que no aparece en la lista anterior y le gustaría trabajar con el agente de contenedores de Amazon ECS, puede bifurcar el proyecto del agente de contenedores de Amazon ECS que está disponible en él GitHub y personalizarlo para que funcione con ese controlador. Le recomendamos enviar solicitudes de inserción para los cambios que desea que incluyamos. No obstante, Amazon Web Services actualmente no permite solicitudes que ejecutan copias modificadas de este software.

`options`

Opciones de configuración de registro que enviar a un controlador de registro para el trabajo.

Este parámetro requiere la versión 1.19 de la API remota de Docker o superior en su instancia de contenedor.

Tipo: mapa de cadena a cadena

Obligatorio: no

`secretOptions`

Un objeto que representa el secreto que transferir a la configuración de registro. Para obtener más información, consulte Especificación de información confidencial.

Tipo: matriz de objetos

Obligatorio: no

`name`

El nombre de la opción del controlador de registro que se va a configurar en el trabajo.

Tipo: cadena

Obligatorio: sí

`valueFrom`

El nombre de recurso de Amazon (ARN) del secreto que se va a exponer en la configuración de registro del contenedor. Los valores admitidos son el ARN completo del secreto del Secrets Manager o el ARN completo del parámetro en el almacén de parámetros SSM.

> **Note**
>
> Si el parámetro SSM Parameter Store existe en la Región de AWS misma tarea que está iniciando, puede usar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

Tipo: cadena

Obligatorio: sí

`memory`

Este parámetro ha quedado obsoleto, utilice *`resourceRequirements`* en su lugar.

El número de MiB de memoria que reservar para el trabajo.

Como ejemplo de cómo usar `resourceRequirements`, si la definición de su trabajo contiene una sintaxis similar a la siguiente.

```
"containerProperties": {
  "memory": 512
}
```

La sintaxis equivalente a través de `resourceRequirements` es de la siguiente manera.

```
"containerProperties": {
```

```
  "resourceRequirements": [
    {
      "type": "MEMORY",
      "value": "512"
    }
  ]
}
```

Tipo: entero

Obligatorio: sí

`mountPoints`

Los puntos de montaje para los volúmenes de datos del contenedor. Este parámetro se asigna a `Volumes` en la sección Crear un contenedor de la API remota de Docker y con la opción `--volume` de docker run.

```
"mountPoints": [
               {
                 "sourceVolume": "string",
                 "containerPath": "string",
                 "readOnly": true|false
               }
             ]
```

Tipo: matriz de objetos

Obligatorio: no

`sourceVolume`

El nombre del volumen a montar.

Tipo: cadena

Obligatorio: sí, si se utiliza `mountPoints`.

`containerPath`

La ruta en el contenedor en la que montar el volumen de host.

Tipo: cadena

Obligatorio: sí, si se utiliza `mountPoints`.

`readOnly`

Si este valor es `true`, el acceso del contenedor al volumen es de solo lectura. Si este valor es `false`, el contenedor puede escribir en el volumen.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

`networkConfiguration`

La configuración de red para los trabajos que se ejecutan en recursos de Fargate. Los trabajos que se ejecutan en los recursos de EC2 no deben especificar este parámetro.

```
"networkConfiguration": {
   "assignPublicIp": "string"
}
```

Tipo: matriz de objetos

Obligatorio: no

`assignPublicIp`

Indica si el trabajo tiene una dirección IP pública. Esto es obligatorio si el trabajo requiere acceso a la red saliente.

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

Valor predeterminado: `DISABLED`

`privileged`

Cuando este parámetro es verdadero, al contenedor se le conceden permisos elevados en la instancia de contenedor de host, similares a los de un usuario `root`. Este parámetro se asigna a `Privileged` en la sección Crear un contenedor de la API remota de Docker y con la opción `--privileged` de docker run. Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No lo proporcione ni lo especifique como falso.

```
"privileged": true|false
```

Tipo: Booleano

Obligatorio: no

`readonlyRootFilesystem`

Cuando este parámetro es verdadero, al contenedor se le concede acceso de solo lectura a su sistema de archivos raíz. Este parámetro se asigna a `ReadonlyRootfs` en la sección Crear un contenedor de la API remota de Docker y con la opción `--read-only` de docker run.

```
"readonlyRootFilesystem": true|false
```

Tipo: Booleano

Obligatorio: no

`resourceRequirements`

El tipo y la cantidad de un recurso para asignar a un contenedor. Los recursos admitidos incluyen `GPU`, `MEMORY`, y `VCPU`.

```
"resourceRequirements" : [
  {
    "type": "GPU",
    "value": "number"
  }
]
```

Tipo: matriz de objetos

Obligatorio: no

`type`

El tipo de recurso para asignar a un contenedor. Los recursos admitidos incluyen `GPU`, `MEMORY`, y `VCPU`.

Tipo: cadena

Obligatorio: sí, si se utiliza `resourceRequirements`.

`value`

La cantidad del recurso especificado que se va a reservar para el contenedor. Los valores varían en función del `type` especificado.

type="GPU"

El número de unidades GPU físicas que reservar para el contenedor. El número de unidades GPU reservadas para todos los contenedores de una tarea no puede superar el número de GPU disponibles en el recurso de computación en el que se lanza la tarea.

type="MEMORY"

El límite máximo (en MiB) de memoria a presentar al contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se cancela. Este parámetro se corresponde con `Memory` en la sección Create a container (Crear un contenedor) de la API remota de Docker y la opción `--memory` se corresponde con docker run. Debe especificar al menos 4 MiB de memoria para un trabajo. Es obligatorio, pero se puede especificar en varios lugares para trabajos paralelos de varios nodos (MNP). Debe especificarse para cada nodo al menos una vez. Este parámetro se corresponde con `Memory` en la sección Create a container (Crear un contenedor) de la API remota de Docker y la opción `--memory` se corresponde con docker run.

> **Note**
>
> Si está intentando maximizar la utilización de los recursos proporcionando a las tareas la mayor cantidad de memoria posible para un tipo de instancia determinado, consulte Administración de la memoria de los recursos informáticos de las .

Para los trabajos que se ejecutan en recursos de Fargate, `value` debe coincidir con uno de los valores admitidos. Además, el valor de `VCPU` debe ser uno de los valores compatibles con el valor de memoria correspondiente.

VCPU	MEMORY
0,25 vCPU	512, 1024 y 2048 MiB
0,5 vCPU	1024-4096 MiB en incrementos de 1024 MiB

VCPU	MEMORY
1 vCPU	2048-8192 MiB en incrementos de 1024 MiB
2 vCPU	4096-16 384 MiB en incrementos de 1024 MiB
4 vCPU	8192-30 720 MiB en incrementos de 1024 MiB
8 vCPU	16 384-61 440 MiB en incrementos de 4096 MiB
16 vCPU	32 768-122 880 MiB en incrementos de 8192 MiB

type="VCPU"

El número de vCPU reservado para el trabajo. Este parámetro se corresponde con `CpuShares` en la sección Create a container (Crear un contenedor) de la API remota de Docker y la opción `--cpu-shares` se corresponde con docker run. Cada vCPU es equivalente a 1 024 cuotas de CPU. Para los trabajos que se ejecutan en recursos EC2, debe especificar al menos una vCPU. Es obligatorio, pero se puede especificar en varias ubicaciones. Debe especificarse para cada nodo al menos una vez.

Para los trabajos que se ejecutan en recursos de Fargate, `value` debe coincidir con uno de los valores admitidos, y los valores de `MEMORY` deben ser uno de los valores admitidos para ese valor de vCPU. Los valores admitidos son 0.25, 0.5, 1, 2, 4, 8 y 16.

El valor predeterminado para la cuota de recuento de recursos de vCPU bajo demanda de Fargate es de 6 vCPU. Para más información sobre las cuotas de Fargate, consulte AWS Cuotas de Fargate en Referencia general de Amazon Web Services.

Tipo: cadena

Obligatorio: sí, si se utiliza `resourceRequirements`.

`secrets`

Los secretos del trabajo que se exponen como variables de entorno. Para obtener más información, consulte Especificación de información confidencial.

```
"secrets": [
    {
      "name": "secretName1",
```

```
      "valueFrom": "secretArn1"
    },
    {
      "name": "secretName2",
      "valueFrom": "secretArn2"
    }
    ...
]
```

Tipo: matriz de objetos

Obligatorio: no

`name`

El nombre de la variable de entorno que contiene el secreto.

Tipo: cadena

Obligatorio: sí, si se utiliza `secrets`.

`valueFrom`

El secreto para exponer en el contenedor. Los valores admitidos son el nombre de recurso de Amazon (ARN) completo del secreto del Secrets Manager o el ARN completo del parámetro en el almacén de parámetros SSM.

> **Note**
>
> Si el parámetro SSM Parameter Store existe en el Región de AWS mismo lugar que el trabajo que está iniciando, puede utilizar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

Tipo: cadena

Obligatorio: sí, si se utiliza `secrets`.

`ulimits`

Una lista de valores para `ulimits` a definir en el contenedor. Este parámetro se asigna a `Ulimits` en la sección Crear un contenedor de la API remota de Docker y con la opción `--ulimit` de docker run.

```
"ulimits": [
  {
    "name": string,
    "softLimit": integer,
    "hardLimit": integer
  }
  ...
]
```

Tipo: matriz de objetos

Obligatorio: no

`name`

El valor `type` de `ulimit`.

Tipo: cadena

Obligatorio: sí, si se utiliza `ulimits`.

`hardLimit`

El límite máximo para el tipo de `ulimit`.

Tipo: entero

Obligatorio: sí, si se utiliza `ulimits`.

`softLimit`

El límite flexible para el tipo de `ulimit`.

Tipo: entero

Obligatorio: sí, si se utiliza `ulimits`.

`user`

El nombre de usuario que utilizar dentro del contenedor. Este parámetro se asigna a `User` en la sección Crear un contenedor de la API remota de Docker y con la opción `--user` de docker run.

```
"user": "string"
```

Tipo: cadena

Obligatorio: no

`vcpus`

Este parámetro ha quedado obsoleto, utilice *`resourceRequirements`* en su lugar.

El número de vCPU reservado para el contenedor.

Como ejemplo sobre el uso de `resourceRequirements`, si la definición de su trabajo contiene líneas similares a las siguientes:

```
"containerProperties": {
  "vcpus": 2
}
```

Las líneas equivalentes que utilizan `resourceRequirements` son las siguientes.

```
"containerProperties": {
  "resourceRequirements": [
    {
      "type": "VCPU",
      "value": "2"
    }
  ]
}
```

Tipo: entero

Obligatorio: sí

`volumes`

Al registrar una definición de trabajos, puede especificar una lista de los volúmenes que se pasan al daemon de Docker en una instancia de contenedor. Los siguientes parámetros están permitidos en las propiedades de contenedor:

```
"volumes": [
  {
```

```
    "name": "string",
    "host": {
      "sourcePath": "string"
    },
    "efsVolumeConfiguration": {
      "authorizationConfig": {
        "accessPointId": "string",
        "iam": "string"
      },
      "fileSystemId": "string",
      "rootDirectory": "string",
      "transitEncryption": "string",
      "transitEncryptionPort": number
    }
  }
]
```

`name`

El nombre del volumen. Se admiten hasta 255 letras (mayúsculas y minúsculas), números, guiones y caracteres de subrayado. Se hace referencia a este nombre en el parámetro `sourceVolume` de la definición de contenedor `mountPoints`.

Tipo: cadena

Obligatorio: no

`host`

El contenido del parámetro `host` determina si el volumen de datos persiste en la instancia de contenedor del host y dónde se almacena. Si el parámetro `host` está vacío, el daemon de Docker le asigna una ruta de host al volumen de datos. Sin embargo, no se garantiza que los datos persistan después de que el contenedor asociado deje de funcionar.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: objeto

Obligatorio: no

`sourcePath`

La ruta de la instancia de contenedor del host que se le presenta al contenedor. Si este parámetro está vacío, el daemon de Docker asigna una ruta de host.

Si el parámetro `host` contiene una ubicación de ubicación de archivos `sourcePath`, el volumen de datos persiste en la ubicación especificada en la instancia de contenedor del host hasta que la elimine manualmente. Si el valor `sourcePath` no existe en la instancia de contenedor del host, el daemon de Docker la crea. Si la ubicación existe, el contenido de la carpeta de la ruta de origen se exporta.

Tipo: cadena

Requerido: no

`efsVolumeConfiguration`

Este parámetro se especifica cuando se utiliza un sistema de archivos de Amazon Elastic File System para el almacenamiento de tareas. Para obtener más información, consulte Volúmenes de Amazon EFS.

Tipo: objeto

Obligatorio: no

`authorizationConfig`

Los detalles de configuración de autorización en el sistema de archivos de Amazon EFS.

Tipo: cadena

Requerido: no

`accessPointId`

El ID del punto de acceso de Amazon EFS que se va a utilizar. Si se especifica un punto de acceso, el valor del directorio raíz que se especifica en `EFSVolumeConfiguration` debe omitirse o establecerse en `/`. Esto impone la ruta establecida en el punto de acceso de EFS. Si se utiliza un punto de acceso, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Para obtener más información, consulte Trabajo con puntos de acceso de Amazon EFS en la Guía del usuario de Amazon Elastic File System.

Tipo: cadena

Requerido: no

`iam`

Determina si se debe utilizar la función de IAM de AWS Batch trabajo definida en una definición de trabajo al montar el sistema de archivos Amazon EFS. Si está habilitado, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Si se omite este parámetro, se usa el valor predeterminado de `DISABLED`. Para obtener más información, consulte Uso de puntos de acceso de Amazon EFS.

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

`fileSystemId`

El ID del sistema de archivos de Amazon EFS que se va a usar.

Tipo: cadena

Obligatorio: no

`rootDirectory`

Directorio del sistema de archivos de Amazon EFS que se va a montar como directorio raíz dentro del host. Si se omite este parámetro, se utiliza la raíz del volumen de Amazon EFS. Si especifica `/`, se obtiene el mismo efecto que si se omite este parámetro. La longitud máxima es de 4096 caracteres.

> **Important**
>
> Si se especifica un punto de acceso de EFS en el `authorizationConfig`, se debe omitir el parámetro del directorio raíz o establecerlo en `/`. Esto impone la ruta establecida en el punto de acceso de Amazon EFS.

Tipo: cadena

Obligatorio: no

`transitEncryption`

Determina si se habilita el cifrado de los datos en tránsito de Amazon EFS entre el host de Amazon ECS y el servidor de Amazon EFS. El cifrado en tránsito debe estar habilitado si se utiliza la autorización de IAM en Amazon EFS. Si se omite este parámetro, se usa el valor predeterminado de `DISABLED`. Para obtener más información, consulte Cifrado de datos en tránsito en la Guía del usuario de Amazon Elastic File System.

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

`transitEncryptionPort`

El puerto que se utilizará al enviar datos cifrados entre el host de Amazon ECS y el servidor de Amazon EFS. Si no se especifica un puerto de cifrado en tránsito, se emplea la estrategia de selección de puertos que utiliza el ayudante de montaje de Amazon EFS. Este valor debe estar entre 0 y 65 535. Para obtener más información, consulte Ayudante de montaje de EFS en la Guía del usuario de Amazon Elastic File System.

Tipo: entero

Obligatorio: no

Propiedades de Amazon EKS

Un objeto con diversas propiedades específicas para los trabajos basados en Amazon EKS. No se puede especificar para definiciones de trabajo basadas en Amazon ECS.

`podProperties`

Las propiedades de los recursos del pod de Kubernetes de un trabajo.

Tipo: objeto EksPodProperties

Obligatorio: no

`containers`

Las propiedades del contenedor que se utiliza en el pod de Amazon EKS.

Tipo: objeto EksContainer

Obligatorio: no

`args`

Una matriz de argumentos para el punto de entrada. Si no se especifica, se utiliza el `CMD` de la imagen de contenedor. Esto corresponde al miembro `args` en la parte Punto de entrada del Pod en Kubernetes. Las referencias de variables de entorno se expanden mediante el entorno del contenedor.

Si la variable de entorno a la que se hace referencia no existe, la referencia en el comando no cambia. Por ejemplo, si la referencia es "`$(NAME1)`" y la variable de entorno `NAME1` no existe, la cadena de comando seguirá siendo "`$(NAME1)`". `$$` se reemplaza por `$`, y la cadena resultante no se expande. Por ejemplo, `$$(VAR_NAME)` se pasa como `$(VAR_NAME)`, ya sea que existe o no la variable de entorno `VAR_NAME`. Para obtener más información, consulte CMD en la Referencia de Dockerfile y Definición de un comando y los argumentos para un pod en la documentación de Kubernetes.

Tipo: matriz de cadenas

Obligatorio: no

`command`

El punto de entrada para el contenedor. Esto no se ejecuta dentro de un intérprete de comandos. Si no se especifica, se utiliza el `ENTRYPOINT` de la imagen de contenedor. Las referencias de variables de entorno se expanden mediante el entorno del contenedor.

Si la variable de entorno a la que se hace referencia no existe, la referencia en el comando no cambia. Por ejemplo, si la referencia es "`$(NAME1)`" y la variable de entorno `NAME1` no existe, la cadena de comando seguirá siendo "`$(NAME1)`". `$$` se reemplaza por `$`, y la cadena resultante no se expande. Por ejemplo, `$$(VAR_NAME)` se pasará como `$(VAR_NAME)`, ya sea que existe o no la variable de entorno `VAR_NAME`. El punto de entrada no se puede actualizar. Para obtener más información, consulte PUNTO DE ENTRADA en la Referencia de Dockerfile y Definición de un comando y los argumentos para un contenedor y Punto de entrada en la documentación de Kubernetes.

Tipo: matriz de cadenas

Obligatorio: no

`env`

Las variables de entorno a transferir a un contenedor.

> **Note**
>
> Las variables de entorno no pueden empezar por “AWS_BATCH”. Esta convención de nomenclatura está reservada para las variables que se AWS Batch establecen.

Tipo: matriz de objetos EksContainerEnvironmentVariable

Obligatorio: no

`name`

El nombre de la variable de entorno.

Tipo: cadena

Obligatorio: sí

`value`

El valor de la variable de entorno.

Tipo: cadena

Obligatorio: no

`image`

La imagen de Docker que se utiliza para iniciar el contenedor.

Tipo: cadena

Obligatorio: sí

`imagePullPolicy`

La política de extracción de imágenes para el contenedor. Los valores admitidos son `Always`, `IfNotPresent` y `Never`. El valor predeterminado de este parámetro es `IfNotPresent`. Sin embargo, si se especifica la etiqueta `:latest`, el valor predeterminado es `Always`. Para obtener más información, consulte Actualización de imágenes en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: no

`name`

El nombre del contenedor. Si no se especifica el nombre, el nombre predeterminado es "`Default`". Cada contenedor de un pod debe tener un nombre único.

Tipo: cadena

Obligatorio: no

`resources`

El tipo y la cantidad de recursos para asignar a un contenedor. Los recursos admitidos incluyen `memory`, `cpu`, y `nvidia.com/gpu`. Para obtener más información, consulte Administración de recursos para pods y contenedores en la documentación de Kubernetes.

Tipo: objeto EksContainerResourceRequirements

Obligatorio: no

`limits`

El tipo y la cantidad de recursos que se reservan para el contenedor. Los valores varían en función del `name` que se especifica. Los recursos se pueden solicitar mediante los `limits` o los objetos `requests`.

memoria

El límite máximo de memoria (en MiB) para el contenedor, mediante números enteros, con un sufijo "Mi". Si su contenedor intenta superar la memoria especificada aquí, se cancela el contenedor. Debe especificar al menos 4 MiB de memoria para un trabajo. `memory` se puede especificar en `limits`, `requests` o en ambos. Si `memory` se especifica en ambos lugares, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

> **Note**
>
> Para maximizar el uso de los recursos, proporcione a los trabajos la mayor cantidad de memoria posible para el tipo de instancia específico que se está utilizando. Para saber cómo hacerlo, consulte Administración de la memoria de los recursos informáticos de las .

cpu

La cantidad de CPU reservadas para el contenedor. Los valores deben ser múltiplos pares de `0.25`. `cpu` se puede especificar en `limits`, `requests` o en ambos. Si `cpu` se especifica en ambos lugares, el valor especificado en `limits` debe ser tan grande como el valor especificado en `requests`.

nvidia.com/gpu

La cantidad de GPU reservadas para el contenedor. Los valores deben ser números enteros. `memory` se puede especificar en `limits`, `requests` o en ambos. Si `memory` se especifica en ambos lugares, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

Tipo: mapa de cadena a cadena

Limitaciones de longitud de los valores: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

`requests`

El tipo y la cantidad de recursos que se solicitan para el contenedor. Los valores varían en función del `name` que se especifica. Los recursos se pueden solicitar mediante los `limits` o los objetos `requests`.

memoria

El límite máximo de memoria (en MiB) para el contenedor, mediante números enteros, con un sufijo “Mi”. Si su contenedor intenta superar la memoria especificada aquí, se cancela el contenedor. Debe especificar al menos 4 MiB de memoria para un trabajo. `memory` se puede especificar en `limits`, `requests` o en ambos. Si `memory` se especifica en ambos, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

> **Note**
>
> Si está intentando maximizar la utilización de los recursos proporcionando a las tareas la mayor cantidad de memoria posible para un tipo de instancia determinado, consulte Administración de la memoria de los recursos informáticos de las .

cpu

La cantidad de CPU reservadas para el contenedor. Los valores deben ser múltiplos pares de `0.25`. `cpu` se puede especificar en `limits`, `requests` o en ambos. Si `cpu` se especifica en ambos, el valor especificado en `limits` debe ser tan grande como el valor especificado en `requests`.

nvidia.com/gpu

La cantidad de GPU reservadas para el contenedor. Los valores deben ser números enteros. `nvidia.com/gpu` se puede especificar en `limits`, `requests` o en ambos. Si `nvidia.com/gpu` se especifica en ambos, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

Tipo: mapa de cadena a cadena

Limitaciones de longitud de los valores: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

`securityContext`

El contexto de seguridad para un trabajo. Para obtener más información, consulte Configuración de un contexto de seguridad para un pod o un contenedor en la documentación de Kubernetes.

Tipo: objeto EksContainerSecurityContext

Obligatorio: no

`privileged`

Cuando este parámetro es `true`, al contenedor se le conceden permisos elevados en la instancia de contenedor de host. El nivel de permisos es similar al de los permisos de usuario `root`. El valor predeterminado es `false`. Este parámetro se asigna a la política `privileged` de Políticas de seguridad de pod privilegiados en la documentación de Kubernetes.

Tipo: Booleano

Obligatorio: no

`readOnlyRootFilesystem`

Cuando este parámetro es `true`, al contenedor se le concede acceso de solo lectura al sistema de archivos raíz. El valor predeterminado es `false`. Este parámetro se asigna a la política `ReadOnlyRootFilesystem` de Políticas de seguridad de pods de sistemas de archivos y volúmenes en la documentación de Kubernetes.

Tipo: Booleano

Obligatorio: no

`runAsGroup`

Cuando se especifica este parámetro, el contenedor se ejecuta como el ID del grupo especificado (`gid`). Si no se especifica este parámetro, el valor predeterminado es el grupo que se especifica en los metadatos de la imagen. Este parámetro se asigna a las políticas `RunAsGroup` y `MustRunAs` de Políticas de seguridad de pods de grupos y usuarios en la documentación de Kubernetes.

Tipo: largo

Obligatorio: no

`runAsNonRoot`

Cuando se especifica este parámetro, el contenedor se ejecuta como un usuario con un `uid` distinto de 0. Si no se especifica este parámetro, se aplica la regla mencionada. Este parámetro se asigna a las políticas `RunAsUser` y `MustRunAsNonRoot` de Políticas de seguridad de pods de grupos y usuarios en la documentación de Kubernetes.

Tipo: largo

Obligatorio: no

`runAsUser`

Cuando se especifica este parámetro, el contenedor se ejecuta como el ID del usuario especificado (`uid`). Si no se especifica este parámetro, el valor predeterminado es el usuario que se especifica en los metadatos de la imagen. Este parámetro se asigna a las políticas `RunAsUser` y `MustRanAs` de Políticas de seguridad de pods de grupos y usuarios en la documentación de Kubernetes.

Tipo: largo

Obligatorio: no

`volumeMounts`

El volumen se monta para un contenedor para un trabajo de Amazon EKS. Para obtener más información sobre los volúmenes y los montajes de volúmenes en Kubernetes, consulte Volúmenes en la documentación de Kubernetes.

Tipo: matriz de objetos EksContainerVolumeMount

Obligatorio: no

`mountPath`

La ruta en el contenedor donde se monta el volumen.

Tipo: cadena

Obligatorio: no

`name`

El nombre del montaje de volumen. Debe coincidir con el nombre de uno de los volúmenes del pod.

Tipo: cadena

Obligatorio: no

`readOnly`

Si este valor es `true`, el acceso del contenedor al volumen es de solo lectura. De lo contrario, el contenedor puede escribir en el volumen. El valor predeterminado es `false`.

Tipo: Booleano

Obligatorio: no

`dnsPolicy`

La política de DNS del pod. El valor predeterminado es `ClusterFirst`. Si no se especifica el parámetro `hostNetwork`, el valor predeterminado es `ClusterFirstWithHostNet`. `ClusterFirst` indica que cualquier consulta de DNS que no coincida con el sufijo

de dominio del clúster configurado se reenvía al servidor de nombres ascendente heredado del nodo. Si no se especificó ningún valor `dnsPolicy` en la operación de la RegisterJobDefinitionAPI, ninguna de las operaciones de la DescribeJobsAPI devolverá DescribeJobDefinitionsningún valor. `dnsPolicy` La configuración de las especificaciones del pod contendrá `ClusterFirst` o `ClusterFirstWithHostNet`, según el valor del parámetro `hostNetwork`. Para obtener más información, consulte Política de DNS del pod en la documentación de Kubernetes.

Valores válidos: `Default`| `ClusterFirst`| `ClusterFirstWithHostNet`

Tipo: cadena

Obligatorio: no

`hostNetwork`

Indica si el pod utiliza la dirección IP de red de los hosts. El valor predeterminado es `true`. Cuando se establece en `false`, se activa el modelo de red de pods de Kubernetes. La mayoría AWS Batch de las cargas de trabajo son solo de salida y no requieren la sobrecarga de asignación de IP para cada pod para las conexiones entrantes. Para obtener más información, consulte Espacios de nombres de hosts y Redes de pods en la documentación de Kubernetes.

Tipo: Booleano

Obligatorio: no

`serviceAccountName`

El nombre de la cuenta de servicio que se utiliza para ejecutar el pod. Para obtener más información, consulte Cuentas de servicio de Kubernetes y Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM en la Guía del usuario de Amazon EKS y Configuración de cuentas de servicio para pods en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: no

`volumes`

Especifica los volúmenes para una definición de trabajo que utiliza recursos de Amazon EKS.

Tipo: matriz de objetos EksVolume

Obligatorio: no

emptyDir

Especifica la configuración de un volumen de Kubernetes `emptyDir`. Un volumen `emptyDir` se crea por primera vez cuando se asigna un pod a un nodo. Existe mientras ese pod se ejecuta en ese nodo. Al principio, el volumen `emptyDir` está vacío. Todos los contenedores del pod pueden leer y escribir los archivos del volumen `emptyDir`. Sin embargo, el volumen `emptyDir` se puede montar en la misma ruta o en rutas diferentes en cada contenedor. Cuando por algún motivo se elimina un pod de un nodo, los datos de `emptyDir` se eliminan de manera permanente. Para obtener más información, consulte emptyDir en la documentación de Kubernetes.

Tipo: objeto EksEmptyDir

Obligatorio: no

medium

El medio para almacenar el volumen. El valor predeterminado es una cadena vacía, que utiliza el almacenamiento del nodo.

""

(Predeterminado) Utilice el almacenamiento en disco del nodo.

“Memoria”

Utilice el volumen `tmpfs` respaldado por la RAM del nodo. El contenido del volumen se pierde cuando el nodo se reinicia, y cualquier almacenamiento en el volumen cuenta respecto del límite de memoria del contenedor.

Tipo: cadena

Obligatorio: no

sizeLimit

El tamaño máximo del volumen. De forma predeterminada, no hay un tamaño máximo definido.

Tipo: string

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

hostPath

Especifica la configuración de un volumen de Kubernetes `hostPath`. Un volumen `hostPath` monta un archivo o un directorio existente del sistema de archivos del nodo host en el pod. Para obtener más información, consulte hostPath en la documentación de Kubernetes.

Tipo: objeto EksHostPath

Obligatorio: no

path

La ruta del archivo o del directorio en el host para montar en contenedores en el pod.

Tipo: cadena

Obligatorio: no

name

El nombre del volumen. El nombre debe estar permitido como nombre de subdominio de DNS. Para obtener más información, consulte Nombres de subdominio de DNS en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: sí

secreta

Especifica la configuración de un volumen de Kubernetes `secret`. Para obtener más información, consulte secreto en la documentación de Kubernetes.

Tipo: objeto EksSecret

Obligatorio: no

opcional

Especifica si se deben definir el secreto o las claves del secreto.

Tipo: Booleano

Obligatorio: no

secretName

El nombre del secreto. El nombre debe estar permitido como nombre de subdominio de DNS. Para obtener más información, consulte Nombres de subdominio de DNS en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: sí

Capacidades de plataforma

`platformCapabilities`

Las capacidades de plataforma requeridas por la definición de trabajo. Si no se especifica ningún valor, el valor predeterminado es `EC2`. Para los trabajos que se ejecutan en recursos de Fargate, se especifica `FARGATE`.

> **Note**
>
> Si el trabajo se ejecuta en recursos de Amazon EKS, entonces no debe especificar `platformCapabilities`.

Tipo: cadena

Valores válidos: `EC2` | `FARGATE`

Obligatorio: no

Propagar etiquetas

`propagateTags`

Especifica si se van a propagar las etiquetas desde el trabajo o la definición del trabajo a la tarea de Amazon ECS correspondiente. Si no se especifica ningún valor, las etiquetas no se propagan. Las etiquetas solo se pueden propagar a las tareas cuando se crea la tarea. En el caso de etiquetas con el mismo nombre, las etiquetas de trabajo tienen prioridad sobre las etiquetas de

definiciones de trabajo. Si el número total de etiquetas combinadas del trabajo y la definición del trabajo es superior a 50, el trabajo se mueve al estado `FAILED`.

> **Note**
>
> Si el trabajo se ejecuta en recursos de Amazon EKS, entonces no debe especificar `propagateTags`.

Tipo: Booleano

Obligatorio: no

Propiedades del nodo

`nodeProperties`

Al registrar una definición de trabajo paralelo de varios nodos, debe especificar una lista de propiedades de nodos. Estas propiedades de los nodos definen el número de nodos que se van a utilizar en el trabajo, el índice de nodos principales y los diferentes rangos de nodos que se van a utilizar. Si el trabajo se ejecuta en recursos de Fargate, no puede especificar `nodeProperties`. En su lugar, utilice `containerProperties`. En una definición de trabajo, se permiten las siguientes propiedades de nodo. Para obtener más información, consulte Trabajos paralelos de varios nodos.

> **Note**
>
> Si el trabajo se ejecuta en recursos de Amazon EKS, entonces no debe especificar `nodeProperties`.

Tipo: objeto NodeProperties

Obligatorio: no

`mainNode`

Especifica el índice de nodo del nodo principal de un trabajo paralelo de varios nodos. Este valor de índice de nodo debe ser más pequeño que la cantidad de nodos.

Tipo: entero

Obligatorio: sí

`numNodes`

La cantidad de nodos que están asociados a un trabajo paralelo de varios nodos.

Tipo: entero

Obligatorio: sí

`nodeRangeProperties`

Una lista de rangos de nodos y sus propiedades que están asociados a un trabajo paralelo de varios nodos.

> **Note**
>
> Un grupo de nodos es un conjunto de nodos de trabajo idénticos que comparten las mismas propiedades de contenedor. Se puede utilizar AWS Batch para especificar hasta cinco grupos de nodos distintos para cada trabajo.

Tipo: matriz de objetos NodeRangeProperty

Obligatorio: sí

`targetNodes`

El rango de nodos, utilizando valores de índice de nodo. Un rango de `0:3` indica nodos con valores de índice de `0` a `3`. Si se omite el valor inicial del rango (`:n`), se utiliza 0 para iniciar el rango. Si se omite el valor final del rango (`n:`), se utiliza el índice de nodo más alto posible para finalizar el rango. Los rangos de nodos acumulativos deben tener en cuenta todos los nodos (`0:n`). Puede anidar rangos de nodos, por ejemplo, `0:10` y `4:5`. Para este caso, las propiedades del rango `4:5` anulan las propiedades de `0:10`.

Tipo: cadena

Obligatorio: no

`container`

Los detalles del contenedor del rango de nodos. Para obtener más información, consulte Propiedades de contenedor.

Tipo: objeto ContainerProperties

Obligatorio: no

Estrategia de reintento

`retryStrategy`

Al registrar una definición de trabajo, se puede especificar una estrategia de reintento para aplicarla a los trabajos fallidos que se envíen con esta definición de trabajo. Cualquier estrategia de reintento que se especifique durante una SubmitJoboperación anula la estrategia de reintento aquí definida. De forma predeterminada, cada trabajo se intenta una vez. Si especifica más de un intento, el trabajo se vuelve a intentar si falla. Algunos ejemplos de intentos fallidos son que el trabajo devuelva un código de salida distinto de cero o que la instancia de contenedor se termine. Para obtener más información, consulte Reintentos automáticos de trabajo.

Tipo: objeto RetryStrategy

Obligatorio: no

`attempts`

La cantidad de veces que toma pasar un trabajo al estado `RUNNABLE`. Puede especificar entre 1 y 10 intentos. Si `attempts` es mayor que uno y el envío del trabajo falla, se reintenta tantas veces como se haya especificado hasta que pase a `RUNNABLE`.

```
"attempts": integer
```

Tipo: entero

Obligatorio: no

`evaluateOnExit`

Matriz de hasta 5 objetos que especifican las condiciones según las que el trabajo se vuelve a intentar o falla. Si se especifica este parámetro, entonces también debe especificarse el parámetro `attempts`. Si `evaluateOnExit` se especifica pero ninguna de las entradas coincide, se vuelve a intentar el trabajo.

```
"evaluateOnExit": [
```

```
   {
      "action": "string",
      "onExitCode": "string",
      "onReason": "string",
      "onStatusReason": "string"
   }
]
```

Tipo: matriz de objetos EvaluateOnExit

Obligatorio: no

`action`

Especifica la acción que se debe realizar si se cumplen todas las condiciones especificadas (`onStatusReason`, `onReason` y `onExitCode`). Los valores no distinguen entre mayúsculas y minúsculas.

Tipo: cadena

Obligatorio: sí

Valores válidos: `RETRY` | `EXIT`

`onExitCode`

Contiene un patrón global para que coincida con la representación decimal del `ExitCode` devuelto para un trabajo. El patrón puede tener 512 caracteres como máximo. Solo puede contener números. No puede contener letras ni caracteres especiales. Opcionalmente, puede terminar con un asterisco (*) para que solo el inicio de la cadena tenga que ser una coincidencia exacta.

Tipo: cadena

Obligatorio: no

`onReason`

Contiene un patrón global para que coincida con el `Reason` devuelto para un trabajo. El patrón puede tener 512 caracteres como máximo. Puede tener letras, números, puntos (.), dos puntos (:) y espacios en blanco (espacios, pestañas). Opcionalmente, puede terminar con un asterisco (*) para que solo el inicio de la cadena tenga que ser una coincidencia exacta.

Tipo: cadena

Obligatorio: no

`onStatusReason`

Contiene un patrón global para que coincida con el `StatusReason` devuelto para un trabajo. El patrón puede tener 512 caracteres como máximo. Puede tener letras, números, puntos (.), dos puntos (:) y espacios en blanco (espacios, pestañas). Opcionalmente, puede terminar con un asterisco (*) para que solo el inicio de la cadena tenga que ser una coincidencia exacta.

Tipo: cadena

Obligatorio: no

Prioridad de programación

`schedulingPriority`

La prioridad de programación de los trabajos que se enviaron con esta definición de trabajo. Esto solo afecta a las colas de trabajo con una política de reparto justo. Los trabajos con una prioridad de programación más alta se programan antes que los trabajos con una prioridad de programación más baja.

El valor mínimo admitido es 0 y el valor máximo admitido es 9999.

Tipo: entero

Obligatorio: no

Etiquetas

`tags`

Etiquetas de pares clave-valor para asociarlas a la definición del trabajo. Para obtener más información, consulte Etiquetado de los recursos de AWS Batch.

Tipo: mapa de cadena a cadena

Obligatorio: no

Tiempo de espera

`timeout`

Puede configurar un tiempo de espera para sus trabajos de modo que, si un trabajo dura más tiempo, AWS Batch finalice el trabajo. Para obtener más información, consulte Tiempos de espera de trabajo. Si un trabajo se termina debido a que se ha agotado el tiempo de espera, no se vuelve a intentar. Cualquier configuración de tiempo de espera que se especifique durante una SubmitJoboperación anula la configuración de tiempo de espera definida aquí. Para obtener más información, consulte Tiempos de espera de trabajo.

Tipo: objeto JobTimeout

Obligatorio: no

`attemptDurationSeconds`

La duración en segundos (medida a partir de la marca de tiempo `startedAt` del intento de trabajo) después de la cual AWS Batch termina los trabajos pendientes. El valor mínimo del tiempo de espera es 60 segundos.

En el caso de los trabajos de matriz, el tiempo de espera se aplica a los trabajos secundarios, no al trabajo de matriz principal.

En el caso de los trabajos paralelos de varios nodos (MNP), el tiempo de espera se aplica a todo el trabajo, no a los nodos individuales.

Tipo: entero

Obligatorio: no

Creación de definiciones de trabajo mediante EcsProperties

Si utiliza las definiciones de AWS Batch trabajo `EcsProperties`, puede modelar el hardware, los sensores, los entornos 3D y otras simulaciones en contenedores separados. Puede utilizar esta función para organizar de forma lógica los componentes de la carga de trabajo y separarlos de la aplicación principal. Esta función se puede utilizar AWS Batch en Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) y. AWS Fargate

`ContainerProperties`en comparación con las definiciones de puestos `EcsProperties`

Puede optar por utilizar `ContainerProperties`nuestras definiciones de `EcsProperties`trabajo según lo indique su caso de uso. A un nivel superior, ejecutar AWS Batch trabajos con `EcsProperties` es similar a ejecutar trabajos con un`ContainerProperties`.

La estructura de definición de trabajos heredada, que utilizaba`ContainerProperties`, sigue siendo compatible. Si actualmente tiene flujos de trabajo que utilizan esta estructura, puede seguir ejecutándolos.

La principal diferencia es que se ha añadido un objeto nuevo a la definición del trabajo para adaptarlo a las definiciones `EcsProperties` basadas.

Por ejemplo, una definición de trabajo que se usa `ContainerProperties` en Amazon ECS y Fargate tiene la siguiente estructura:

```
{
   "containerProperties": {
     ...
     "image": "my_ecr_image1",
     ...
  },
...
}
```

Una definición de trabajo que se usa `EcsProperties` en Amazon ECS y Fargate tiene la siguiente estructura:

```
{
  "ecsProperties": {
    "taskProperties": [{
      "containers": [
        {
          ...
          "image": "my_ecr_image1",
          ...
        },
        {
          ...
          "image": "my_ecr_image2",
```

```
        ...
    },
```

Cambios generales en las API AWS Batch

A continuación se describen con más detalle algunas de las principales diferencias entre el uso de los tipos de datos `EcsProperties` y los de la `EcsProperties` API:

- Muchos de los parámetros que se utilizan dentro `ContainerProperties` aparecen dentro`TaskContainerProperties`. Algunos ejemplos incluyen`command`,`image`, `privileged`, `secrets`, y`users`. Todos se pueden encontrar dentro TaskContainerProperties.
- Algunos de los `TaskContainerProperties` parámetros no tienen equivalentes funcionales en la estructura heredada. Algunos ejemplos incluyen`dependsOn`,`essential`, `name`, `ipcMode`, y`pidMode`. Para obtener más información, consulte EcsTaskDetailsy TaskContainerProperties.

  Además, algunos `ContainerProperties` parámetros no tienen equivalentes ni aplicaciones en la `EcsProperties` estructura. En `taskProperties`, se `container` ha sustituido por `containers` para que el nuevo objeto pueda aceptar hasta diez elementos. Para obtener más información, consulte: ContainerProperties y: containersRegisterJobDefinition. EcsTaskProperties
- `taskRoleArn`es funcionalmente `jobRoleArn` equivalente a. Para obtener más información, consulte EcsTaskProperties: taskRoleArn y ContainerProperties: jobRoleArn.
- Puede incluir de uno (1) a diez (10) contenedores en la `EcsProperties` estructura. Para obtener más información, consulte: contenedoresEcsTaskProperties.
- Los objetos InstancEtypes `taskProperties` e InstanceTypes son matrices, pero actualmente solo aceptan un elemento. Por ejemplo, :taskProperties y:instanceTypesEcsProperties. NodeRangeProperty

Definiciones de trabajos con varios contenedores para Amazon ECS

Para adaptarse a la estructura de varios contenedores de Amazon ECS, algunos tipos de datos de la API son diferentes. Por ejemplo:

- `ecsProperties`tiene el mismo nivel que `containerProperties` en la definición de contenedor único. Para obtener más información, consulte la Guía EcsPropertiesde referencia AWS Batch de la API.
- `taskProperties`contiene las propiedades definidas para la tarea Amazon ECS. Para obtener más información, consulte EcsPropertiesla Guía de referencia de la AWS Batch API.

- `containers`incluye información similar a la de `containerProperties` la definición de contenedor único. La principal diferencia es que `containers` permite definir hasta diez contenedores. Para obtener más información, consulte ECS:Containers TaskProperties en la Guía de AWS Batch referencia de la API.
- `essential`Este parámetro indica cómo afecta el contenedor al trabajo. Todos los contenedores esenciales deben completarse correctamente (salir como 0) para que el trabajo avance. Si un contenedor marcado como esencial falla (sale con un valor distinto de 0), el trabajo fallará.

  El valor predeterminado es `true` y al menos un contenedor debe estar marcado como. `essential` Para obtener más información, consulte `essential` en la Guía de referencia de la API de AWS Batch .
- Con el `dependsOn`parámetro, puede definir una lista de dependencias de contenedores. Para obtener más información, consulte `dependsOn` en la Guía de referencia de la API de AWS Batch .

> **Note**
>
> La complejidad de la `dependsOn` lista y el tiempo de ejecución del contenedor asociado pueden afectar a la hora de inicio del trabajo. Si las dependencias tardan mucho en ejecutarse, el trabajo permanecerá en ese `STARTING` estado hasta que se complete.

Para obtener más información sobre la estructura `ecsProperties` y, consulte la sintaxis de RegisterJobDefinitionsolicitud de ECSProperties.

Definiciones de trabajos con varios contenedores para Amazon EKS

Para adaptarse a la estructura de varios contenedores de Amazon EKS, algunos tipos de datos de la API son diferentes. Por ejemplo:

- `name`es un identificador único del contenedor. Este objeto no es necesario para un único contenedor, pero sí para definir varios contenedores en un pod. Si `name` no está definido para contenedores individuales, se aplica el nombre predeterminado`default`,.
- `initContainers`se definen dentro del tipo eksPodPropertiesde datos. Se ejecutan antes que los contenedores de aplicaciones, siempre se ejecutan hasta completarse y deben completarse correctamente antes de que se inicie el siguiente contenedor.

  Estos contenedores están registrados en el agente Amazon EKS Connector y conservan la información de registro en el almacén de datos de backend de Amazon Elastic Kubernetes

Service. El `initContainers` objeto puede aceptar hasta diez (10) elementos. Para obtener más información, consulte Init Containers en la Kubernetesdocumentación.

> **Note**
>
> El `initContainers` objeto puede afectar a la hora de inicio del trabajo. Si `initContainers` tardan mucho en ejecutarse, el trabajo permanecerá en ese `STARTING` estado hasta que se complete.

- `shareProcessNamespace`indica si los contenedores del pod pueden compartir el mismo espacio de nombres del proceso. El valor predeterminado es `false`. Configurarlo `true` para permitir que los contenedores vean y señalen los procesos de otros contenedores que se encuentran en el mismo pod.
- Cada contenedor es importante. Todos los contenedores deben completarse correctamente (salir como 0) para que el trabajo se realice correctamente. Si un contenedor falla (sale con un valor distinto de 0), se produce un error en el trabajo.

Para obtener más información sobre la estructura `eksProperties` y, consulte la sintaxis de RegisterJobDefinitionsolicitud de eksProperties.

AWS Batch escenarios de trabajo utilizando EcsProperties

Para ilustrar cómo se `EcsProperties` pueden estructurar las definiciones de AWS Batch trabajo que se utilizan en función de sus necesidades, en este tema se presentan las siguientes `RegisterJobDefinition`cargas útiles. Puede copiar estos ejemplos en un archivo, personalizarlos según sus necesidades y, a continuación, usar el AWS Command Line Interface (AWS CLI) para llamar`RegisterJobDefinition`.

AWS Batch trabajo para Amazon Elastic Container Service en Amazon Elastic Compute Cloud

```
{
    "jobDefinitionName": "multicontainer-ecs-ec2",
    "type": "container",
    "ecsProperties": {
        "taskProperties": [
          {
            "containers": [
```

```
            {
              "name": "c1",
              "essential": false,
              "command": [
                "echo",
                "hello world"
              ],
              "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
              "resourceRequirements": [
                {
                  "type": "VCPU",
                  "value": "2"
                },
                {
                  "type": "MEMORY",
                  "value": "4096"
                }
              ]
            },
            {
              "name": "c2",
              "essential": true,
              "command": [
                "echo",
                "hello world"
              ],
              "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
              "resourceRequirements": [
                {
                  "type": "VCPU",
                  "value": "6"
                },
                {
                  "type": "MEMORY",
                  "value": "12288"
                }
              ]
            }
          ]
        }
      ]
  }
}
```

AWS Batch trabajo para Amazon ECS en AWS Fargate

```
{
    "jobDefinitionName": "multicontainer-ecs-fargate",
    "type": "container",
    "platformCapabilities": [
        "FARGATE"
    ],
    "ecsProperties": {
        "taskProperties": [
          {
            "containers": [
              {
                "name": "c1",
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "2"
                  },
                  {
                    "type": "MEMORY",
                    "value": "4096"
                  }
                ]
              },
              {
                "name": "c2",
                "essential": true,
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "6"
                  },
```

```
                {
                  "type": "MEMORY",
                  "value": "12288"
                }
              ]
            }
          ],
          "executionRoleArn": "arn:aws:iam::1112223333:role/ecsTaskExecutionRole"
        }
      ]
  }
}
```

AWS Batch trabajo para Amazon Elastic Kubernetes Service

```
{
  "jobDefinitionName": "multicontainer-eks",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "shareProcessNamespace": true,
      "initContainers": [
        {
          "name": "init-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo"
          ],
          "args": [
            "hello world"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        },
        {
          "name": "init-container-2",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo",
```

```
            "my second init container"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        }
      ],
      "containers": [
        {
          "name": "c1",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo world"
         ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        },
        {
          "name": "sleep-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "20"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        }
      ]
    }
  }
}
```

AWS Batch Trabajo en paralelo de varios nodos (MNP) con varios contenedores por nodo

```
{
  "jobDefinitionName": "multicontainer-mnp",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 6,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:5",
        "ecsProperties": {
          "taskProperties": [
            {
              "containers": [
                {
                  "name": "range05-c1",
                  "command": [
                    "echo",
                    "hello world"
                  ],
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                  "resourceRequirements": [
                    {
                      "type": "VCPU",
                      "value": "2"
                    },
                    {
                      "type": "MEMORY",
                      "value": "4096"
                    }
                  ]
                },
                {
                  "name": "range05-c2",
                  "command": [
                    "echo",
                    "hello world"
                  ],
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                  "resourceRequirements": [
                    {
```

```
                    "type": "VCPU",
                    "value": "2"
                  },
                  {
                    "type": "MEMORY",
                    "value": "4096"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Uso del controlador de registros awslogs

De forma predeterminada, AWS Batch permite que el controlador de registro `awslogs` envíe información de registro a CloudWatch Logs. Puede usar esta característica para ver distintos registros desde sus contenedores en una ubicación cómoda y evitar que los registros de contenedor ocupen espacio en disco en sus instancias de contenedor. Este tema le ayuda a configurar el controlador de registro `awslogs` en sus definiciones de trabajo.

> **Note**
>
> En la consola de AWS Batch, puede configurar el controlador de registro `awslogs` en la sección Configuración de registros al crear una definición de trabajo.

> **Note**
>
> El tipo de información que registran los contenedores del trabajo depende en gran medida del comando `ENTRYPOINT`. De forma predeterminada, los registros que se capturan muestran la salida del comando que aparece normalmente en un terminal interactivo si el contenedor se ejecutara localmente, que son los flujos de E/S `STDOUT` y `STDERR`. El controlador del registros `awslogs` simplemente transfiere estos registros de Docker

a CloudWatch Logs. Para obtener más información acerca de cómo se procesan los registros de Docker, incluidas formas alternativas de capturar diferentes datos de archivo o flujos, consulte la página sobre cómo Ver los registros de un contenedor o servicio en la documentación de Docker.

Para enviar registros del sistema desde las instancias de contenedor a CloudWatch Logs, consulte Uso de CloudWatch Logs con AWS Batch. Para obtener más información acerca de CloudWatch Logs, consulte Monitoreo de archivos de registro y Cuotas de CloudWatch Logs en la Guía del usuario de Amazon CloudWatch Logs.

Opciones disponibles del controlador de registros awslogs

El controlador de registros `awslogs` admite las opciones siguientes en las definiciones de trabajo de AWS Batch. Para obtener más información, consulte Controladores de registro CloudWatch logs en la documentación de Docker.

`awslogs-region`

Obligatorio: no

Especifique la región a la que el controlador de registros `awslogs` debe enviarle sus registros de Docker. De forma predeterminada, la región que se utiliza es la misma que la del trabajo. Puede elegir enviar todos los registros desde trabajos de diferentes regiones a una única región de CloudWatch Logs. De este modo, podrá verlos todos desde una misma ubicación. Como alternativa, puede separarlos por región para obtener un enfoque más detallado. Sin embargo, cuando elija esta opción, asegúrese de que los grupos de registro especificados existan en la región que especifique.

`awslogs-group`

Obligatorio: opcional

Con la opción de `awslogs-group`, puede especificar el grupo de registro al que el controlador de registro `awslogs` envía sus flujos de registro. Si esto campo no está especificado, se utiliza `aws/batch/job`.

`awslogs-stream-prefix`

Obligatorio: opcional

Utilice esta opción de `awslogs-stream-prefix` para asociar un flujo de registro con el prefijo especificado, y el ID del trabajo de Amazon ECS AWS Batch al que pertenece el contenedor. Si especifica un prefijo con esta opción, entonces el flujo de registros adopta el siguiente formato:

```
prefix-name/default/ecs-task-id
```

`awslogs-datetime-format`

Obligatorio: no

Esta opción define un patrón de inicio de varias líneas en formato `strftime` de Python. Un mensaje de registro consta de una línea que coincide con el patrón y de líneas siguientes que no coinciden con el patrón. Por tanto, la línea coincidente es el delimitador entre los mensajes de registro.

Un ejemplo de caso de uso de este formato es para analizar la salida como un volcado de pila, que de lo contrario podría registrarse en varias entradas. El patrón correcto permite capturarla en una sola entrada.

Para obtener más información, consulte awslogs-datetime-format.

Esta opción siempre tiene prioridad si ambos `awslogs-datetime-format` y `awslogs-multiline-pattern` están configurados.

> **Note**
>
> El registro de varias líneas realiza análisis de las expresiones regulares y correspondencia de todos los mensajes de registro. Esto puede tener un impacto negativo sobre el rendimiento de registro.

`awslogs-multiline-pattern`

Obligatorio: no

Esta opción define un patrón de inicio de varias líneas con una expresión regular. Un mensaje de registro consta de una línea que coincide con el patrón y de líneas siguientes que no coinciden con el patrón. Por tanto, la línea coincidente es el delimitador entre los mensajes de registro.

Para obtener más información, consulte awslogs-multiline-pattern en la documentación de Docker.

Esta opción se pasa por alto si también se ha configurado `awslogs-datetime-format`.

> **Note**
>
> El registro de varias líneas realiza análisis de las expresiones regulares y correspondencia de todos los mensajes de registro. Esto puede tener un impacto negativo sobre el rendimiento de registro.

`awslogs-create-group`

Obligatorio: no

Especifique si desea que el grupo de registros se cree automáticamente. Si no se especifica esta opción, el valor predeterminado es `false`.

> **Warning**
>
> No se recomienda esta opción. Le recomendamos que cree el grupo de registros con antelación mediante la acción de la API CreateLogGroup de CloudWatch Logs, ya que cada trabajo intenta crear el grupo de registros, lo que aumenta la probabilidad de que el trabajo falle.

> **Note**
>
> Su política de IAM para la ejecución de rol debe incluir el permiso `logs:CreateLogGroup` antes de intentar utilizar `awslogs-create-group`.

Especificación de una configuración de registro en la definición de trabajo

De forma predeterminada, AWS Batch habilita el controlador de registro `awslogs`. En esta sección, se describe cómo personalizar la configuración de registro `awslogs`de un trabajo. Para obtener más información, consulte Creación de una definición de trabajo de un solo nodo .

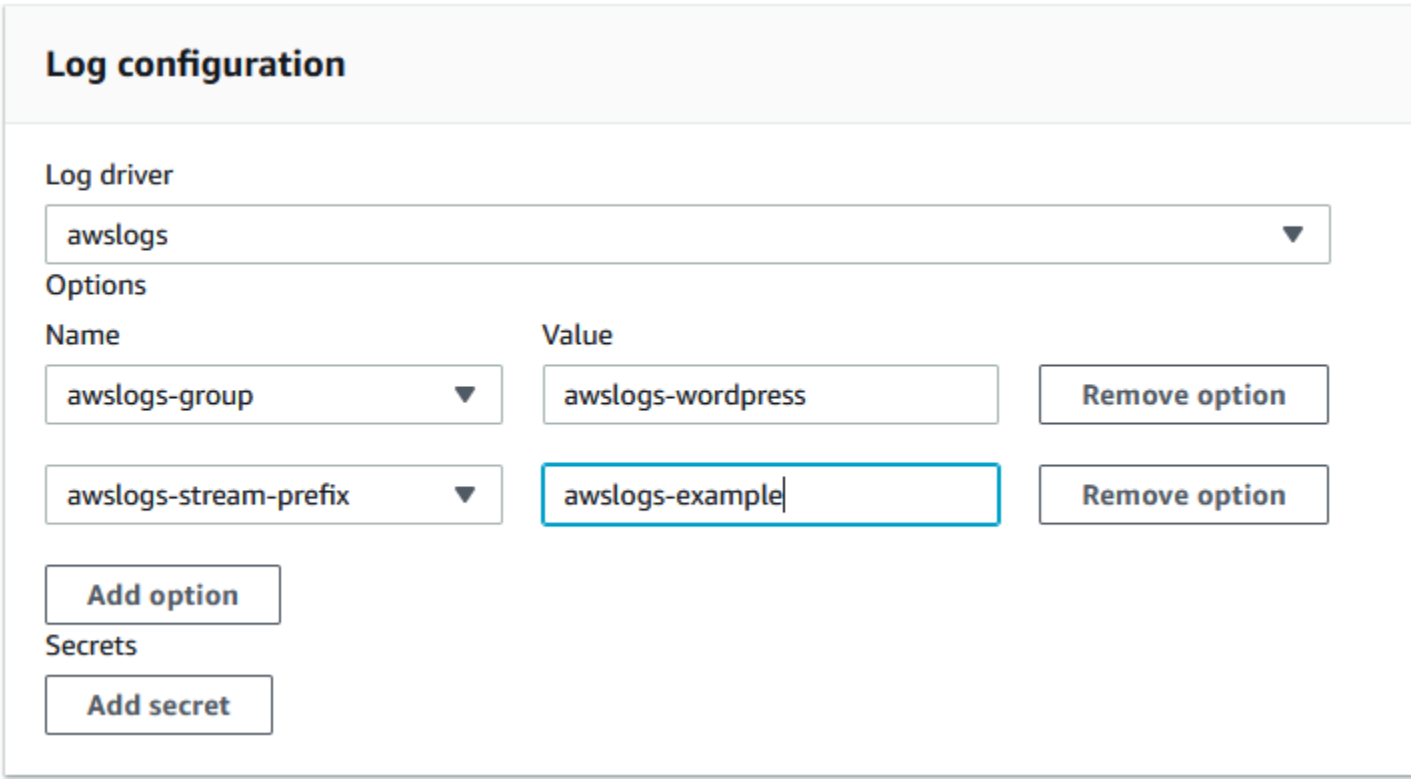
Los siguientes fragmentos de JSON de configuración de registro tienen un objeto `logConfiguration`especificado para cada trabajo. Uno es para un trabajo de WordPress que

envía registros a un grupo de registro denominado `awslogs-wordpress`, y otro es para un contenedor MySQL que envía registros a un grupo de registro denominado `awslogs-mysql`. Ambos contenedores utilizan el prefijo de flujo de registros `awslogs-example`.

```
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "awslogs-wordpress",
        "awslogs-stream-prefix": "awslogs-example"
    }
}
```

```
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "awslogs-mysql",
        "awslogs-stream-prefix": "awslogs-example"
    }
}
```

En la consola de AWS Batch, la configuración de registros para la definición de trabajo `wordpress` se especifica tal como se muestra en la imagen siguiente.



Después de haber registrado una definición de tarea con el controlador de registros `awslogs` en una configuración de registros de definición de trabajo puede enviar un trabajo o crear un servicio con dicha definición de trabajo para comenzar a enviar registros a CloudWatch Logs. Para obtener más información, consulte Enviar un trabajo.

Especificación de información confidencial

Con AWS Batch, puede inyectar datos confidenciales en sus trabajos, almacenando sus datos confidenciales en secretos de AWS Secrets Manager o parámetros de AWS Systems Manager de Parameter Store, y luego hacer referencia a ellos en la definición de su trabajo.

Los secretos se pueden exponer a un trabajo de las siguientes formas:

- Para inyectar datos sensibles en sus contenedores como variables de entorno, utilice el parámetro de definición de trabajos `secrets`.
- Para hacer referencia a información sensible en la configuración del registro de un trabajo, utilice el parámetro de definición del trabajo `secretOptions`.

Temas

- Especificación de información confidencial mediante Secrets Manager
- Especificación de información confidencial mediante el Parameter Store de Systems Manager

Especificación de información confidencial mediante Secrets Manager

Con AWS Batch, puede inyectar datos confidenciales en sus trabajos almacenándolos en AWS Secrets Manager secreto y luego haciendo referencia a ellos en la definición de su trabajo. La información confidencial almacenada en secretos de Secrets Manager se puede exponer a un trabajo como variables de entorno o como parte de la configuración del registro.

Cuando inyecta un secreto como variable de entorno, puede especificar una clave JSON o versión de un secreto para inyectar. Este proceso le ayuda a controlar la información confidencial expuesta al trabajo. Para obtener más información acerca del control de versiones de los secretos, consulte los Términos y conceptos clave de AWS Secrets Manager en la Guía del usuario de AWS Secrets Manager .

Consideraciones para la especificación de información confidencial mediante Secrets Manager

Cuando se utilice Secrets Manager para especificar información confidencial para trabajos, se debe tener en cuenta lo siguiente.

- Para inyectar un secreto utilizando una clave JSON específica o una versión de un secreto, la instancia de contenedor de su entorno de computación debe tener instalada la versión 1.37.0 o posterior del agente de contenedores de Amazon ECS. No obstante, recomendamos utilizar la versión del agente de contenedor más reciente. Para obtener información sobre cómo comprobar la versión del agente y actualizarlo a la versión más reciente, consulte Actualización del agente de contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

  Para insertar todo el contenido de un secreto como variable de entorno o para insertar un secreto en una configuración de registro, la instancia de contenedor debe tener la versión 1.23.0 o posterior del agente contenedor.
- Solo se admiten los secretos que almacenan datos de texto, que son secretos creados con el `SecretString` parámetro de la CreateSecretAPI. No se admiten los secretos que almacenan datos binarios, que son secretos creados con el `SecretBinary` parámetro de la CreateSecretAPI.
- Cuando utilice una definición de trabajos que haga referencia a secretos de Secrets Manager para recuperar información confidencial de los trabajos, si también está utilizando puntos de conexión de VPC de la interfaz, debe crear los puntos de conexión de VPC de la interfaz para Secrets Manager. Para obtener más información, consulte Utilización de Secrets Manager con puntos de enlace de la VPC en la Guía del usuario de AWS Secrets Manager .
- Los datos confidenciales se inyectan en el contenedor al iniciar el trabajo. Si el secreto se actualiza posteriormente o se rota, el trabajo no recibe automáticamente el valor actualizado. Debe lanzar un nuevo trabajo para obligar al servicio a lanzar uno nuevo con el valor secreto actualizado.

Permisos de IAM necesarios para los secretos AWS Batch

Para utilizar esta característica, debe tener la función de ejecución y hacer referencia a ella en la definición del trabajo. Esto permite que el agente de contenedor extraiga los recursos necesarios de Secrets Manager. Para obtener más información, consulte AWS Batch función de IAM de ejecución.

Para proporcionar acceso a los secretos de Secrets Manager que cree, agregue manualmente los siguientes permisos como una política insertada al rol de ejecución. Para obtener más información, consulte Adición y eliminación de políticas de IAM en la Guía del usuario de IAM.

- `secretsmanager:GetSecretValue`: obligatorio si se referencia un secreto de Secrets Manager.
- `kms:Decrypt`: obligatorio solo si el secreto utiliza una clave de KMS personalizada y no la clave predeterminada. El ARN de su clave personalizada debe añadirse como un recurso.

La siguiente política insertada de ejemplo agrega los permisos necesarios.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
    }
  ]
}
```

Introducción de información confidencial como variable de entorno

Dentro de la definición del trabajo, puede especificar los siguientes elementos:

- El objeto `secrets` que contiene el nombre de la variable de entorno que se va a establecer en el trabajo
- Nombre de recurso de Amazon (ARN) del secreto de Secrets Manager
- Parámetros adicionales que contienen información confidencial que se debe presentar al trabajo

En el ejemplo siguiente, se muestra la sintaxis completa que se debe especificar para el secreto de Secrets Manager.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-stage:version-id
```

En la siguiente sección se describen los parámetros adicionales. Algunos parámetros son opcionales. Sin embargo, si no los utiliza, debe incluir los dos puntos : para utilizar los valores por defecto. A continuación se ofrecen ejemplos para obtener más contexto.

`json-key`

Especifica el nombre de la clave en un par clave-valor con el valor que desea establecer como valor de variable de entorno. Solo se admiten valores en formato JSON. Si no especifica una clave JSON, se usa el contenido completo del secreto.

`version-stage`

Especifica la etiqueta de ensayo de la versión de un secreto que desea utilizar. Si se especifica una etiqueta de ensayo de versión, no se puede especificar un ID de versión. Si no se especifica ninguna etapa de versión, el comportamiento predeterminado consiste en recuperar el secreto con la etiqueta de ensayo `AWSCURRENT`.

Las etiquetas de ensayo se utilizan para realizar un seguimiento de las distintas versiones de un secreto cuando se actualizan o rotan. Cada versión de un secreto tiene una o varias etiquetas de ensayo y un ID. Para obtener más información, consulte Términos y conceptos clave de AWS Secrets Manager en la Guía del AWS Secrets Manager usuario.

`version-id`

Especifica el identificador único de la versión del secreto que desea utilizar. Si se especifica un ID de versión, no se puede especificar una etiqueta de ensayo de versión. Si no se especifica ningún ID de versión, el comportamiento predeterminado consiste en recuperar el secreto con la etiqueta de ensayo `AWSCURRENT`.

Los ID de versión se utilizan para realizar un seguimiento de las distintas versiones de un secreto cuando se actualizan o rotan. Cada versión de un secreto tiene un ID. Para obtener más información, consulte Términos y conceptos clave de AWS Secrets Manager en la Guía del AWS Secrets Manager usuario.

Definiciones de contenedor de ejemplo

En los siguientes ejemplos, se muestran las formas en las que se pueden referenciar secretos de Secrets Manager en las definiciones de contenedor.

Example hacer referencia a un secreto completo

A continuación, se incluye un fragmento de código de una definición de tarea que muestra el formato cuando se referencia el texto completo de un secreto de Secrets Manager.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
    }]
  }]
}
```

Example hacer referencia a una clave específica dentro de un secreto

A continuación se muestra un ejemplo del resultado de un get-secret-valuecomando que muestra el contenido de un secreto junto con la etiqueta provisional de la versión y el identificador de versión asociados al mismo.

```
{
    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
    "VersionId": "871d9eca-18aa-46a9-8785-981dd39ab30c",
    "SecretString": "{\"username1\":\"password1\",\"username2\":\"password2\",
\"username3\":\"password3\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreatedDate": 1581968848.921
}
```

Haga referencia a una clave específica de la salida anterior en una definición de contenedor especificando el nombre de clave al final del ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::"
```

```
    }]
  }]
}
```

Example hacer referencia a una versión de secreto específica

A continuación se muestra una salida de ejemplo de un comando describe-secret que muestra el contenido sin cifrar de un secreto junto con los metadatos de todas las versiones del secreto.

```
{
    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
    "Description": "Example of a secret containing application authorization data.",
    "RotationEnabled": false,
    "LastChangedDate": 1581968848.926,
    "LastAccessedDate": 1581897600.0,
    "Tags": [],
    "VersionIdsToStages": {
        "871d9eca-18aa-46a9-8785-981dd39ab30c": [
            "AWSCURRENT"
        ],
        "9d4cb84b-ad69-40c0-a0ab-cead36b967e8": [
            "AWSPREVIOUS"
        ]
    }
}
```

Haga referencia a una etiqueta de ensayo de versión específica de la salida anterior en una definición de contenedor especificando el nombre de clave al final del ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::AWSPREVIOUS:"
    }]
  }]
}
```

Haga referencia a un ID de versión específico de la salida anterior en una definición de contenedor especificando el nombre de clave al final del ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Example hacer referencia a una clave específica y una etiqueta de ensayo de versión de un secreto

A continuación se muestra cómo hacer referencia tanto a una clave específica dentro de un secreto como a una etiqueta de ensayo de versión específica.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1:AWSPREVIOUS:"
    }]
  }]
}
```

Para especificar una clave y un ID de versión específicos, utilice la sintaxis siguiente.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Introducción de información confidencial en una configuración de registro

En la definición de trabajo, al especificar una `logConfiguration`, puede especificar `secretOptions` con el nombre de la opción del controlador de registros para definir el contenedor y

el ARN completo del secreto de Secrets Manager que contiene la información confidencial que se va a presentar al contenedor.

A continuación, se incluye un fragmento de código de una definición de trabajo que muestra el formato cuando se referencia un secreto de Secrets Manager.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://cloud.splunk.com:8080"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
      }]
    }]
  }]
}
```

Crear un secreto AWS Secrets Manager

Puede utilizar la consola de Secrets Manager para crear un secreto con su información confidencial. Para obtener más información, consulte Creación de un secreto básico en la Guía del usuario de AWS Secrets Manager .

Para crear un secreto básico

Utilice Secrets Manager para crear un secreto con su información confidencial.

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija Almacenar un secreto nuevo.
3. En Select secret type (Seleccionar tipo de secreto), elija Other type of secrets (Otro tipo de secretos).
4. Especifique los detalles de su secreto personalizado como pares de clave y valor. Por ejemplo, puede especificar una clave `UserName` y, a continuación, proporcionar el nombre de usuario como su valor. Agregue una segunda clave con el nombre `Password` y el texto de la contraseña como su valor. También puede añadir entradas para el nombre de una base de datos, la

dirección del servidor o el puerto TCP. Puede añadir tantas parejas como sea necesario para almacenar la información que necesite.

Como opción, puede elegir la pestaña Plaintext (Texto no cifrado) y escribir el valor del secreto del modo que prefiera.

5. Elija la clave de AWS KMS cifrado que desee usar para cifrar el texto protegido del secreto. Si no elige ninguna, Secrets Manager comprueba si existe una clave predeterminada para la cuenta y la utiliza si existe. Si no existe una clave predeterminada, Secrets Manager crea una automáticamente. También puede elegir Add new key (Añadir nueva clave) para crear una clave de KMS personalizada específicamente para este secreto. Para crear su propia clave de KMS, debe tener permisos para crear claves de KMS en su cuenta.
6. Elija Siguiente.
7. En Secret name (Nombre del secreto), escriba una ruta y un nombre opcionales, tales como **`production/MyAwesomeAppSecret`** o **`development/TestSecret`** y elija Next (Siguiente). Opcionalmente, puede añadir una descripción para ayudarle a recordar el objetivo de este secreto más adelante.

   El nombre del secreto debe estar formado por letras ASCII, dígitos o cualquiera de los siguientes caracteres: /_+=.@-
8. (Opcional) En este punto puede configurar la rotación para su secreto. Para este procedimiento, deje seleccionado Disable automatic rotation (Deshabilitar la rotación automática) y, a continuación, elija Next (Siguiente).

   Para obtener información sobre cómo configurar la rotación de datos secretos nuevos o existentes, consulte Rotación de sus datos AWS Secrets Manager secretos.
9. Revise la configuración y, a continuación, elija Store secret (Almacenar secreto) para guardar todo lo que ingresó como nuevo secreto en Secrets Manager.

Especificación de información confidencial mediante el Parameter Store de Systems Manager

Con AWS Batch, puede inyectar datos confidenciales en sus contenedores almacenándolos en los parámetros del almacén de parámetros y, a continuación, haciendo referencia a ellos en la definición de su contenedor. AWS Systems Manager

Temas

- Consideraciones para especificar información confidencial mediante el Parameter Store de Systems Manager
- Permisos de IAM necesarios para los secretos AWS Batch
- Introducción de información confidencial como variable de entorno
- Introducción de información confidencial en una configuración de registro
- Crear un AWS Systems Manager parámetro de almacén de parámetros

Consideraciones para especificar información confidencial mediante el Parameter Store de Systems Manager

Cuando se especifica información confidencial para contenedores que utilizan parámetros del Parameter Store de Systems Manager, se debe tener en cuenta lo siguiente.

- Esta función requiere que la instancia de contenedor tenga la versión 1.23.0 o posterior del agente contenedor. No obstante, recomendamos utilizar la versión del agente de contenedor más reciente. Para obtener información sobre cómo comprobar la versión del agente y actualizarlo a la versión más reciente, consulte Actualización del agente de contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.
- Los datos confidenciales se inyectan en el contenedor para su trabajo al iniciar el contenedor. Si el secreto o el parámetro de Parameter Store se actualizan posteriormente o se rotan, el contenedor no recibe automáticamente el valor actualizado. Debe lanzar un nuevo trabajo para obligar el lanzamiento de uno nuevo con los secretos actualizados.

Permisos de IAM necesarios para los secretos AWS Batch

Para utilizar esta característica, debe tener la función de ejecución y hacer referencia a ella en la definición del trabajo. Esto permite al agente de contenedores de Amazon ECS obtener los AWS Systems Manager recursos necesarios. Para obtener más información, consulte AWS Batch función de IAM de ejecución.

Para proporcionar acceso a los AWS Systems Manager parámetros del almacén de parámetros que cree, añada manualmente los siguientes permisos como política integrada a la función de ejecución. Para obtener más información, consulte Adición y eliminación de políticas de IAM en la Guía del usuario de IAM.

- `ssm:GetParameters`: obligatorio si se hace referencia a un parámetro del Parameter Store de Systems Manager en una definición de tareas.
- `secretsmanager:GetSecretValue`: obligatorio si se hace referencia a un secreto de Secrets Manager directamente o si el parámetro del Parameter Store de Systems Manager hace referencia a un secreto de Secrets Manager en una definición de tareas.
- `kms:Decrypt`: obligatorio solo si el secreto utiliza una clave de KMS personalizada y no la clave predeterminada. El ARN de su clave personalizada debe añadirse como un recurso.

A continuación, mostramos un ejemplo de política insertada que añade los permisos necesarios.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:<region>:<aws_account_id>:parameter/<parameter_name>",
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
    }
  ]
}
```

Introducción de información confidencial como variable de entorno

En la definición de contenedor, especifique `secrets` con el nombre de la variable de entorno que se va a establecer en el contenedor y el ARN completo del parámetro del Parameter Store de Systems Manager que contiene la información confidencial que se va a presentar al contenedor.

A continuación, se incluye un fragmento de código de una definición de tarea que muestra el formato cuando se hace referencia a un parámetro del almacén de parámetros de Systems Manager. Si el parámetro del almacén de parámetros de Systems Manager está en la misma región que la tarea que se va a lanzar, se puede utilizar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Introducción de información confidencial en una configuración de registro

En la definición de contenedor, al especificar una `logConfiguration`, puede especificar `secretOptions` con el nombre de la opción del controlador de registros que va a definir en el contenedor y el ARN completo del parámetro del Parameter Store de Systems Manager que contiene la información confidencial que se va a presentar al contenedor.

> **Important**
>
> Si el parámetro del almacén de parámetros de Systems Manager está en la misma región que la tarea que se va a lanzar, se puede utilizar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

A continuación, se incluye un fragmento de código de una definición de tarea que muestra el formato cuando se hace referencia a un parámetro del almacén de parámetros de Systems Manager.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
        "name": "fluentd-address",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
      }]
    }]
  }]
```

```
}
```

Crear un AWS Systems Manager parámetro de almacén de parámetros

Puede utilizar la AWS Systems Manager consola para crear un parámetro del almacén de parámetros de Systems Manager para sus datos confidenciales. Para obtener más información, consulte Explicación: Creación y utilización de un parámetro en un comando (consola) en la Guía del usuario de AWS Systems Manager .

Para crear un parámetro con Parameter Store

1. Abra la AWS Systems Manager consola en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Parameter Store, Create parameter (Crear parámetro).
3. En Name (Nombre), escriba una jerarquía y un nombre de parámetro. Por ejemplo, escriba `test/database_password`.
4. En Description (Descripción), escriba una descripción opcional.
5. En Tipo, elija Cadena o SecureString. StringList

> **Note**
>
> - Si lo elige SecureString, aparece el campo ID de clave de KMS. Si no proporciona un ID de clave de KMS, un ARN de clave de KMS, un nombre de alias o un ARN de alias, el sistema utiliza `alias/aws/ssm`. Esta es la clave de KMS predeterminada para Systems Manager. Para evitar utilizar esta clave, elija una clave personalizada. Para obtener más información, consulte Utilización de parámetros de cadena segura en la Guía del usuario de AWS Systems Manager .
> - Si crea un parámetro de cadena segura en la consola mediante el parámetro `key-id` con un nombre de alias de clave de KMS o un ARN de alias personalizado, debe especificar el prefijo `alias/` antes del alias. A continuación se muestra un ejemplo de ARN:
>
>   ```
>   arn:aws:kms:us-east-2:123456789012:alias/MyAliasName
>   ```
>
>   El siguiente es un ejemplo de un nombre de alias:
>
>   ```
>   alias/MyAliasName
>   ```

6. En Value (Valor), escriba un valor. Por ejemplo, `MyFirstParameter`. Si lo elige SecureString, el valor se enmascara exactamente como lo ingresó.
7. Elija Create parameter.

Autenticación de registro privado para trabajos

La autenticación del registro privado para los trabajos AWS Secrets Manager le permite almacenar sus credenciales de forma segura y luego hacer referencia a ellas en la definición de su trabajo. Esto proporciona una forma de hacer referencia a las imágenes de contenedores que existen en registros privados y AWS que no requieren autenticación en las definiciones de trabajo. Esta función es compatible con los trabajos alojados en las instancias de Amazon EC2 y en Fargate.

> **Important**
>
> Si la definición de su puesto hace referencia a una imagen almacenada en Amazon ECR, este tema no se aplica. Para obtener más información, consulte Utilización de imágenes de Amazon ECR con Amazon ECS en la Guía del usuario de Amazon Elastic Container Registry.

Para los trabajos alojados en instancias de Amazon EC2, esta función requiere una versión `1.19.0` o posterior del agente contenedor. No obstante, recomendamos utilizar la versión del agente de contenedor más reciente. Para obtener información sobre cómo comprobar la versión del agente y actualizarla a la última versión, consulte Actualización del agente contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

Para los trabajos alojados en Fargate, esta función requiere una versión de plataforma `1.2.0` o posterior. Para obtener más información, consulte las versiones de la plataforma AWS Fargate Linux en la Guía para desarrolladores de Amazon Elastic Container Service.

En la definición de contenedor, especifique el objeto `repositoryCredentials` con los detalles del secreto que ha creado. El secreto al que haga referencia puede provenir de una cuenta distinta Región de AWS o distinta de la del trabajo que lo utiliza.

> **Note**
>
> Al usar la AWS Batch API o el AWS SDK, si el secreto existe en el Región de AWS mismo lugar que el trabajo que está lanzando, puede usar el ARN completo o el nombre del

secreto. AWS CLI Si el secreto existe en otra cuenta, debe especificarse el ARN completo del secreto. Al utilizar el AWS Management Console, se debe especificar siempre el ARN completo del secreto.

A continuación se muestra un fragmento de una definición de trabajo que muestra los parámetros necesarios:

```
"containerProperties": [
  {
    "image": "private-repo/private-image",
    "repositoryCredentials": {
      "credentialsParameter":
 "arn:aws:secretsmanager:region:123456789012:secret:secret_name"
    }
  }
]
```

Permisos de IAM requeridos para la autenticación de registros privados

El rol de ejecución es obligatorio para usar esta función. Esto permite que el agente de contenedor extraiga la imagen del contenedor. Para obtener más información, consulte AWS Batch función de IAM de ejecución.

Para proporcionar acceso a los secretos que cree, añada los siguientes permisos como política integrada a la función de ejecución. Para obtener más información, consulte Adición y eliminación de políticas de IAM.

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`: solo se requiere si la clave utiliza una clave de KMS personalizada y no la clave de KMS predeterminada. Se debe agregar el nombre de recurso de Amazon (ARN) de la clave de personalizada como un recurso.

Es siguiente es un ejemplo de política insertada que agrega los permisos.

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

```
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:123456789012:secret:secret_name",
        "arn:aws:kms:region:123456789012:key/key_id"
      ]
    }
  ]
}
```

Uso de autenticación de registros privados

Para crear un secreto básico

Úselo AWS Secrets Manager para crear un secreto para sus credenciales de registro privado.

1. Abra la AWS Secrets Manager consola en https://console.aws.amazon.com/secretsmanager/.
2. Elija Almacenar un secreto nuevo.
3. En Select secret type (Seleccionar tipo de secreto), elija Other type of secrets (Otro tipo de secretos).
4. Seleccione Plaintext (Texto no cifrado) y especifique sus credenciales de registros privados con el siguiente formato:

   ```
   {
     "username" : "privateRegistryUsername",
     "password" : "privateRegistryPassword"
   }
   ```

5. Seleccione Siguiente.
6. En Secret name (Nombre del secreto), ingrese una ruta y un nombre opcionales, como **`production/MyAwesomeAppSecret`** o **`development/TestSecret`**, y elija Next (Siguiente). Opcionalmente, puede añadir una descripción para ayudarle a recordar el objetivo de este secreto más adelante.

   El nombre del secreto debe estar formado por letras ASCII, dígitos o alguno de los siguientes caracteres: `/_+=.@-`.

7. (Opcional) En este punto puede configurar la rotación para su secreto. Para este procedimiento, deje seleccionado Disable automatic rotation (Deshabilitar la rotación automática) y, a continuación, elija Next (Siguiente).

   Para obtener instrucciones sobre cómo configurar la rotación de datos secretos nuevos o existentes, consulte Cómo cambiar tus datos AWS Secrets Manager secretos.
8. Revise la configuración y, a continuación, elija Store secret (Almacenar secreto) para guardar todo lo que ingresó como secreto nuevo en Secrets Manager.

Registre una definición de trabajo y, en Registro privado, active la autenticación del registro privado. A continuación, en nombre o ARN de Secrets Manager, introduzca el nombre de recurso de Amazon (ARN) del secreto. Para obtener más información, consulte Permisos de IAM requeridos para la autenticación de registros privados.

Volúmenes de Amazon EFS

Amazon Elastic File System (Amazon EFS) proporciona almacenamiento de archivos sencillo y escalable para usarlo con sus trabajos AWS Batch. Con Amazon EFS, la capacidad de almacenamiento es elástica. Escala automáticamente a medida que se agregan o eliminan archivos. Las aplicaciones disponen del almacenamiento que necesitan, cuando lo necesitan.

Puede utilizar sistemas de archivos de Amazon EFS con AWS Batch para exportar los datos del sistema de archivos a través de la flota de instancias de contenedor. De esta forma, sus trabajos tienen acceso al mismo almacenamiento persistente. No obstante, debe configurar su AMI de instancia de contenedor para montar el sistema de archivos de Amazon EFS antes de que se inicie el daemon de Docker. Además, sus definiciones de trabajo deben hacer referencia a montajes de volúmenes en la instancia de contenedor para utilizar el sistema de archivos. Las secciones siguientes le ayudarán a comenzar a utilizar Amazon EFS con AWS Batch.

Consideraciones acerca de volúmenes de Amazon EFS

Al utilizar volúmenes de Amazon EFS, se debe tener en cuenta lo siguiente:

- Para los trabajos que utilizan recursos EC2, se ha agregado compatibilidad con el sistema de archivos de Amazon EFS como vista previa pública en la versión AMI optimizada para Amazon ECS `20191212` con el agente de contenedor versión 1.35.0. Sin embargo, el sistema de archivos de Amazon EFS está disponible con carácter general a partir de la versión `20200319` de la AMI

optimizada para Amazon ECS con el agente de contenedor versión 1.38.0, que contenía las características de punto de acceso de Amazon EFS y autorización de IAM. Recomendamos utilizar la versión AMI optimizada para Amazon ECS `20200319` o una posterior para aprovechar estas características. Para obtener más información, consulte la sección sobre Versiones AMI Linux optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

> **Note**
>
> Si crea su propia AMI, debe usar el agente de contenedor 1.38.0 o posterior, versión `ecs-init` 1.38.0-1 o una posterior, y ejecutar los siguientes comandos en su instancia de Amazon EC2. Todo esto es para habilitar el complemento de volumen Amazon ECS. Los comandos dependen de si utiliza Amazon Linux 2 o Amazon Linux como imagen base.
>
> Amazon Linux 2
>
> ```
> $ yum install amazon-efs-utils
> systemctl enable --now amazon-ecs-volume-plugin
> ```
>
> Amazon Linux
>
> ```
> $ yum install amazon-efs-utils
> sudo shutdown -r now
> ```

- Para los trabajos que utilizan recursos de Fargate, se agregó la compatibilidad con el sistema de archivos de Amazon EFS al usar la versión 1.4.0 o una posterior de la plataforma. Para obtener más información, consulte AWSVersiones de la plataforma de Fargate en la Guía para desarrolladores de Amazon Elastic Container Service.
- Cuando se utilizan volúmenes de Amazon EFS en trabajos que usan recursos de Fargate, Fargate crea un contenedor supervisor responsable de administrar el volumen de Amazon EFS. El contenedor supervisor utiliza una pequeña cantidad de la memoria del trabajo. El contenedor supervisor puede verse al consultar la versión 4 del punto de conexión de metadatos de la tarea. Para obtener más información, consulte Versión 4 del punto de conexión de metadatos de tareas en la Guía del usuario de Amazon Elastic Container Service para AWS Fargate.

Uso de puntos de acceso de Amazon EFS

Los puntos de acceso de Amazon EFS son puntos de entrada específicos de la aplicación a un sistema de archivos EFS que le ayudan a administrar el acceso de la aplicación a conjuntos de datos compartidos. Para obtener más información acerca de los puntos de acceso de Amazon EFS y cómo controla el acceso a ellos, consulte Uso de puntos de acceso de Amazon EFS en la Guía del usuario de Amazon Elastic File System.

Los puntos de acceso pueden imponer una identidad de usuario, incluidos los grupos POSIX del usuario, para todas las solicitudes del sistema de archivos que se realizan a través del punto de acceso. Los puntos de acceso también pueden imponer un directorio raíz diferente para el sistema de archivos, de modo que los clientes solo puedan acceder a los datos del directorio especificado o de sus subdirectorios.

> **Note**
>
> Cuando se crea un punto de acceso EFS, se especifica una ruta en el sistema de archivos para que sirva como directorio raíz. Cuando se hace referencia al sistema de archivos de EFS con un ID de punto de acceso en la definición de trabajo de AWS Batch, el directorio raíz se debe omitir o establecer en `/`, lo que aplicará la ruta establecida en el punto de acceso de EFS.

Puede utilizar un rol de IAM con el trabajo de AWS Batch para exigir que determinadas aplicaciones utilicen un punto de acceso específico. Al combinar políticas de IAM con puntos de acceso, puede proporcionar fácilmente acceso seguro a conjuntos de datos específicos para sus aplicaciones. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

Especificación de un sistema de archivos de Amazon EFS en la definición de trabajo

Para usar volúmenes del sistema de archivos de Amazon EFS para sus contenedores, debe especificar las configuraciones de volumen y punto de montaje en su definición de trabajo. El siguiente fragmento de JSON de definición de trabajo muestra la sintaxis de los objetos `volumes` y `mountPoints` para un contenedor:

```
{
    "containerProperties": [
        {
            "image": "amazonlinux:2",
            "command": [
                "ls",
                "-la",
                "/mount/efs"
            ],
            "mountPoints": [
                {
                    "sourceVolume": "myEfsVolume",
                    "containerPath": "/mount/efs",
                    "readOnly": true
                }
            ],
            "volumes": [
                {
                    "name": "myEfsVolume",
                    "efsVolumeConfiguration": {
                        "fileSystemId": "fs-12345678",
                        "rootDirectory": "/path/to/my/data",
                        "transitEncryption": "ENABLED",
                        "transitEncryptionPort": integer,
                        "authorizationConfig": {
                            "accessPointId": "fsap-1234567890abcdef1",
                            "iam": "ENABLED"
                        }
                    }
                }
            ]
        }
    ]
}
```

`efsVolumeConfiguration`

Tipo: objeto

Obligatorio: no

Este parámetro se especifica cuando se usan volúmenes de Amazon EFS.

`fileSystemId`

Tipo: cadena

Obligatorio: sí

El ID del sistema de archivos de Amazon EFS que se va a usar.

`rootDirectory`

Tipo: cadena

Obligatorio: no

Directorio del sistema de archivos de Amazon EFS que se va a montar como directorio raíz dentro del host. Si se omite este parámetro, se utiliza la raíz del volumen de Amazon EFS. Si se especifica `/`, se obtiene el mismo efecto que si se omite este parámetro. Puede tener hasta 4 096 caracteres de longitud.

> **Important**
>
> Si se especifica un punto de acceso de EFS en el `authorizationConfig`, se debe omitir el parámetro del directorio raíz o establecerlo en `/`. Esto impone la ruta establecida en el punto de acceso EFS.

`transitEncryption`

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

Determina si se habilita el cifrado de los datos en tránsito de Amazon EFS entre el host de AWS Batch y el servidor de Amazon EFS. El cifrado en tránsito debe estar habilitado si se utiliza la autorización de IAM en Amazon EFS. Si se omite este parámetro, se usa el valor predeterminado de `DISABLED`. Para obtener más información, consulte Cifrado de datos en tránsito en la Guía del usuario de Amazon Elastic File System.

`transitEncryptionPort`

Tipo: entero

Obligatorio: no

El puerto que se utilizará al enviar datos cifrados entre el host de AWS Batch y el servidor de Amazon EFS. Si no se especifica un puerto de cifrado en tránsito, se emplea la estrategia de selección de puertos que utiliza el ayudante de montaje de Amazon EFS. Este valor debe estar entre 0 y 65 535. Para obtener más información, consulte Ayudante de montaje de EFS en la Guía del usuario de Amazon Elastic File System.

`authorizationConfig`

Tipo: objeto

Obligatorio: no

Los detalles de configuración de autorización en el sistema de archivos de Amazon EFS.

`accessPointId`

Tipo: cadena

Obligatorio: no

ID de punto de acceso que se va a utilizar. Si se especifica un punto de acceso, el valor del directorio raíz en `efsVolumeConfiguration` se debe omitir o establecer en `/`. Esto impone la ruta establecida en el punto de acceso EFS. Si se utiliza un punto de acceso, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Para obtener más información, consulte Trabajo con puntos de acceso de Amazon EFS en la Guía del usuario de Amazon Elastic File System.

`iam`

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

Determina si se debe utilizar el rol de IAM del trabajo AWS Batch definido en una definición de trabajo al montar el sistema de archivos de Amazon EFS. Si está habilitado, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Si se omite este parámetro, se usa el valor predeterminado de `DISABLED`. Para obtener más información sobre roles IAM de ejecución consulte AWS Batch función de IAM de ejecución.

Ejemplos de definiciones de trabajo

Los ejemplos de definiciones de trabajo a continuación ilustran cómo utilizar patrones comunes como, por ejemplo, variables de entorno, sustituciones de parámetros y montajes de volúmenes.

Utilización de variables de entorno

En el siguiente ejemplo, la definición de trabajo utiliza variables de entorno para especificar un tipo de archivo y un URL de Amazon S3. Este ejemplo se ha extraído del artículo de blog de informática titulado Creating a Simple "Fetch & Run" AWS Batch Job. El script `fetch_and_run.sh` que se describe en el artículo utiliza estas variables de entorno para descargar el script `myjob.sh` de S3 y declarar su tipo de archivo.

Aunque en este ejemplo el comando y las variables de entorno están codificadas de forma rígida en la definición de trabajo, puede especificar sustituciones de comandos y de variables de entorno para que la definición de trabajo más versátil.

```
{
    "jobDefinitionName": "fetch_and_run",
    "type": "container",
    "containerProperties": {
        "image": "123456789012.dkr.ecr.us-east-1.amazonaws.com/fetch_and_run",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "2000"
            },
            {
                "type": "VCPU",
                "value": "2"
            }
        ],
        "command": [
            "myjob.sh",
            "60"
        ],
        "jobRoleArn": "arn:aws:iam::123456789012:role/AWSBatchS3ReadOnly",
        "environment": [
            {
                "name": "BATCH_FILE_S3_URL",
                "value": "s3://my-batch-scripts/myjob.sh"
            },
```

```
            {
                "name": "BATCH_FILE_TYPE",
                "value": "script"
            }
        ],
        "user": "nobody"
    }
}
```

Cómo usar la sustitución de parámetros

El siguiente ejemplo de definición de trabajo ilustra cómo permitir la sustitución de parámetros y cómo establecer valores predeterminados.

Las declaraciones `Ref::` de la sección `command` se utilizan para definir marcadores de posición para sustituir parámetros. Al enviar un trabajo con esta definición de trabajo, se especifican las sustituciones de parámetros que ocuparán dichos valores, como `inputfile` y `outputfile`. La sección `parameters` a continuación establece un valor predeterminado para `codec`, pero es posible invalidar ese parámetro si fuera necesario.

Para obtener más información, consulte Parámetros.

```
{
    "jobDefinitionName": "ffmpeg_parameters",
    "type": "container",
    "parameters": {"codec": "mp4"},
    "containerProperties": {
        "image": "my_repo/ffmpeg",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "2000"
            },
            {
                "type": "VCPU",
                "value": "2"
            }
        ],
        "command": [
            "ffmpeg",
            "-i",
            "Ref::inputfile",
```

```
            "-c",
            "Ref::codec",
            "-o",
            "Ref::outputfile"
        ],
        "jobRoleArn": "arn:aws:iam::123456789012:role/ECSTask-S3FullAccess",
        "user": "nobody"
    }
}
```

Funcionalidad de GPU de prueba

En el siguiente ejemplo, la definición de trabajo prueba si la AMI de carga de trabajo de GPU descrita en Cómo utilizar una AMI de carga de trabajo de GPU se ha configurado correctamente. Esta definición de trabajo de ejemplo ejecuta el ejemplo del clasificador TensorFlow deep MNIST de GitHub.

```
{
    "containerProperties": {
        "image": "tensorflow/tensorflow:1.8.0-devel-gpu",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "32000"
            },
            {
                "type": "VCPU",
                "value": "8"
            }
        ],
        "command": [
            "sh",
            "-c",
            "cd /tensorflow/tensorflow/examples/tutorials/mnist; python mnist_deep.py"
        ]
    },
    "type": "container",
    "jobDefinitionName": "tensorflow_mnist_deep"
}
```

Es posible crear un archivo con el texto JSON indicado arriba, que se denomine `tensorflow_mnist_deep.json` y, a continuación, registrar una definición de trabajo de AWS Batch con el siguiente comando:

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Trabajo paralelo de varios nodos

La definición de trabajo de ejemplo siguiente muestra un trabajo paralelo de varios nodos. Para obtener más información, consulte Creación de un flujo de trabajo de dinámica molecular estrechamente relacionado con trabajos paralelos de múltiples nodos en AWS Batch en el blog de AWS Compute.

```
{
  "jobDefinitionName": "gromacs-jobdef",
  "jobDefinitionArn": "arn:aws:batch:us-east-2:123456789012:job-definition/gromacs-
jobdef:1",
  "revision": 6,
  "status": "ACTIVE",
  "type": "multinode",
  "parameters": {},
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:1",
        "container": {
          "image": "123456789012.dkr.ecr.us-east-2.amazonaws.com/gromacs_mpi:latest",
          "resourceRequirements": [
              {
                  "type": "MEMORY",
                  "value": "24000"
              },
              {
                  "type": "VCPU",
                  "value": "8"
              }
          ],
          "command": [],
          "jobRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
          "ulimits": [],
```

```
          "instanceType": "p3.2xlarge"
        }
      }
    ]
  }
}
```

Colas de trabajo

Los trabajos se envían a una cola de trabajos, donde permanecen hasta que pueden programarse para ejecutarse en un entorno informático. Una cuenta de AWS puede tener varias colas de trabajo. Por ejemplo, puede crear una cola que utilice instancias bajo demanda de Amazon EC2 para trabajos de alta prioridad y otra cola que utilice instancias de spot de Amazon EC2 para trabajos de baja prioridad. Las colas de trabajo tienen una prioridad que el programador utiliza para determinar qué cola debe evaluarse primero para su ejecución.

Temas

Cómo crear de una cola de trabajos

Antes de enviar trabajos en AWS Batch, es necesario crear una cola de trabajo. Al crear una cola de trabajos, se asocian uno o varios entornos informáticos a la cola y se asigna un orden de preferencia.

También se establece la prioridad de la cola de trabajos que determina el orden en que el programador de lotes AWS coloca los trabajos. Esto significa que, si un entorno informático está asociado a más de una cola de trabajos, se da preferencia a la cola de trabajos con mayor prioridad.

Creación de una cola de trabajos de Fargate

Para crear una cola de trabajos Fargate

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS a utilizar.
3. En el panel de navegación, elija Colas de trabajos.
4. Seleccione Create (Crear).
5. En Tipo de orquestación, elija Fargate.
6. En Nombre, escriba un nombre único para su cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números y guiones bajos (_).

7. En Prioridad, introduzca un valor entero para la prioridad de la cola de trabajos. Las colas de trabajo con mayor prioridad se ejecutan antes que con las colas de menor prioridad asociadas con un mismo entorno informático. La prioridad se determina en orden descendente. Por ejemplo, una cola de trabajos con una prioridad 10, tendrá mayor preferencia de programación que una cola de trabajos cuyo valor de prioridad sea 1.
8. (Opcional) En Política de programación Nombre de recurso de Amazon (ARN), elija una política de programación existente.
9. En Entornos informáticos conectados, seleccione uno o varios entornos informáticos de la lista para asociarlos a la cola de trabajos. Seleccione los entornos informáticos en el orden en que desee que la cola intente colocarlos en la cola de trabajos. El programador de trabajos utiliza el orden en el que selecciona los entornos informáticos para determinar qué entorno informático inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos informáticos deben estar en estado `VALID`. Puede asociar hasta tres entornos informáticos con una cola de trabajos.

   > **Note**
   >
   > Todos los entornos informáticos asociados a una cola de trabajos deben compartir el mismo modelo de aprovisionamiento. AWS Batch no admite la mezcla de modelos de aprovisionamiento en una sola cola de trabajos.

10. En Orden de entorno informático, seleccione las flechas arriba y abajo para configurar el orden que desee.
11. Seleccione Crear cola de trabajos para finalizar y crear su cola de trabajos.

Creación de una cola de trabajos de Amazon EC2

Para crear una cola de Amazon EC2

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS a utilizar.
3. En el panel de navegación, elija Colas de trabajos.
4. Seleccione Create (Crear).
5. Para el Tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).

6. En Nombre, escriba un nombre único para su cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números y guiones bajos (_).
7. En Prioridad, introduzca un valor entero para la prioridad de la cola de trabajos. Las colas de trabajo con mayor prioridad se ejecutan antes que con las colas de menor prioridad asociadas con un mismo entorno informático. La prioridad se determina en orden descendente. Por ejemplo, una cola de trabajos con una prioridad 10, tendrá mayor preferencia de programación que una cola de trabajos cuyo valor de prioridad sea 1.
8. (Opcional) En Política de programación Nombre de recurso de Amazon (ARN), elija una política de programación existente.
9. En Entornos informáticos conectados, seleccione uno o varios entornos informáticos de la lista para asociarlos a la cola de trabajos. Seleccione los entornos informáticos en el orden en que desee que la cola intente colocarlos en la cola de trabajos. El programador de trabajos utiliza el orden en el que selecciona los entornos informáticos para determinar qué entorno informático inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos informáticos deben estar en estado `VALID`. Puede asociar hasta tres entornos informáticos con una cola de trabajos. Si no tiene un entorno informático existente, elija Crear entorno informático

   > **Note**
   >
   > Todos los entornos informáticos asociados a una cola de trabajos deben compartir el mismo modelo de aprovisionamiento. AWS Batch no admite la mezcla de modelos de aprovisionamiento en una sola cola de trabajos.

10. En Orden de entorno informático, seleccione las flechas arriba y abajo para configurar el orden que desee.
11. Seleccione Crear cola de trabajos para finalizar y crear su cola de trabajos.

Creación de una cola de trabajos de Amazon EKS

Para crear una cola de trabajos de Amazon EKS

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS a utilizar.
3. En el panel de navegación, elija Colas de trabajos.

4. Seleccione Create (Crear).
5. Para el Tipo de orquestación, elija Amazon Elastic Kubernetes Service (Amazon EKS).
6. En Nombre, escriba un nombre único para su cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números y guiones bajos (_).
7. En Prioridad, escriba un valor entero para la prioridad de la cola de trabajos. Las colas de trabajo con mayor prioridad se ejecutan antes que con las colas de menor prioridad asociadas con un mismo entorno informático. La prioridad se determina en orden descendente. Por ejemplo, una cola de trabajos con una prioridad 10, tendrá mayor preferencia de programación que una cola de trabajos cuyo valor de prioridad sea 1.
8. (Opcional) En Política de programación Nombre de recurso de Amazon (ARN), elija una política de programación existente.
9. En Entornos informáticos conectados, seleccione uno o varios entornos informáticos de la lista para asociarlos a la cola de trabajos. Seleccione los entornos informáticos en el orden en que desee que la cola intente colocarlos en la cola de trabajos. El programador de trabajos utiliza el orden en el que selecciona los entornos informáticos para determinar qué entorno informático inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos informáticos deben estar en estado `VALID`. Puede asociar hasta tres entornos informáticos con una cola de trabajos.

   > **Note**
   >
   > Todos los entornos informáticos asociados a una cola de trabajos deben compartir el mismo modelo de aprovisionamiento. AWS Batch no admite la mezcla de modelos de aprovisionamiento en una sola cola de trabajos.

   > **Note**
   >
   > Todos los entornos informáticos asociados a una cola de trabajos deben compartir la misma arquitectura. AWS Batch no admite la mezcla de tipos de arquitecturas de entorno informático en una sola cola de trabajos.

10. En Orden de entorno informático, seleccione las flechas arriba y abajo para configurar el orden que desee.
11. Seleccione Crear cola de trabajos para finalizar y crear su cola de trabajos.

Plantilla de cola de trabajos

A continuación se muestra una plantilla de cola de trabajos vacía. Puede utilizar esta plantilla para crear su cola de trabajos. A continuación, puede guardar esta cola de trabajos en un archivo y utilizarla con la AWS CLI `--cli-input-json` opción. Para obtener más información sobre estos parámetros, consulte la referencia CreateJobQueuede la AWS Batch API.

```
{
   "computeEnvironmentOrder": [
      {
         "computeEnvironment": "",
         "order": 0
      }
   ],
   "jobQueueName": "",
   "jobStateTimeLimitActions": [
      {
         "state": "RUNNABLE",
         "action": "CANCEL",
   "maxTimeSeconds": 0,
         "reason": ""

      }
   ],
   "priority": 0,
   "schedulingPolicyArn": "",
   "state": "ENABLED",
   "tags": {
      "KeyName": ""
   }
}
```

Note

Puede generar la plantilla de cola de trabajos anterior con el siguiente AWS CLI comando.

```
$ aws batch create-job-queue --generate-cli-skeleton
```

Parámetros de cola de trabajos

Las colas de trabajos se dividen en cuatro componentes básicos: nombre, estado, prioridad y orden del entorno de cómputo. En esta sección se analizan estos componentes asociados a los componentes.

Temas

Nombre de la cola de trabajos

`jobQueueName`

El nombre de la cola de trabajos. Se admiten hasta 128 letras (mayúsculas y minúsculas), números y caracteres de subrayado.

Tipo: cadena

Obligatorio: sí

Acciones de límite de tiempo del estado de la cola de trabajos

`jobStateTimeLimitActions`

El conjunto de acciones que se AWS Batch realizan en los trabajos que permanecen al principio de la cola de trabajos en el estado especificado durante más tiempo del especificado. AWS Batch realizará cada acción una vez que `maxTimeSeconds` haya pasado. (Nota: el valor mínimo `maxTimeSeconds` es 600 (10 minutos) y su valor máximo es 86.400 (24 horas).)

Tipo: matriz de objetos `JobStateTimeLimitActions`

Obligatorio: no

Priority (Prioridad)

`priority`

Prioridad de la cola de trabajo. Las colas con mayor prioridad (o con un valor entero mayor en el parámetro `priority`) se evalúan primero al asociarse con un mismo entorno de computación. La prioridad se determina en orden descendente; por ejemplo, a una cola de trabajos con una prioridad valor de 10 se le dará preferencia por encima de una cola de trabajos cuyo valor de prioridad sea 1. Todos los entornos de cómputo deben ser Amazon EC2 (`EC2`o`SPOT`) o Fargate (o). `FARGATE` `FARGATE_SPOT` Los entornos de cómputo de Amazon EC2 y Fargate no se pueden mezclar.

Tipo: entero

Obligatorio: sí

Política de programación

`schedulingPolicyArn`

El nombre de recurso de Amazon (ARN) de la política de programación para la cola de trabajo. Las colas de trabajos que no tienen una política de programación se programan según el modelo FIFO (primero en entrar, primero en salir). Una vez que una cola de trabajos tiene una política de programación, se puede reemplazar, pero no se puede eliminar. Una cola de trabajos sin una política de programación se programa como una cola de trabajos FIFO y no se le puede agregar una política de programación. Las colas de trabajos con una política de programación pueden tener un máximo de 500 identificadores de reparto justo activos. Cuando se alcanza el límite, no se admite la presentación de ningún trabajo que añada un nuevo identificador de reparto justo.

Tipo: cadena

Requerido: no

Estado

`state`

Estado de la cola de trabajos. Si el estado es `ENABLED` (el valor predeterminado), puede aceptar trabajos. Si el estado de la cola de trabajos es `DISABLED`, no se pueden agregar trabajos nuevos a la cola, pero se pueden finalizar los trabajos que ya se encuentran en la cola.

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

Orden del entorno de computación

`computeEnvironmentOrder`

El conjunto de entornos de computación asignado a una cola de trabajos y el orden de cada uno en función del otro. El programador de trabajos utiliza este parámetro para determinar qué entorno de computación debería ejecutar un trabajo específico. Los entornos de computación deben estar en estado `VALID` antes de asociarlos a una cola de trabajos. Puede asociar hasta tres entornos de computación con una cola de trabajos. Todos los entornos de cómputo deben ser Amazon EC2 (`EC2oSPOT`) o Fargate (o). `FARGATE` `FARGATE_SPOT` Los entornos de cómputo de Amazon EC2 y Fargate no se pueden mezclar.

> **Note**
>
> Todos los entornos informáticos asociados a una cola de trabajos deben compartir la misma arquitectura. AWS Batch no admite la combinación de tipos de arquitectura de entornos de cómputo en una sola cola de trabajos.

Tipo: matriz de objetos ComputeEnvironmentOrder

Obligatorio: sí

`computeEnvironment`

El Nombre de recurso de Amazon (ARN) del entorno de computación.

Tipo: cadena

Obligatorio: sí

`order`

El orden del entorno de computación. Los entornos de computación se intentan en orden ascendente. Por ejemplo, si hay dos entornos de computación asociados a una cola de trabajos, se intenta primero colocar el trabajo del entorno de computación del valor entero `order` más bajo.

Etiquetas

`tags`

Etiquetas de pares clave-valor para asociarlas a la cola de trabajos. Para obtener más información, consulte Etiquetado de los recursos de AWS Batch.

Tipo: mapa de cadena a cadena

Obligatorio: no

Programación de trabajos

El programador de AWS Batch evalúa cuándo, dónde y cómo ejecutar los trabajos que se envían a una cola de trabajos. Si no especifica una política de programación al crear una cola de trabajos, el programador de trabajos de AWS Batch utilizará de forma predeterminada la estrategia de «primero en entrar, primero en salir» (FIFO). Una estrategia FIFO puede provocar que los trabajos importantes se queden «atascados» con respecto a los trabajos que se enviaron anteriormente. Al especificar una política de programación diferente, puede asignar los recursos informáticos de acuerdo con sus necesidades específicas.

> **Note**
>
> Si desea programar el orden específico en el que se ejecutan los trabajos, utilice el parámetro `dependsOn` en SubmitJob para especificar las dependencias de cada trabajo.

Si crea una política de programación y la adjuntas a una cola de trabajos, se activará la programación de participaciones justas. Si la cola de trabajos tiene una política de programación, la política de programación determina el orden en que se ejecutan los trabajos. Para obtener más información, consulte Políticas de programación.

Compartir identificadores

Puede usar los identificadores de recursos compartidos para etiquetar los trabajos y diferenciar entre usuarios y cargas de trabajo. El programador de AWS Batch rastrea el uso de cada identificador de cuota justa mediante la fórmula *(`T * weightFactor`)*, donde *`T`* es el uso de la vCPU a lo largo del tiempo. El programador selecciona los trabajos con el menor uso a partir del identificador de recursos compartidos. Puede usar un identificador de cuota justa sin anularlo.

> **Note**
>
> Los identificadores de acciones son únicos dentro de una cola de trabajos y no se agregan entre las colas de trabajos.

Puede establecer la prioridad de programación para configurar el orden en que se ejecutan los trabajos en un identificador de recursos compartidos. Los trabajos con una prioridad de

programación más alta se programan primero. Si no especifica una política de programación, todos los trabajos que se envían a la cola de trabajos se programan en orden FIFO. Al enviar trabajos, no puede especificar identificadores de cuota ni prioridades de programación.

> **Note**
>
> Los recursos informáticos adjuntos se asignan por igual entre todos los identificadores de recursos compartidos, a menos que se anulen de forma explícita.

Programación de participaciones justas

La programación de participaciones justas proporciona un conjunto de controles para ayudar a programar los trabajos.

> **Note**
>
> Para obtener más información acerca de los parámetros de política de programación, consulte Parámetros de la política de programación.

- Segundos de caída de participación: el período de tiempo (en segundos) que el programador AWS Batch utiliza para calcular un porcentaje de participación justa para cada identificador de participación justa. Un valor de cero indica que solo se mide el uso actual. Un tiempo de decaimiento más largo da más peso al tiempo.

  > **Note**
  >
  > El período de tiempo de descomposición se calcula de la siguiente manera: *`shareDecaySeconds + OrderMinutes`* donde *`OrderMinutes`* es el tiempo del pedido en minutos.

- Reserva de computación: evita que los trabajos de un único identificador de recurso compartido consuman todos los recursos adjuntos a la cola de trabajos. La proporción reservada es *`computeReservation/100)^ActiveFairShares`*, donde ActiveFairShares es el número de identificadores activos de la cuota justa.

> **Note**
>
> Si un identificador de recursos compartidos tiene trabajos en un estado `SUBMITTED`, `PENDING`, `RUNNABLE`, `STARTING`, o `RUNNING` se considera un identificador de recurso compartido activo. Una vez transcurrido el período de inactividad, el identificador de acciones se considera inactivo.

- Factor de ponderación: el factor de ponderación para el identificador de cuota justa. El valor predeterminado es 1. Un valor inferior permite ejecutar los trabajos del identificador de recurso compartido o proporciona tiempo de ejecución adicional al identificador de recurso compartido. Por ejemplo, los trabajos que utilizan un identificador de recurso compartido con un factor de peso de 0,125 (1/8) obtienen ocho veces los recursos de computación de los trabajos que utilizan un identificador de recurso compartido con un factor de ponderación de 1.

  > **Note**
  >
  > Solo necesita definir este atributo cuando necesite actualizar el factor de ponderación predeterminado de 1.

Entorno de computación

Las colas de trabajos se asignan a uno o varios entornos de computación. Los entornos de computación incluyen instancias de contenedor de Amazon ECS que se utilizan para ejecutar trabajos por lotes en contenedores. Un entorno de computación específico también se puede asignar a una o más de una cola de trabajos. Dentro de una cola de trabajos, los entornos de computación asociados tienen cada uno un orden que utiliza el programador para determinar dónde se ejecutarán los trabajos que están listos para ejecutarse. Si el entorno de computación tiene un estado de `VALID` y tiene recursos gratuitos, el trabajo se programa para una instancia de contenedor dentro de dicho entorno de computación. Si el entorno de computación tiene un estado de `INVALID` o no puede proporcionar un recurso de computación apropiado, el programador intenta ejecutar el trabajo en el siguiente entorno de computación.

Temas

Entornos de computación administrados

Puede utilizar un entorno informático gestionado para AWS Batch gestionar la capacidad y los tipos de instancia de los recursos informáticos del entorno. Esto se basa en la especificación del recurso de cálculo que usted defina al crear el entorno de computación. Puede optar por utilizar Instancias bajo demanda de Amazon EC2 o Instancias de spot de Amazon EC2. O bien, también puede utilizar

la capacidad de Fargate y Fargate Spot en su entorno de computación administrado. Si utiliza instancias de spot, también puede establecer un precio máximo. De este modo, las instancias de spot solo se lanzan cuando el precio de oferta de spot está por debajo de un determinado porcentaje del precio bajo demanda.

> **Important**
>
> Las instancias Fargate Spot no son compatibles con. Windows containers on AWS Fargate Se bloqueará una cola de trabajos si se envía un FargateWindows trabajo a una cola de trabajos que solo utilice entornos de cómputo Fargate Spot.

Los entornos de computación gestionados lanzan instancias de Amazon EC2 en la VPC y las subredes que especifique y, a continuación, las registran en un clúster de Amazon ECS. Las instancias de Amazon EC2 necesitan acceso de red externo para comunicarse con el punto de conexión de servicio de Amazon ECS. Algunas subredes no proporcionan direcciones IP públicas a las instancias de Amazon EC2. Si las instancias de Amazon EC2 no tienen una dirección IP pública, deberán utilizar traducción de direcciones de red (NAT) para obtener este acceso. Para obtener información, consulte Gateways NAT en la Guía del usuario de Amazon VPC. Para obtener más información acerca de cómo crear una VPC, consulte Creación de una nube virtual privada (VPC) .

De forma predeterminada, los entornos informáticos AWS Batch gestionados utilizan una versión reciente y aprobada de la AMI optimizada de Amazon ECS para los recursos informáticos. Sin embargo, es posible que desee crear sus propias AMI para utilizarlas en sus entornos de computación administrados por varias razones. Para obtener más información, consulte AMI de recursos de computación.

> **Note**
>
> AWS Batch no actualiza automáticamente las AMI en un entorno informático una vez creado. Por ejemplo, no actualiza las AMI en su entorno de computación cuando se lanza una versión más nueva de la AMI optimizada de Amazon ECS. Usted es responsable de la administración del sistema operativo invitado. Esto incluye actualizaciones y parches de seguridad. También es el responsable de cualquier otra utilidad o software de aplicaciones que se instale en los recursos de computación. Hay dos maneras de utilizar una AMI nueva para sus AWS Batch trabajos. El método original consiste en completar estos pasos:
>
> 1. Cree un nuevo entorno de computación con la nueva AMI.

2. Añada el entorno de computación a una cola de trabajos existente.
3. Quite el entorno de computación anterior de la cola de trabajos.
4. Elimine el entorno de computación anterior.

En abril de 2022, AWS Batch se agregó un soporte mejorado para actualizar los entornos de cómputo. Para obtener más información, consulte Actualizar entornos informáticos. Para usar la actualización mejorada de los entornos de computación para actualizar las AMI, siga estas reglas:

- No definas el parámetro service role (`serviceRole`) o establézcalo en el rol AWSServiceRoleForBatchvinculado al servicio.
- Defina el parámetro de estrategia de asignación (`allocationStrategy`) en `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` o `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Defina el parámetro de actualización a la última versión de la imagen (`updateToLatestImageVersion`) en `true`.
- No especifique un ID de AMI en `imageId`, `imageIdOverride` (en `ec2Configuration`) o en la plantilla de lanzamiento (`launchTemplate`). En ese caso, AWS Batch selecciona la última AMI optimizada para Amazon ECS compatible AWS Batch en el momento en que se inicia la actualización de la infraestructura. Como alternativa, puede especificar el ID de la AMI en los parámetros `imageId` o `imageIdOverride`, o la plantilla de lanzamiento identificada por las propiedades `LaunchTemplate`. El cambio de cualquiera de estas propiedades inicia una actualización de la infraestructura. Si el ID de AMI se especifica en la plantilla de lanzamiento, no se puede reemplazar especificando un ID de AMI en los parámetros `imageId` o `imageIdOverride`. Solo se puede reemplazar especificando una plantilla de lanzamiento diferente. O bien, si la versión de la plantilla de lanzamiento está configurada en `$Default` o `$Latest`, configurando una nueva versión predeterminada para la plantilla de lanzamiento (si es `$Default`) o añadiendo una nueva versión a la plantilla de lanzamiento (si es `$Latest`).

Si se siguen estas reglas, cualquier actualización que inicie una actualización de la infraestructura hará que se vuelva a seleccionar el ID de la AMI. Si la configuración `version` en la plantilla de lanzamiento (`launchTemplate`) toma el valor `$Latest` o `$Default`, se evaluará la versión más reciente o predeterminada de la plantilla de lanzamiento en el

momento de la actualización de la infraestructura, incluso si `launchTemplate` no se ha actualizado.

Consideraciones a la hora de crear trabajos paralelos de varios nodos

AWS Batch recomienda crear entornos de cómputo dedicados para ejecutar trabajos paralelos (MNP) con varios nodos y trabajos que no sean MNP. Esto se debe a la forma en que se crea la capacidad informática en su entorno de computación administrado. Al crear un nuevo entorno de computación gestionado, si especifica un valor `minvCpu` superior a cero, entonces AWS Batch crea un grupo de instancias para usarlo únicamente con trabajos que no son de MNP. Si se envía un trabajo paralelo de varios nodos, AWS Batch crea una nueva capacidad de instancia para ejecutar los trabajos paralelos de varios nodos. En los casos en los que haya trabajos paralelos de un solo nodo y de varios nodos ejecutándose en el mismo entorno informático en el que se establezca un `maxvCpus` valor `minvCpus` o, si los recursos informáticos necesarios no están disponibles, AWS Batch esperará a que finalicen los trabajos actuales antes de crear los recursos informáticos necesarios para ejecutar los nuevos trabajos.

Entornos de computación no administrados

En un entorno de informática no administrado, usted gestiona sus propios recursos de computación. Debe verificar que la AMI que utiliza para sus recursos de computación cumple la especificación de la AMI de instancia de contenedor de Amazon ECS. Para obtener más información, consulte Especificaciones de AMI de recursos de computación y Cómo crear una AMI de recursos informáticos.

> **Note**
>
> AWS Los recursos de Fargate no son compatibles con los entornos informáticos no gestionados.

Después de crear el entorno informático no gestionado, utilice la operación de la DescribeComputeEnvironmentsAPI para ver los detalles del entorno informático. Encuentre el clúster de Amazon ECS asociado al entorno y, a continuación, lance manualmente las instancias de contenedor en ese clúster de Amazon ECS.

El siguiente AWS CLI comando también proporciona el ARN del clúster Amazon ECS.

```
$ aws batch describe-compute-environments \
    --compute-environments unmanagedCE \
    --query "computeEnvironments[].ecsClusterArn"
```

Para obtener más información, consulte Lanzamiento de una instancia de contenedor de Amazon ECS en la Guía del desarrollador de Amazon Elastic Container Service. Al lanzar los recursos de computación, especifique el ARN del clúster de Amazon ECS que los recursos deben registrar con los siguientes datos de usuario de Amazon EC2. *ecsClusterArn*Sustitúyalo por el ARN del clúster que obtuvo con el comando anterior.

```
#!/bin/bash
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

AMI de recursos de computación

De forma predeterminada, los entornos informáticos AWS Batch gestionados utilizan una versión reciente y aprobada de la AMI optimizada de Amazon ECS para los recursos informáticos. Sin embargo, es posible que desee crear sus propias AMI; para utilizarlas en sus entornos de computación administrados y no administrados. Si necesita alguna de las siguientes opciones, le recomendamos que cree su propia AMI:

- Aumentar el tamaño de su almacenamiento AMI; raíz o volúmenes de datos
- Agregar volúmenes de almacenamiento de instancias para los tipos de instancias de Amazon EC2; admitidos
- Personalizar el agente de contenedor de Amazon ECS
- Personalización de Docker
- Configurar una AMI de carga de trabajo de GPU que permite a los contenedores acceder a hardware de GPU en los tipos de instancias de Amazon EC2 admitidos

> **Note**
>
> Una vez creado un entorno informático, AWS Batch no actualiza las AMI del entorno informático. AWS Batch tampoco actualiza las AMI de su entorno informático cuando hay disponible una versión más reciente de la AMI optimizada para Amazon ECS. Usted es responsable de la administración del sistema operativo invitado. Esto incluye actualizaciones y parches de seguridad. También es el responsable de cualquier otra utilidad o software de

aplicaciones que se instale en los recursos de computación. Para usar una AMI nueva para sus AWS Batch trabajos, haga lo siguiente:

1. Cree un nuevo entorno de computación con la nueva AMI.
2. Añada el entorno de computación a una cola de trabajos existente.
3. Quite el entorno de computación anterior de la cola de trabajos.
4. Elimine el entorno de computación anterior.

En abril de 2022, AWS Batch se agregó un soporte mejorado para actualizar los entornos de cómputo. Para obtener más información, consulte Actualizar entornos informáticos. Para usar la actualización mejorada de los entornos de computación para actualizar las AMI, siga estas reglas:

- No definas el parámetro service role (`serviceRole`) o establézcalo en el rol AWSServiceRoleForBatchvinculado al servicio.
- Defina el parámetro de estrategia de asignación (`allocationStrategy`) en `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` o `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Defina el parámetro de actualización a la última versión de la imagen (`updateToLatestImageVersion`) en `true`.
- No especifique un ID de AMI en `imageId`, `imageIdOverride` (en `ec2Configuration`) o en la plantilla de lanzamiento (`launchTemplate`). Si no especifica un ID de AMI, AWS Batch selecciona la última AMI optimizada para Amazon ECS AWS Batch compatible en el momento en que se inicia la actualización de la infraestructura. También, puede especificar el ID de AMi en los parámetros `imageId` o `imageIdOverride`. También puede especificar la plantilla de lanzamiento que se identifica mediante las propiedades de `LaunchTemplate`. El cambio de cualquiera de estas propiedades inicia una actualización de la infraestructura. Si el ID de AMI se especifica en la plantilla de lanzamiento, este se puede reemplazar especificando un ID de AMI en los parámetros `imageId` o `imageIdOverride`. El ID de AMI solo se puede reemplazar especificando una plantilla de lanzamiento diferente. Si la versión de la plantilla de lanzamiento está configurada en `$Default` o `$Latest`, el ID de AMI puede sustituirse configurando una nueva versión predeterminada para la plantilla de lanzamiento (si es `$Default`) o añadiendo una nueva versión a la plantilla de lanzamiento (si es `$Latest`).

Si se siguen estas reglas, cualquier actualización que inicie una actualización de la infraestructura hará que se vuelva a seleccionar el ID de la AMI. Si la configuración `version` en la plantilla de lanzamiento (`launchTemplate`) toma el valor `$Latest` o `$Default`, se evaluará la versión más reciente o predeterminada de la plantilla de lanzamiento en el momento de la actualización de la infraestructura, incluso si `launchTemplate` no se ha actualizado.

Temas

Especificaciones de AMI de recursos de computación

La especificación AMI de recursos AWS Batch informáticos básica consiste en lo siguiente:

Obligatorio

- Una moderna distribución de Linux que ejecuta al menos la versión 3,10 de kernel de Linux en una AMI; de tipo de virtualización HVM. No se admiten los contenedores de Windows.

  Important

  Los trabajos paralelos de varios nodos solo se pueden ejecutar en recursos de computación que se hayan lanzado en una instancia de Amazon Linux con el paquete `ecs-init` instalado. Recomendamos utilizar la AMI optimizada para Amazon ECS predeterminada al crear el entorno de computación. Para ello, no especifique una AMI personalizada. Para obtener más información, consulte Trabajos paralelos de varios nodos.

- El agente de contenedor de Amazon ECS. Le recomendamos que utilice la última versión de . Para obtener más información, consulte Instalar el agente contenedor de instancia de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

- El controlador de registros `awslogs` debe especificarse como un controlador de registros disponible con la variable de entorno `ECS_AVAILABLE_LOGGING_DRIVERS` cuando el agente de contenedor de Amazon ECS se inicia. Para obtener más información, consulte Configuración del agente de contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.
- Un daemon de Docker que ejecuta al menos la versión 1.9, y cualquier dependencia de tiempo de ejecución de Docker. Para obtener más información, consulte Check runtime dependencies en la documentación de Docker.

> **Note**
>
> Le recomendamos la versión de Docker que se envía y se prueba con la versión de agente de Amazon ECS correspondiente que está utilizando. Amazon ECS proporciona un registro de cambios para la variante Linux de la AMI optimizada para Amazon ECS on. GitHub Para obtener más información, consulte Changelog (Registro de cambios).

Recomendado

- Un proceso de inicialización y nanny para ejecutar y monitorear el agente de Amazon ECS. La AMI optimizada para Amazon ECS usa el proceso upstart `ecs-init` y otros sistemas operativos pueden utilizar `systemd`. Para obtener más información y ejemplos, consulte Scripts de configuración de datos de usuario de instancias de contenedor de ejemplo en la Guía para desarrolladores de Amazon Elastic Container Service. Para obtener más información al respecto`ecs-init`, consulte el proyecto en. `ecs-init` GitHub Los entornos de computación administrados requieren, como mínimo, que el agente de Amazon ECS se inicie al arrancar. Si el agente de Amazon ECS no se está ejecutando en su recurso informático, no podrá aceptar trabajos de AWS Batch.

Las AMI optimizadas para Amazon ECS están preconfiguradas con estos requisitos y recomendaciones. Recomendamos que utilice la AMI optimizada para Amazon ECS o una AMI de Amazon Linux con el paquete `ecs-init` instalado para sus recursos de computación. Elija otra AMI si la aplicación requiere un sistema operativo específico o una versión de Docker que aún no esté disponible en estas AMI. Para obtener más información, consulte la AMI optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

Cómo crear una AMI de recursos informáticos

Es posible crear sus propias AMI de recursos informáticos personalizadas para usarlas en entornos informáticos administrados y sin administrar. Para obtener instrucciones, consulte la Especificaciones de AMI de recursos de computación. Después de crear la AMI personalizada, puede crear un entorno informático que utilice dicha AMI al que puede asociar una cola de trabajos. Por último, comience a enviar trabajos a esa cola.

Para crear una AMI de recursos informáticos personalizada

1. Elija una AMI de base como punto de partida. Las AMI de base deben utilizar virtualización HVM. Las AMI de base no pueden ser una AMI de Windows.

   > **Note**
   >
   > La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno informático. Por ejemplo, si su entorno informático utiliza tipos de instancias A1, la AMI de recursos informáticos que elija debe admitir instancias Arm. Amazon ECS ofrece versiones x86 y Arm de la AMI Amazon Linux 2 optimizada para Amazon ECS. Para obtener más información, consulte la sección sobre AMI Amazon Linux 2 optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

   La AMI Amazon Linux 2 optimizada para Amazon ECS es la predeterminada para recursos informáticos en entornos informáticos administrados. La AMI Amazon Linux 2 optimizada para Amazon ECS está preconfigurada y probada en AWS Batch por los ingenieros de AWS. Se trata de una AMI mínima con la que puede empezar y que permite que sus recursos informáticos se ejecuten en AWS rápidamente. Para obtener más información, consulte AMI Linux optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

   También puede elegir otra variante de Amazon Linux 2 e instalar el paquete `ecs-init` con los siguientes comandos. Para obtener más información, consulte Instalar el agente contenedor en una instancia de EC2 Linux 2 de Amazon en la Guía del desarrollador de Amazon Elastic Container Service:

   ```
   $ sudo amazon-linux-extras disable docker
   $ sudo amazon-linux-extras install ecs-init
   ```

Por ejemplo, si desea ejecutar cargas de trabajo de GPU en los recursos informáticos de AWS Batch, puede empezar con la AMI de aprendizaje profundo de Amazon Linux como punto de partida. A continuación, configure la AMI para ejecutar los trabajos de AWS Batch. Para obtener más información, consulte Cómo utilizar una AMI de carga de trabajo de GPU.

> **Important**
>
> Puede elegir una AMI básica que no sea compatible con el paquete `ecs-init`. Sin embargo, si lo hace, debe configurar una forma de iniciar el agente Amazon ECS durante el arranque y mantenerlo en funcionamiento. También puede ver varios ejemplos de scripts de configuración de datos de usuario que utilizan `systemd` para iniciar y supervisar el agente contenedor de Amazon ECS. Para obtener más información, consulte Scripts de configuración de datos de usuario de instancias de contenedor de ejemplo en la Guía para desarrolladores de Amazon Elastic Container Service.

2. Lance una instancia desde su AMI de base seleccionada con las opciones de almacenamiento adecuadas para su AMI. Es posible configurar el tamaño y la cantidad de volúmenes de Amazon EBS o de almacenamiento de instancias, si el tipo de instancia que ha seleccionado es compatible con ellos. Para obtener más información, consulte Lanzamiento de instancias y Almacén de instancias de Amazon EBS en la Guía del usuario de Amazon EC2 para instancias de Linux.
3. Conéctese a la instancia con SSH y lleve a cabo las tareas de configuración que sean necesarias. Esto puede incluir una de las siguientes etapas, o todas:

   - Instalación del agente de contenedor de Amazon ECS. Para obtener más información, consulte Instalar el agente contenedor de instancia de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.
   - Configurar un script para formatear volúmenes de almacén de instancias.
   - Añadir un volumen de almacén de instancias o sistemas de archivos de Amazon EFS al archivo `/etc/fstab` para que puedan montarse al arrancar.
   - Configurar opciones de Docker, como activar la depuración o ajustar el tamaño de la imagen de base.
   - Instalar paquetes o copiar archivos.

Para más información, consulte Conexión a la instancia de Linux mediante SSH en la Guía del usuario de Amazon EC2 para instancias de Linux.

4. Si ha iniciado el agente contenedor de Amazon ECS en su instancia, debe detenerlo y eliminar cualquier archivo de comprobación de datos persistente antes de crear la AMI. De lo contrario, si no lo hace, el agente no se iniciará en las instancias que se lancen desde su AMI.

   a. Detenga el agente de contenedor de Amazon ECS.

      - AMI de Amazon Linux 2 optimizada para Amazon ECS:

        ```
        sudo systemctl stop ecs
        ```

      - AMI de Amazon Linux optimizada para Amazon ECS:

        ```
        sudo stop ecs
        ```

   b. Elimine los archivos de comprobación de datos persistentes. De forma predeterminada, estos archivos se ubican en el directorio `/var/lib/ecs/data/`. Use el siguiente comando para eliminar estos archivos, si los hay.

      ```
      sudo rm -rf /var/lib/ecs/data/*
      ```

5. Cree una nueva AMI; desde su instancia en ejecución. Para más información, consulte Creación de una AMI de Linux con respaldo en Amazon EBS en la Guía del usuario de Amazon EC2 para instancias de Linux.

Para utilizar la AMI nueva con AWS Batch

1. Cuando se haya creado una nueva AMI, cree un nuevo entorno informático con la nueva AMI. Para ello elija el tipo de imagen e introduzca el ID de AMI personalizado en la casilla deID de imagen al crear el entorno informático de AWS Batch. Para obtener más información, consulte the section called “Para crear un entorno de computación gestionado con los recursos de EC2”.

   > **Note**
   >
   > La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno informático. Por

ejemplo, si su entorno informático utiliza tipos de instancias A1, la AMI de recursos informáticos que elija debe admitir instancias Arm. Amazon ECS ofrece versiones x86 y Arm de la AMI Amazon Linux 2 optimizada para Amazon ECS. Para obtener más información, consulte la sección sobre AMI Amazon Linux 2 optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

2. Cree una cola de trabajos y asocie el entorno informático nuevo. Para obtener más información, consulte Cómo crear de una cola de trabajos.

> **Note**
>
> Todos los entornos informáticos asociados a una cola de trabajos deben compartir la misma arquitectura. AWS Batch no admite la mezcla de tipos de arquitecturas de entorno informático en una sola cola de trabajos.

3. (Opcional) Envíe un trabajo de muestra a la cola de trabajos nueva. Para obtener más información, consulte Ejemplos de definiciones de trabajo, Creación de una definición de trabajo de un solo nodo y Enviar un trabajo.

Cómo utilizar una AMI de carga de trabajo de GPU

Para ejecutar cargas de trabajo de GPU en los recursos informáticos de AWS Batch, debe utilizar una AMI compatible con GPU. Para obtener más información, consulte Trabajar con GPU en Amazon ECS y AMI optimizadas para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

En entornos informáticos administrados, si el entorno informático especifica cualquier familia de instancia o tipo de instancia de `p2`, `p3`, `p4`, `p5`, `g3`, `g3s`, `g4`, o `g5` utiliza una AWS Batch AMI optimizada para GPU de Amazon ECS.

En entornos informáticos no administrados, se recomienda una AMI optimizada para GPU de Amazon ECS. Puede utilizar las operaciones AWS Command Line Interface o AWS Systems Manager Parameter Store GetParameter, GetParameters y GetParametersByPath para recuperar los metadatos de las AMI optimizadas para GPU de Amazon ECS recomendadas.

> **Note**
>
> La familia de instancias p5 solo se admite en versiones iguales o posteriores a la AMI optimizada para GPU de Amazon ECS `20230912` y son incompatibles los tipos de instancia p2 y g2. Si necesita usar instancias p5, asegúrese de que su entorno informático no instancias p2 o g2 y utilice la última AMI de Batch predeterminada. La creación de un nuevo entorno informático utilizará la AMI más reciente, pero si actualiza su entorno informático para incluir p5, puede asegurarse de utilizar la AMI más reciente configurando `updateToLatestImageVersion` en `true` en propiedades `ComputeResource`. Para obtener más información sobre la compatibilidad de AMI con instancias de GPU, consulte Working with GPUs on Amazon ECS en Amazon Elastic Container Service Developer Guide.

Los siguientes ejemplos muestran cómo utilizar el comando GetParameter.

AWS CLI

```
$ aws ssm get-parameter --name /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended \
                        --region us-east-2 --output json
```

El resultado incluye la información de AMI en el parámetro de valor `Value`:.

```
{
    "Parameter": {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended",
        "LastModifiedDate": 1555434128.664,
        "Value": "{\"schema_version\":1,\"image_name\":\"amzn2-ami-ecs-gpu-
hvm-2.0.20190402-x86_64-ebs\",\"image_id\":\"ami-083c800fe4211192f\",\"os\":\"Amazon
 Linux 2\",\"ecs_runtime_version\":\"Docker version 18.06.1-ce\",\"ecs_agent_version
\":\"1.27.0\"}",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended"
    }
}
```

Python

```
from __future__ import print_function

import json
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameter(Name='/aws/service/ecs/optimized-ami/amazon-linux-2/
gpu/recommended')
jsonVal = json.loads(response['Parameter']['Value'])
print("image_id   = " + jsonVal['image_id'])
print("image_name = " + jsonVal['image_name'])
```

El resultado solo incluye el ID y el nombre de AMI:

```
image_id   = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

En los siguientes ejemplos se muestra el uso de GetParameters.

AWS CLI

```
$ aws ssm get-parameters --names  /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_name \
                                  /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_id \
                          --region us-east-2 --output json
```

El resultado incluye todos los metadatos de cada uno de los parámetros:

```
{
    "InvalidParameters": [],
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
            "LastModifiedDate": 1555434128.749,
            "Value": "ami-083c800fe4211192f",
```

```
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
            "LastModifiedDate": 1555434128.712,
            "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
        }
    ]
}
```

Python

```
from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters(
            Names=['/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name',
                   '/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id'])
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

El resultado incluye el ID de AMI y el nombre de AMI, con la ruta completa de los nombres.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
 ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Los siguientes ejemplos muestran cómo utilizar el comando GetParametersByPath.

AWS CLI

```
$ aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended \
                                --region us-east-2 --output json
```

El resultado incluye todos los metadatos de todos los parámetros de la ruta especificada.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_agent_version",
            "LastModifiedDate": 1555434128.801,
            "Value": "1.27.0",
            "Version": 8,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_agent_version"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_runtime_version",
            "LastModifiedDate": 1548368308.213,
            "Value": "Docker version 18.06.1-ce",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_runtime_version"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
            "LastModifiedDate": 1555434128.749,
            "Value": "ami-083c800fe4211192f",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
```

```
            "LastModifiedDate": 1555434128.712,
            "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
os",
            "LastModifiedDate": 1548368308.143,
            "Value": "Amazon Linux 2",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/os"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
schema_version",
            "LastModifiedDate": 1548368307.914,
            "Value": "1",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/schema_version"
        }
    ]
}
```

Python

```
from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters_by_path(Path='/aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended')
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

El resultado incluye los valores de todos los nombres de parámetros de la ruta especificada, con la ruta completa de los nombres.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_agent_version =
 1.27.0
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_runtime_version =
 Docker version 18.06.1-ce
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
 ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/os = Amazon Linux 2
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/schema_version = 1
```

Para obtener más información, consulte Retrieving Amazon ECS-Optimized AMI Metadata en la Guía para desarrolladores de Amazon Elastic Container Service.

Obsolescencia de Amazon Linux

La AMI de Amazon Linux (también llamada Amazon Linux 1) llegó al final de su vida útil el 31 de diciembre de 2023. AWS Batch ha dejado de dar soporte a la AMI de Amazon Linux, ya que no recibirá actualizaciones de seguridad ni correcciones de errores a partir del 1 de enero de 2024. Para obtener más información sobre Amazon Linux end-of-life, consulte las preguntas frecuentes de AL.

Le recomendamos que actualice los entornos informáticos actuales basados en Amazon Linux a Amazon Linux 2023 para evitar interrupciones imprevistas en la carga de trabajo y que siga recibiendo actualizaciones de seguridad y de otro tipo.

Es posible que sus entornos informáticos que utilizan la AMI de Amazon Linux sigan funcionando después de la end-of-life fecha del 31 de diciembre de 2023. Sin embargo, estos entornos informáticos ya no recibirán nuevas actualizaciones de software, parches de seguridad ni correcciones de errores AWS. Posteriormente, es su responsabilidad mantener estos entornos informáticos en la AMI de Amazon Linux end-of-life. Recomendamos migrar los entornos AWS Batch informáticos a Amazon Linux 2023 o Amazon Linux 2 para mantener un rendimiento y una seguridad óptimos.

Para obtener ayuda para migrar AWS Batch de la AMI de Amazon Linux a Amazon Linux 2023 o Amazon Linux 2, consulte Actualización de entornos informáticos - AWS Batch.

Compatibilidad con las plantillas de lanzamiento

AWS Batch admite el uso de plantillas de lanzamiento de Amazon EC2 en los entornos informáticos EC2. Con las plantillas de lanzamiento, puede modificar la configuración predeterminada de los recursos informáticos de AWS Batch sin necesidad de crear AMI personalizadas.

> **Note**
>
> Los recursos de Fargate AWS no admiten plantillas de lanzamiento.

Es necesario crear una plantilla de lanzamiento para poder asociarla a un entorno informático. Puede crear una plantilla de lanzamiento en la consola de Amazon EC2. O bien, puede usar el AWS CLI o un SDK AWS. Por ejemplo, el siguiente archivo JSON representa una plantilla de lanzamiento que redimensiona el volumen de datos de Docker para la predeterminada del recurso informático AMI de AWS Batch y también lo configura como cifrado.

```
{
    "LaunchTemplateName": "increase-container-volume-encrypt",
    "LaunchTemplateData": {
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "Encrypted": true,
                    "VolumeSize": 100,
                    "VolumeType": "gp2"
                }
            }
        ]
    }
}
```

Puede crear la plantilla de lanzamiento anterior guardando el JSON en un archivo denominado `lt-data.json` y ejecutando el siguiente comando de la AWS CLI.

```
aws ec2 --region <region> create-launch-template --cli-input-json file://lt-data.json
```

Para obtener más información sobre las plantillas de lanzamiento, consulte Lanzar una instancia desde una plantilla de lanzamiento en la Guía del usuario de Amazon EC2 para instancias de Linux.

Si utiliza una plantilla de lanzamiento para crear el entorno informático, puede mover los siguientes parámetros del entorno informático existente a la plantilla de lanzamiento:

> **Note**
>
> Supongamos que alguno de estos parámetros (excepto las etiquetas de Amazon EC2) se especifica tanto en la plantilla de lanzamiento como en la configuración del entorno de procesamiento. Entonces, los parámetros del entorno de cómputo tienen prioridad. Las etiquetas Amazon EC2 se combinan entre la plantilla de lanzamiento y la configuración del entorno de cómputo. Si se produce una colisión en la clave de la etiqueta, el valor de la configuración del entorno informático tiene prioridad.

- Pares de claves de Amazon EC2
- ID de la AMI de Amazon EC2
- ID de grupo de seguridad
- Etiquetas de Amazon EC2

Los siguientes parámetros de la plantilla de lanzamiento son ignorados por AWS Batch:

- Tipo de instancia (especifique los tipos de instancia que desea utilizar al crear el entorno informático)
- Rol de instancia (especifique el rol de instancia que desea utilizar al crear el entorno informático)
- Subredes de interfaz de red (especifique las subredes que desea utilizar al crear el entorno informático)
- Opciones del mercado de instancias (AWS Batch debe controlar la configuración de instancias de spot)
- Deshabilitar la terminación de la API (AWS Batch debe controlar el ciclo de vida de la instancia)

AWS Batch solo actualiza la plantilla de lanzamiento con una nueva versión de la plantilla de lanzamiento durante las actualizaciones de infraestructura. Para obtener más información, consulte Actualizar entornos informáticos.

Datos de usuario de Amazon EC2 en las plantillas de lanzamiento

Puede proporcionar datos de usuario de Amazon EC2 en la plantilla de lanzamiento que será ejecutada por cloud-init al lanzar las instancias. Los datos de usuario pueden realizar escenarios de configuración comunes, incluidos pero sin limitarse a los siguientes:

- Inclusión de usuarios o grupos
- Instalación de paquetes
- Creación de particiones y sistemas de archivos

Los datos de usuario de Amazon EC2 en las plantillas de lanzamiento deben estar en el formato archivo multiparte MIME. Esto se debe a que los datos de usuario se combinan con otros datos de usuario de AWS Batch necesarios para configurar los recursos de procesamiento. Puede combinar varios bloques de datos de usuario en un único archivo multiparte MIME. Por ejemplo, es posible que desee combinar un boothook de nube que configure el daemon de Docker con un script de shell de datos de usuario que escribe información de configuración para el agente de contenedor de Amazon ECS.

Si usa AWS CloudFormation, el tipo AWS: :CloudFormation: :Init puede utilizarse con el script auxiliar cfn-init para realizar escenarios de configuración comunes.

Un archivo multiparte MIME consta de los siguientes componentes:

- El tipo de contenido y declaración de límite de partes: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- La declaración de versión de MIME: `MIME-Version: 1.0`
- Uno o más bloques de datos de usuario, que contienen los siguientes componentes:
  - El límite de apertura, que señala el inicio de un bloque de datos de usuario: `--==BOUNDARY==`. Debe dejar en blanco la línea anterior a este límite.
  - La declaración de tipo de contenido para el bloque: `Content-Type: text/cloud-config; charset="us-ascii"`. Para obtener más información sobre los tipos de contenido, consulte la documentación de Cloud-Init. Debe dejar en blanco la línea que sigue a la declaración de tipo de contenido.
  - El contenido de los datos de usuario, por ejemplo, una lista de intérprete de comandos o políticas de `cloud-init`.

- El límite de cierre, que señala el final del archivo multiparte MIME: `--==BOUNDARY==--`. Debe dejar en blanco la línea anterior al límite de cierre.

A continuación, se muestra un ejemplo de un archivo multiparte MIME que puede utilizar para crear el suyo propio.

> **Note**
>
> Si añade datos de usuario a una plantilla de lanzamiento en la consola de Amazon EC2, puede pegarlos como texto sin formato. O bien, puede cargarlos desde un archivo. Si utiliza la AWS CLI o un SDK de AWS, primero debe codificar en `base64` los datos de usuario y enviar esa cadena como valor del parámetro `UserData` al llamar a CreateLaunchTemplate, como se muestra en el JSON siguiente.
>
> ```
> {
>     "LaunchTemplateName": "base64-user-data",
>     "LaunchTemplateData": {
>         "UserData":
>  "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWFzZS1jb250YWluZXItdm9sdW..."
>     }
> }
> ```

Ejemplos

- Ejemplo: montaje de un sistema de archivos de Amazon EFS existente
- Ejemplo: anulación de la configuración predeterminada del agente de contenedor de Amazon ECS
- Ejemplo: montar un sistema de archivos de Amazon FSx para Lustre

Ejemplo: montaje de un sistema de archivos de Amazon EFS existente

Example

Este archivo multiparte MIME de ejemplo configura el recurso informático para instalar el paquete `amazon-efs-utils` y montar un sistema de archivos de Amazon EFS existente en `/mnt/efs`.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="
```

```
--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults

--==MYBOUNDARY==--
```

Ejemplo: anulación de la configuración predeterminada del agente de contenedor de Amazon ECS

Example

Este archivo multiparte MIME de ejemplo anula la configuración de limpieza predeterminada de las imágenes de Docker de un recurso informático.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
echo ECS_IMAGE_CLEANUP_INTERVAL=60m >> /etc/ecs/ecs.config
echo ECS_IMAGE_MINIMUM_CLEANUP_AGE=60m >> /etc/ecs/ecs.config

--==MYBOUNDARY==--
```

Ejemplo: montar un sistema de archivos de Amazon FSx para Lustre

Example

Este archivo multiparte MIME de ejemplo configura el recurso informático para instalar el paquete `lustre2.10` de la biblioteca Extras y montar un FSx existente para el sistema de archivos de Lustre

en `/scratch` y con un nombre de montaje de `fsx`. Este ejemplo es para Amazon Linux 2. Para obtener instrucciones de instalación de otras distribuciones de Linux, consulte Instalación del cliente Lustre en la Guía del usuario de Amazon FSx para Lustre. Para obtener más información, consulte Montaje automático del sistema de archivos de Amazon FSx en la Guía del usuario de Amazon FSx para Lustre.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- file_system_id_01=fs-0abcdef1234567890
- region=us-east-2
- fsx_directory=/scratch
- amazon-linux-extras install -y lustre2.10
- mkdir -p ${fsx_directory}
- mount -t lustre ${file_system_id_01}.fsx.${region}.amazonaws.com@tcp:fsx
 ${fsx_directory}

--==MYBOUNDARY==--
```

En los miembros volumes y mountPoints de las propiedades del contenedor, se deben asignar los puntos de montaje al contenedor.

```
{
    "volumes": [
        {
            "host": {
                "sourcePath": "/scratch"
            },
            "name": "Scratch"
        }
    ],
    "mountPoints": [
        {
            "containerPath": "/scratch",
            "sourceVolume": "Scratch"
        }
    ],
}
```

Cómo crear un entorno de computación

Antes de poder ejecutar trabajos AWS Batch, debe crear un entorno informático. Puede crear un entorno informático gestionado en el que AWS Batch administre las instancias de Amazon EC2 o los recursos de AWS Fargate dentro del entorno en función de sus especificaciones. O bien, puede crear un entorno de computación no administrado en el que gestione la configuración de la instancia de Amazon EC2 dentro del entorno.

Important

Las instancias de spot de Fargate no se admiten en los siguientes escenarios:

- En contenedores de Amazon Linux con arquitectura ARM64.
- Windows containers on AWS Fargate

En estas situaciones, se bloqueará una cola de trabajos si un trabajo se envía a una cola de trabajos que solo utiliza entornos de computación Fargate Spot.

Contenido

Para crear un entorno informático gestionado con los recursos de AWS Fargate

1. Abra la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. Seleccione Crear.
5. Configure el entorno de computación.

> **Note**
>
> Los entornos de cómputo para los Windows containers on AWS Fargate trabajos deben tener al menos una vCPU.

a. Para la Configuración del entorno de cómputo, elija Fargate.

b. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

c. En el rol de servicio, elige un rol vinculado al servicio que permita al AWS Batch servicio realizar llamadas a las operaciones de AWS API requeridas en tu nombre. En este ejemplo, elija AWSServiceRoleForBatch. Para obtener más información, consulte Permisos de rol vinculados al servicio para AWS Batch.

d. (Opcional) Amplíe las Etiquetas. Para agregar una etiqueta, elija Add tag (Añadir etiqueta). Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta.

e. Seleccione Página siguiente.

6. Sección de Configuración de instancias:

a. (Opcional) En Utilizar la capacidad de Fargate Spot, active Fargate Spot. Para obtener información sobre Fargate Spot, consulte Uso de Amazon EC2 Spot y Fargate_SPOT.

b. En Máximo de CPU virtuales, seleccione la cantidad máxima de vCPUs admitida que su entorno de computación puede escalar horizontalmente, independientemente de la demanda de las colas de trabajos.

c. Seleccione Página siguiente.

7. Configure redes.

> **Important**
>
> Los recursos de computación de las deben obtener acceso para comunicarse con el punto de conexión del servicio de Amazon ECS. Esto puede ser a través de un punto de conexión de la VPC de la interfaz o a través de recursos de computación de las con direcciones IP públicas.

Para obtener más información acerca de los puntos de enlace de la VPC de la interfaz, consulte Puntos de enlace de la VPC de la interfaz de Amazon ECS (AWS PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.
Si no tiene configurado un punto de conexión de la VPC de la interfaz y los recursos de computación de las no tienen direcciones IP públicas, deberán utilizar traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulte Gateways NAT en la Guía del usuario de Amazon VPC. Para obtener más información, consulte the section called “Creación de una VPC”.

a. Para el ID de la nube privada virtual (VPC), elija una VPC en la que quiera lanzar sus instancias.

b. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, se escogen todas las subredes dentro de la VPC disponible.

> **Note**
>
> AWS Batch on Fargate no es compatible actualmente con las Zonas Locales. Para obtener más información, consulte los Clústeres de Amazon ECS en Local Zones, Wavelength Zones y AWS Outposts en la Guía para desarrolladores de Amazon Elastic Container Service.

c. En Grupos de seguridad, seleccione un su grupo de seguridad para asociarlo a las instancias. De forma predeterminada, se escoge el grupo de seguridad predeterminado para la VPC.

d. Seleccione Página siguiente.

8. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Para crear un entorno de computación gestionado con los recursos de EC2

1. Abre la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. Seleccione Crear.

5. Configure el entorno.

   a. Para la Configuración del entorno de computación, elija Amazon Elastic Compute Cloud (Amazon EC2).

   b. Para el Tipo de orquestación, seleccione Administrado.

   c. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

   d. (Opcional) En el rol de servicio, elige un rol vinculado al servicio que permita al AWS Batch servicio realizar llamadas a las operaciones de AWS API requeridas en tu nombre. En este ejemplo, elija AWSServiceRoleForBatch. Para obtener más información, consulte Permisos de rol vinculados al servicio para AWS Batch.

   e. En Instance role (Rol de instancia), elija si desea crear un perfil de instancia nuevo o utilizar uno ya existente que tenga asociados los permisos de IAM necesarios. Este perfil de instancia permite que las instancias de contenedor de Amazon ECS que se crean para su entorno de cómputo realicen llamadas a las operaciones de AWS API requeridas en su nombre. Para obtener más información, consulte Función de instancia de Amazon ECS. Si decide crear un perfil de instancia nuevo, se creará el rol requerido (`ecsInstanceRole`).

   f. (Opcional) Amplíe las Etiquetas.

   g. (Opcional) En el caso de las Etiquetas de EC2, seleccione Agregar etiqueta para añadir una etiqueta a los recursos que se lanzan en el entorno de computación. Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta.

   h. (Opcional) En Etiquetas, seleccione Agregar etiqueta. Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta.

      Para obtener más información, consulte Etiquetado de los recursos de AWS Batch.

   i. Seleccione Página siguiente.

6. Sección de Configuración de instancias:

   a. (Opcional) En Habilitar el uso de instancias de spot, active Spot. Para obtener más información, consulte Instancias de spot de .

   b. (Opcional) En Precio máximo % bajo demanda, seleccione el porcentaje máximo del precio que puede tener una instancia de spot en relación con su precio bajo demanda antes de que lancen las instancias. Por ejemplo, si el precio máximo es el 20%, el precio de spot de esa instancia de EC2 deberá ser inferior al 20% del precio bajo demanda que tenga en ese

momento. Siempre se paga el precio más bajo (de mercado) y nunca más que lo marcado por el porcentaje máximo. Si se deja este campo en blanco, el valor predeterminado es el 100% del precio bajo demanda.

c. (Solo Spot) En Rol de la flota de spot, seleccione un rol de IAM para la flota de spot de Amazon EC2 que quiera aplicar a su entorno de computación de spot. Si aún no tiene un rol de IAM para la flota de spot de Amazon EC2, primero debe crear uno. Para obtener más información, consulte Rol de flota de spot de Amazon EC2.

> **Important**
>
> Para etiquetar las instancias puntuales al crearlas, su función de IAM de Amazon EC2 Spot Fleet debe utilizar la política gestionada más reciente de SpotFleetTaggingRoleAmazonEC2. La política SpotFleetRole administrada de AmazonEC2 no tiene los permisos necesarios para etiquetar las instancias puntuales. Para obtener más información, consulte Instancias de spot no etiquetadas en el momento de su creación y the section called “Etiquetado de los recursos de ”.

d. En Mínimo de CPU virtuales, seleccione la cantidad mínima de vCPUs que mantiene el entorno de computación, independientemente de la demanda de las colas de trabajos.

e. En CPU virtuales deseadas, seleccione la cantidad de vCPU con las que el entorno de computación realiza lanzamientos. A medida que aumenta la demanda de la cola de trabajos, AWS Batch también puede incrementar la cantidad de vCPU en su entorno de computación y añadir instancias EC2, hasta alcanzar la cantidad máxima de vCPU. A medida que la demanda disminuye, AWS Batch puede reducir la cantidad de vCPU en su entorno de computación y eliminar instancias, hasta alcanzar la cantidad mínima de vCPU.

f. En Máximo de CPU virtuales, seleccione la cantidad máxima de vCPUs admitida que su entorno de computación puede escalar horizontalmente, independientemente de la demanda de las colas de trabajos.

g. En Tipos de instancias permitidos, elija los tipos de instancia de Amazon EC2 que se pueden lanzar. Se pueden especificar familias de instancias para lanzar cualquier tipo de instancia en esas familias (por ejemplo, `c5`, `c5n` o `p3`). O bien puede especificar tamaños específicos dentro de una familia (por ejemplo, `c5.8xlarge`). Los tipos de instancias metálicas no están en las familias de instancias. Por ejemplo, `c5` no incluye `c5.metal`. También puede seleccionar `optimal` para elegir tipos de instancias (de las familias de instancias C4, M4 y R4) que se correspondan con la demanda de las colas de trabajos.

> **Note**
>
> Cuando se crea un entorno de computación, los tipos de instancias que se seleccionen para dicho entorno de computación deben compartir la misma arquitectura. Por ejemplo, no se puede mezclar instancias x86 y ARM en el mismo entorno de computación.

> **Note**
>
> AWS Batch escalará las GPU en función de la cantidad requerida en sus colas de trabajos. Para utilizar la programación de GPU, el entorno de computación debe incluir tipos de instancia de las familias `p2`, `p3`, `p4`, `p5`, `g3`, `g3s`, `g4` o `g5`.

> **Note**
>
> Actualmente, `optimal` utiliza tipos de instancia de las familias de instancias C4, M4 y R4. Si Regiones de AWS no tiene tipos de instancias de esas familias de instancias, se utilizan los tipos de instancia de C5M5, y las familias de R5 instancias.

h. Expanda Configuración adicional.

i. (Opcional) En Grupo de ubicación, introduzca un nombre de grupo de ubicación para agrupar los recursos en el entorno de computación.

j. (Opcional) En Par de claves EC2, elija un par de claves pública y privada como credenciales de seguridad cuando se conecte a la instancia. Para obtener más información sobre pares de claves de Amazon EC2, consulte pares de claves de Amazon EC2 e instancias de Linux.

k. Para Allocation strategy (Estrategia de asignación), elija la estrategia de asignación que se utilizará al seleccionar los tipos de instancia de la lista de tipos de instancia permitidos. BEST_FIT_PROGRESSIVE suele ser la mejor opción para los entornos de computación bajo demanda de EC2, SPOT_CAPACITY_OPTIMIZED y SPOT_PRICE_CAPACITY_OPTIMIZED para los entornos de computación Spot de EC2. Para obtener más información, consulte the section called “Estrategias de asignación”.

l. (Opcional) Para la configuración de EC2, elija los valores de anulación de tipo de imagen e ID de imagen AWS Batch para proporcionar información y seleccionar Amazon Machine Images (AMI) para las instancias del entorno informático. Si no se especifica la anulación del ID de imagen para cada tipo de imagen, AWS Batch selecciona una AMI reciente optimizada para Amazon ECS. Si no se especifica ningún tipo de imagen, el valor predeterminado es una instancia de Amazon Linux 2 para instancias que no sean de GPU ni de AWS Graviton.

> **Important**
>
> Para usar una AMI personalizada, elija el tipo de imagen y, a continuación, introduzca el ID de AMI personalizado en el cuadro de Cambio de ID de imagen.

Amazon Linux 2

Es el valor predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo,, `C6g` `M6gR6g`, `yT4g`) y se puede usar para todos los tipos de instancias que no sean de GPU.

Amazon Linux 2 (GPU)

Es el valor predeterminado para todas las familias de instancias de GPU (por ejemplo, `P4` `yG4`) y se puede usar para todos los tipos de instancias que no estén basadas en AWS Graviton.

Amazon Linux

Se puede usar para familias de instancias que no utilizan GPU ni AWS Graviton. El soporte estándar para Amazon Linux ha finalizado. Para obtener más información, consulte AMI de Amazon Linux.

> **Note**
>
> La AMI que elija para un entorno de computación debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno de computación. Por ejemplo, si su entorno de computación utiliza tipos de instancias A1, la AMI de recursos de computación que elija debe admitir instancias Arm. Amazon ECS ofrece versiones x86 y Arm de la AMI Amazon Linux 2 optimizada

para Amazon ECS. Para obtener más información, consulte la sección sobre AMI Amazon Linux 2 optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

m. (Opcional) En Plantilla de lanzamiento, seleccione una plantilla de lanzamiento de Amazon EC2 existente para configurar sus recursos de computación. La versión predeterminada de la plantilla se rellena automáticamente. Para obtener más información, consulte Compatibilidad con las plantillas de lanzamiento.

> **Note**
>
> En una plantilla de lanzamiento, puede especificar una AMI personalizada que haya creado.

n. (Opcional) En Launch template version (Versión de la plantilla de lanzamiento), introduzca `$Default`, `$Latest` o el número de versión específico que desea utilizar.

> **Important**
>
> Si el parámetro de versión de la plantilla de lanzamiento es `$Default` o `$Latest`, la versión predeterminada o más reciente de la plantilla de lanzamiento especificada se evalúa durante una actualización de la infraestructura. Si se selecciona un ID de AMI diferente de forma predeterminada o se selecciona la última versión de la plantilla de lanzamiento, ese ID de AMI se utiliza en la actualización. Para obtener más información, consulte the section called “Actualización del ID de la AMI”.

o. Seleccione Página siguiente.

7. En la sección Configuración de red:

> **Important**
>
> Los recursos de computación de las deben obtener acceso para comunicarse con el punto de conexión del servicio de Amazon ECS. Esto puede ser a través de un punto de conexión de la VPC de la interfaz o a través de recursos de computación de las con direcciones IP públicas.
> Para obtener más información acerca de los puntos de enlace de la VPC de la interfaz, consulte Puntos de enlace de la VPC de la interfaz de Amazon ECS (AWS PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.

Si no tiene configurado un punto de conexión de la VPC de la interfaz y los recursos de computación de las no tienen direcciones IP públicas, deberán utilizar traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulte Gateways NAT en la Guía del usuario de Amazon VPC. Para obtener más información, consulte the section called “Creación de una VPC”.

a. Para el ID de la nube privada virtual (VPC), elija una VPC donde lanzar sus instancias.

b. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, se escogen todas las subredes dentro de la VPC disponible.

> **Note**
>
> AWS Batch en Amazon EC2 es compatible con Zonas Locales. Para obtener más información, consulte Zonas locales en la Guía del usuario de Amazon EC2 para instancias de Linux y clústeres de Amazon ECS en zonas locales, zonas de Wavelength y AWS Outposts en la Guía para desarrolladores de Amazon Elastic Container Service.

c. (Opcional) En Grupos de seguridad, seleccione su grupo de seguridad para asociarlo a las instancias. De forma predeterminada, se escoge el grupo de seguridad predeterminado para la VPC.

8. Seleccione Página siguiente.
9. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Para crear un entorno de computación no gestionado con los recursos de EC2

1. Abra la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En la página Entornos de computación, elija Crear.
4. Configure el entorno.

a. Para la Configuración del entorno de computación, elija Amazon Elastic Compute Cloud (Amazon EC2).

b. Para el Tipo de orquestación, seleccione No administrado.

5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
6. (Opcional) En el rol de servicio, elige un rol que permita al AWS Batch servicio realizar llamadas a las operaciones de AWS API requeridas en tu nombre. En este ejemplo, elija AWSBatchServiceRole. Para obtener más información, consulte the section called “Usar roles vinculados a servicios”.
7. En Máximo de CPU virtuales, seleccione la cantidad máxima de vCPUs admitida que su entorno de computación puede escalar horizontalmente, independientemente de la demanda de las colas de trabajos.
8. (Opcional) Amplíe las Etiquetas. Para agregar una etiqueta, elija Add tag (Añadir etiqueta). Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta. Para obtener más información, consulte Etiquetado de los recursos de AWS Batch.
9. Seleccione Página siguiente.
10. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Para crear un entorno informático gestionado con los recursos de Amazon EKS

1. Abra la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. Seleccione Crear.
5. Para la Configuración del entorno de computación, elija Amazon Elastic Kubernetes Service (Amazon EKS).
6. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

7. En Rol de instancia, elija un perfil de instancia existente que tenga asociados los permisos de IAM necesarios.

   > **Note**
   >
   > Para crear un entorno de procesamiento en la AWS Batch consola, elija un perfil de instancia que tenga los `eks:DescribeCluster` permisos `eks:ListClusters` y.

8. Para el clúster de EKS, elija un clúster de Amazon EKS existente.
9. En Espacio de nombres, introduzca un espacio de nombres Kubernetes para agrupar los procesos AWS Batch en el clúster.
10. (Opcional) Amplíe las Etiquetas. Elija Agregar etiqueta y, a continuación, introduzca un par clave-valor.
11. Seleccione Página siguiente.
12. (Opcional) En Utilizar las instancias de spot de EC2, active Habilitar el uso de instancias spot para utilizar las instancias de spot de Amazon EC2.
13. (Opcional) En Precio máximo % bajo demanda, seleccione el porcentaje máximo del precio que puede tener una instancia de spot en relación con su precio bajo demanda antes de que lancen las instancias. Por ejemplo, si el precio máximo es el 20%, el precio de spot de esa instancia de EC2 deberá ser inferior al 20% del precio bajo demanda que tenga en ese momento. Siempre se paga el precio más bajo (de mercado) y nunca más que lo marcado por el porcentaje máximo. Si se deja este campo en blanco, el valor predeterminado es el 100% del precio bajo demanda.
14. (Solo Spot) En Rol de la flota de spot, seleccione un rol de IAM para la flota de spot de Amazon EC2 que quiera aplicar a su entorno de computación `SPOT`.

    > **Important**
    >
    > Este rol es necesario si la estrategia de asignación está establecida en `BEST_FIT` o no se ha especificado.

15. (Opcional) En Mínimo de CPU virtuales, seleccione la cantidad mínima de vCPUs que mantiene el entorno de computación, independientemente de la demanda de las colas de trabajos.
16. (Opcional) En Máximo de CPU virtuales, seleccione la cantidad máxima de vCPUs que su entorno de computación puede escalar horizontalmente, independientemente de la demanda de las colas de trabajos.

17. En Tipos de instancias permitidos, elija los tipos de instancia de Amazon EC2 que se pueden lanzar. Se pueden especificar familias de instancias para lanzar cualquier tipo de instancia en esas familias (por ejemplo, `c5`, `c5n` o `p3`). O bien puede especificar tamaños específicos dentro de una familia (por ejemplo, `c5.8xlarge`). Los tipos de instancias metálicas no están en las familias de instancias. Por ejemplo, `c5` no incluye `c5.metal`. También puede seleccionar `optimal` para elegir tipos de instancias (de las familias de instancias C4, M4 y R4), ya que deben corresponderse con la demanda de las colas de trabajos.

> **Note**
>
> Cuando se crea un entorno de computación, los tipos de instancias que se seleccionen para dicho entorno de computación deben compartir la misma arquitectura. Por ejemplo, no se puede mezclar instancias x86 y ARM en el mismo entorno de computación.

> **Note**
>
> AWS Batch escala las GPU en función de la cantidad requerida en tus colas de trabajos. Para utilizar la programación de GPU, el entorno de computación debe incluir tipos de instancia de las familias `p2`, `p3`, `p4`, `p5`, `g3`, `g3s`, `g4` o `g5`.

> **Note**
>
> Actualmente, `optimal` utiliza tipos de instancia de las familias de instancias C4, M4 y R4. Si Regiones de AWS no hay tipos de instancias de esas familias de instancias, se utilizan los tipos de instancia de C5M5, y las familias de R5 instancias.

18. (Opcional) Expandir Configuración adicional.

    a. (Opcional) En Grupo de ubicación, introduzca un nombre de grupo de ubicación para agrupar los recursos en el entorno de computación.

    b. Para la Estrategia de asignación, elija BEST_FIT_PROGRESSIVE.

    c. (Opcional) Para la configuración de Imágenes de máquinas de Amazon (AMI), seleccione la configuración Agregar imágenes de máquinas de Amazon (amis). A continuación, elija un Tipo de imagen, introduzca un Cambio del ID de imagen y la versión Kubernetes.

> **Important**
>
> Para usar una AMI personalizada, elija el tipo de imagen y, a continuación, introduzca el ID de AMI personalizado en el cuadro de Cambio de ID de imagen.

> **Note**
>
> Si no se especifica la anulación del ID de imagen para cada tipo de imagen, AWS Batch selecciona una AMI reciente optimizada para Amazon ECS. Si no se especifica ningún tipo de imagen, el valor predeterminado es una instancia de Amazon Linux 2 para instancias que no sean de GPU ni de AWS Graviton.
>
> Amazon Linux 2
>
> Es el valor predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo,, `C6g` `M6gR6g`, `yT4g`) y se puede usar para todos los tipos de instancias que no sean de GPU.
>
> Amazon Linux 2 (GPU)
>
> Es el valor predeterminado para todas las familias de instancias de GPU (por ejemplo, `P4` `yG4`) y se puede usar para todos los tipos de instancias que no estén basados en AWS Graviton.

d. (Opcional) En Plantilla de lanzamiento, elija una plantilla de lanzamiento existente.

e. (Opcional) En Versión de la plantilla de lanzamiento, introduzca **`$Default`**, **`$Latest`** o un número de versión.

19. Seleccione Página siguiente.
20. Para el ID de la nube privada virtual (VPC), elija una VPC donde lanzar las instancias.
21. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, se escogen todas las subredes dentro de la VPC disponible.

> **Note**
>
> AWS Batch en Amazon, EKS admite Zonas Locales. Para obtener más información, consulte Amazon EKS y Zonas AWS Locales en la Guía del usuario de Amazon EKS.

22. (Opcional) En Grupos de seguridad, seleccione su grupo de seguridad para asociarlo a las instancias. De forma predeterminada, se elige el grupo de seguridad predeterminado para la VPC.
23. Seleccione Página siguiente.
24. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Plantillas de entorno informático

En el siguiente ejemplo, se muestra una plantilla de entorno informático vacía. Puede utilizar esta plantilla para crear un entorno informático que puede guardar en un archivo y utilizarse con la opción `--cli-input-json` de la AWS CLI. Para obtener más información sobre estos parámetros, consulte CreateComputeEnvironment en la AWS Batch Referencia de API.

```
{
    "computeEnvironmentName": "",
    "type": "UNMANAGED",
    "state": "DISABLED",
    "unmanagedvCpus": 0,
    "computeResources": {
        "type": "EC2",
        "allocationStrategy": "BEST_FIT_PROGRESSIVE",
        "minvCpus": 0,
        "maxvCpus": 0,
        "desiredvCpus": 0,
        "instanceTypes": [
            ""
        ],
        "imageId": "",
        "subnets": [
            ""
        ],
        "securityGroupIds": [
            ""
        ],
        "ec2KeyPair": "",
        "instanceRole": "",
        "tags": {
            "KeyName": ""
        },
```

```
        "placementGroup": "",
        "bidPercentage": 0,
        "spotIamFleetRole": "",
        "launchTemplate": {
            "launchTemplateId": "",
            "launchTemplateName": "",
            "version": ""
        },
        "ec2Configuration": [
            {
                "imageType": "",
                "imageIdOverride": "",
                "imageKubernetesVersion": ""
            }
        ]
    },
    "serviceRole": "",
    "tags": {
        "KeyName": ""
    },
    "eksConfiguration": {
        "eksClusterArn": "",
        "kubernetesNamespace": ""
    }
}
```

Note

Puede generar la plantilla de entorno de computación mostrada anteriormente con el siguiente comando de la AWS CLI.

```
$ aws batch create-compute-environment --generate-cli-skeleton
```

Parámetros de un entorno informático

Los entornos informáticos se dividen en cinco componentes básicos: nombre, tipo, estado, definición del recurso informático (en el caso de entornos informáticos administrados), configuración de Amazon EKS (en el caso de recursos de Amazon EKS) y rol de servicio a utilizar para concederle permisos de IAM a AWS Batch, y las etiquetas para el entorno informático.

Temas

Nombre del entorno informático

`computeEnvironmentName`

Nombre del entorno informático. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

Tipo: cadena

Obligatorio: sí

Tipo

`type`

El tipo de entorno informático. Seleccione `MANAGED` para que AWS Batch administre los recursos informáticos de EC2 o Fargate que defina. Para obtener más información, consulte Recursos informáticos. Seleccione `UNMANAGED` para administrar sus propios recursos informáticos de EC2.

Tipo: cadena

Valores válidos: `MANAGED` | `UNMANAGED`

Obligatorio: sí

Estado

`state`

El estado del entorno informático.

Si el estado es `ENABLED`, el programador de AWS Batch intenta colocar los trabajos en el entorno. Estos trabajos provienen de una cola de trabajos asociada en los recursos informáticos. Si se administra el entorno informático, las instancias se escalar horizontalmente o verticalmente automáticamente según la demanda de la cola de trabajos.

Si el estado es `DISABLED`, el programador de AWS Batch no intentará colocar los trabajos en el entorno. Los trabajos que están en estado `STARTING` o `RUNNING` continúan avanzando con normalidad. Los entornos informáticos administrados que están en el estado `DISABLED` no se escalan horizontalmente.

> **Note**
>
> Es posible que los entornos informáticos de un estado `DISABLED` sigan incurriendo en cargos de facturación. Para evitar cargos adicionales, desactive y, a continuación, elimine el entorno informático. Para obtener más información, consulte DeleteComputeEnvironment en la Referencia de la API AWS Batch y Evitar cargos inesperados en la Guía del usuario de AWS Billing.

Cuando una instancia está inactiva, la instancia se reduce al valor `minvCpus`. Sin embargo, el tamaño de instancia no cambia. Por ejemplo, considere una instancia `c5.8xlarge` con un valor `minvCpus` de `4` y un valor `desiredvCpus` de `36`. Esta instancia no se reduce a una instancia `c5.large`.

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

Recursos informáticos

`computeResources`

Detalles de los recursos informáticos administrados por el entorno informático. Para obtener más información, consulte Entorno de computación.

Tipo: objeto ComputeResource

Necesario: este parámetro es necesario para entornos informáticos administrados

`type`

El tipo de entorno informático. Puede optar por utilizar las instancias bajo demanda de EC2 (`EC2`) y las instancias de spot de EC2 (`SPOT`), o bien utilizar la capacidad de Fargate (`FARGATE`) y Fargate Spot (`FARGATE_SPOT`) en su entorno informático administrado. Si elige `SPOT`, también deberá especificar un rol de la flota de spot de Amazon EC2; con el parámetro `spotIamFleetRole`. Para obtener más información, consulte Rol de flota de spot de Amazon EC2.

Valores válidos: `EC2` | `SPOT` | `FARGATE` | `FARGATE_SPOT`

Obligatorio: sí

`allocationStrategy`

La estrategia de asignación que se utilizará para el recurso informático si no se puedan asignar suficientes instancias del tipo de instancia EC2 más adecuado. Esto puede deberse a la disponibilidad del tipo de instancia en la Región de AWS o a los límites de los servicios de Amazon EC2. Para obtener más información, consulte Estrategias de asignación.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

`BEST_FIT` (predeterminado)

AWS Batch selecciona el tipo de instancia que mejor se adapte a las necesidades de los trabajos con preferencia por el tipo de instancia de menor costo. Si no hay instancias adicionales disponibles del tipo de instancia seleccionada, AWS Batch espera a que estén disponibles. Si no hay suficientes instancias disponibles, o si está alcanzando

los límites del servicio de Amazon EC2, los trabajos adicionales no se ejecutan hasta que se completen los trabajos que se están ejecutando actualmente. Esta estrategia de asignación mantiene los costos más bajos, pero puede limitar el escalado. Si está utilizando flotas spot con `BEST_FIT` se debe especificar el rol de IAM de flota de spot. Los recursos informáticos que utilizan una estrategia de asignación `BEST_FIT` no admiten las actualizaciones de infraestructura y no pueden actualizar algunos parámetros. Para obtener más información, consulte Actualizar entornos informáticos.

> **Note**
>
> `BEST_FIT` no es compatible con entornos informáticos que utilicen recursos de Amazon EKS.

`BEST_FIT_PROGRESSIVE`

Utilice tipos de instancia adicionales que sean lo suficientemente grandes como para cumplir los requisitos de los trabajos en la cola. Prefiera tipos de instancias con un coste menor para cada unidad de vCPU. Si las instancias adicionales de los tipos de instancia previamente seleccionados no están disponibles, AWS Batch selecciona nuevos tipos de instancia.

`SPOT_CAPACITY_OPTIMIZED`

(Solo disponible para recursos informáticos de instancia de spot) Utilice tipos de instancia adicionales que sean lo suficientemente grandes como para cumplir los requisitos de los trabajos en la cola. Prefiera los tipos de instancias que tienen menos probabilidades de interrumpirse.

`SPOT_PRICE_CAPACITY_OPTIMIZED`

(Sólo disponible para recursos informáticos de instancia de spot) La estrategia de asignación optimizada de precio y capacidad analiza tanto el precio como la capacidad para seleccionar los grupos de instancias de spot que tienen menos probabilidades de interrupción y tienen el precio más bajo posible.

> **Note**
>
> Le recomendamos que utilice `SPOT_PRICE_CAPACITY_OPTIMIZED` en vez de `SPOT_CAPACITY_OPTIMIZED` la mayoría de los casos.

Con estrategias de asignación `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` y `SPOT_PRICE_CAPACITY_OPTIMIZED` que utilizan instancias de spot o bajo demanda, y la estrategia `BEST_FIT` que usa instancias de spot, es posible que AWS Batch deba superar `maxvCpus` para cumplir los requisitos de capacidad. En este caso, AWS Batch nunca supera `maxvCpus` en más de una sola instancia.

Valores válidos: `BEST_FIT` | `BEST_FIT_PROGRESSIVE` | `SPOT_CAPACITY_OPTIMIZED` | `SPOT_PRICE_CAPACITY_OPTIMIZED`

Obligatorio: no

`minvCpus`

La cantidad mínima de vCPU virtuales que debe mantener un entorno (incluso si un entorno informático es `DISABLED`.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`maxvCpus`

La cantidad máxima de vCPU que el entorno informático AWS Batch puede admitir.

> **Note**
>
> Con estrategias de asignación `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` y `SPOT_PRICE_CAPACITY_OPTIMIZED` que utilizan instancias de spot o bajo demanda, y la estrategia `BEST_FIT` que usa instancias de spot, es posible que AWS Batch deba superar `maxvCpus` para cumplir los requisitos de capacidad. En este caso, AWS Batch nunca supera `maxvCpus` en más de una sola instancia. Por ejemplo, AWS Batch usa no más de una sola instancia entre las especificadas en su entorno informático.

Tipo: entero

Obligatorio: no

`desiredvCpus`

La cantidad deseada de vCPU virtuales en el entorno informático. AWS Batch modifica este valor entre los valores mínimo y máximo, en función de la demanda de la cola de trabajos.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`instanceTypes`

Los tipos de instancia que pueden lanzarse. Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No lo especifique. Se pueden especificar familias de instancias para lanzar cualquier tipo de instancia en esas familias (por ejemplo, `c5`, `c5n` o `p3`). O bien puede especificar tamaños específicos dentro de una familia (por ejemplo, `c5.8xlarge`). Tenga en cuenta que los tipos de instancias metal no están en las familias de instancias (por ejemplo, `c5` no incluye `c5.metal`). También se puede seleccionar `optimal` para elegir tipos de instancias (de las familias de instancias C4, M4 y R4) que se correspondan con la demanda de las colas de trabajos.

> **Note**
>
> Cuando se crea un entorno informático, los tipos de instancias que se seleccionen para dicho entorno informático deben compartir la misma arquitectura. Por ejemplo, no se puede mezclar instancias x86 y ARM en el mismo entorno informático.

> **Note**
>
> Actualmente, `optimal` utiliza tipos de instancia de las familias de instancias C4, M4 y R4. En Regiones de AWS que no tienen los tipos de instancia de las familias mencionadas, se utilizan tipos de instancia de las familias de instancias C5, M5 y R5.

Tipo: matriz de cadenas

Obligatorio: sí

`imageId`

Este parámetro se ha quedado obsoleto.

El ID de la Imagen de máquina de Amazon (AMI) se utiliza para instancias lanzadas en el entorno informático. Este parámetro es invalidado por el miembro `imageIdOverride` de la estructura `Ec2Configuration`.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

> **Note**
>
> La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno informático. Por ejemplo, si su entorno informático utiliza tipos de instancias A1, la AMI de recursos informáticos que elija debe admitir instancias Arm. Amazon ECS ofrece versiones x86 y Arm de la AMI Amazon Linux 2 optimizada para Amazon ECS. Para obtener más información, consulte la sección sobre AMI Amazon Linux 2 optimizada para Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

Tipo: cadena

Obligatorio: no

`subnets`

Las subredes de VPC en las que se lanzan los recursos informáticos. Estas subredes deben estar dentro de la misma VPC. Los recursos informáticos de Fargate pueden contener un máximo de 16 subredes. Para obtener más información, consulte VPC y subredes en la Guía del usuario de Amazon VPC.

> **Note**
>
> AWS Batch en Amazon EC2 y AWS Batch en Amazon EKS admiten zonas locales. Para obtener más información, consulte Zonas locales en la Guía del usuario de Amazon EC2 para instancias de Linux, Amazon EKS y zonas locales de AWS en la Guía del usuario de AmazonEKS y Clústeres de Amazon ECS en zonas locales, zonas de Wavelength y AWS Outposts en la Guía para desarrolladores de Amazon Elastic Container Service.
> AWS Batch en Fargate no admite zonas locales.

Al actualizar los entornos de procesamiento, si proporciona una lista vacía de subredes de VPC, el comportamiento resultante difiere entre los recursos de procesamiento de Fargate y EC2. En el caso de los recursos informáticos de Fargate, el hecho de proporcionar una lista vacía se gestiona como si no se hubiera especificado este parámetro y no se realiza ningún cambio. Para los recursos informáticos de EC2, al proporcionar una lista vacía se eliminan las subredes de la VPC del recurso informático. Si cambia las subredes de la VPC, se requiere una actualización de la infraestructura del entorno informático. Este es el caso de los recursos informáticos de Fargate y EC2. Para obtener más información, consulte Actualizar entornos informáticos.

Tipo: matriz de cadenas

Obligatorio: sí

`securityGroupIds`

Los grupos de seguridad de Amazon EC2 asociados con las instancias lanzadas en el entorno informático. Se deben especificar uno o varios grupos de seguridad, ya sea en `securityGroupIds` o mediante una plantilla de lanzamiento a la que se hace referencia en `launchTemplate`. Este parámetro es necesario para trabajos que se ejecutan en recursos de Fargate y debe contener al menos un grupo de seguridad. (Fargate no admite plantillas de lanzamiento.) Si los grupos de seguridad se especifican mediante `securityGroupIds` y `launchTemplate`, se utilizarán los valores de `securityGroupIds`.

Al actualizar los entornos informáticos, si proporciona una lista vacía de grupos de seguridad, el comportamiento resultante difiere entre los recursos informáticos de Fargate y EC2. En el caso de los recursos informáticos de Fargate, el hecho de proporcionar una lista vacía se gestiona como si no se hubiera especificado este parámetro y no se realiza ningún cambio.

En el caso de los recursos informáticos de EC2, al proporcionar una lista vacía se eliminan los grupos de seguridad del recurso informático. Si cambia los grupos de seguridad, se requiere una actualización de la infraestructura del entorno informático. Este es el caso de los recursos informáticos de Fargate y EC2. Para obtener más información, consulte Actualizar entornos informáticos.

Tipo: matriz de cadenas

Obligatorio: sí

`ec2KeyPair`

El par de claves de EC2 que se utiliza para las instancias lanzadas en el entorno informático. Puede utilizar este par de claves para iniciar sesión en las instancias con SSH. Al actualizar un entorno informático, si cambia el par de claves EC2, se requiere una actualización de la infraestructura del entorno informático. Para obtener más información, consulte Actualizar entornos informáticos.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: cadena

Obligatorio: no

`instanceRole`

El perfil de instancia ECS de Amazon que se adjunta a instancias Amazon EC2 en un entorno informático. Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No lo especifique. Puede especificar el nombre abreviado o el Nombre de recurso de Amazon (ARN) completo de un perfil de instancia. Por ejemplo, `ecsInstanceRole` o `arn:aws:iam::`*`aws_account_id`*`:instance-profile/ecsInstanceRole`. Para obtener más información, consulte Función de instancia de Amazon ECS.

Al actualizar un entorno informático, si cambia esta configuración, se requiere una actualización de la infraestructura del entorno informático. Para obtener más información, consulte Actualizar entornos informáticos.

Tipo: string

Obligatorio: no

`tags`

Las etiquetas de los pares de valores de claves que se aplican a las instancias EC2 que se lanzan en el entorno informático. Por ejemplo, puede especificar `"Name": "AWS Batch Instance - C4OnDemand"` como etiqueta, de modo que cada instancia de su entorno informático tenga ese nombre. Esto es útil para reconocer las instancias de AWS Batch en la consola de Amazon EC2. Estas etiquetas no se ven cuando se utiliza la operación de la API AWS Batch de ListTagsForResource.

Al actualizar un entorno informático, el cambio de las etiquetas EC2 requiere una actualización de la infraestructura del entorno informático. Para obtener más información, consulte Actualizar entornos informáticos.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: mapa de cadena a cadena

Obligatorio: no

`placementGroup`

El grupo de ubicación de Amazon EC2 que se va a asociar a sus recursos informáticos. Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No lo especifique. Si va a enviar trabajos paralelos de varios nodos a su entorno informático, considere la posibilidad de crear un grupo con ubicación en clúster y asociarlo a sus recursos informáticos. Esto mantiene el trabajo paralelo de varios nodos en una agrupación lógica de instancias con una sola zona de disponibilidad con alto potencial de flujo de la red. Para obtener más información, consulte Grupos de ubicación en la Guía del usuario de instancias de Linux de Amazon EC2.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: cadena

Obligatorio: no

`bidPercentage`

El porcentaje máximo que puede tener el precio una instancia de spot de EC2 en comparación con el precio bajo demanda para ese tipo de instancia antes de que se lancen las instancias. Por ejemplo, si el porcentaje máximo es del 20 %, el precio de spot de la instancia de EC2 debe ser inferior al 20 % del precio bajo demanda actual. Siempre se paga el precio más bajo (de mercado) y nunca más que lo marcado por el porcentaje máximo. Si se deja este campo en blanco, el valor predeterminado es el 100% del precio bajo demanda. En la mayoría de los casos de uso, recomendamos dejar este campo vacío.

Al actualizar un entorno informático, si cambia el porcentaje de oferta, se requiere una actualización de la infraestructura del entorno informático. Para obtener más información, consulte Actualizar entornos informáticos.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Obligatorio: no

`spotIamFleetRole`

El Nombre de recurso de Amazon (ARN) del rol de IAM para la flota de spot de Amazon EC2 que se aplica a un entorno informático `SPOT`. Este rol es necesario si la estrategia de asignación está establecida en `BEST_FIT` o si no se ha especificado la estrategia de asignación. Para obtener más información, consulte Rol de flota de spot de Amazon EC2.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

> **Important**
>
> Para etiquetar las instancias puntuales al crearlas, la función de IAM de la flota puntual especificada aquí debe utilizar la política gestionada más reciente de AmazonEC2 SpotFleetTaggingRole. La política SpotFleetRole administrada de AmazonEC2

recomendada anteriormente no tiene los permisos necesarios para etiquetar las instancias puntuales. Para obtener más información, consulte Instancias de spot no etiquetadas en el momento de su creación.

Tipo: cadena

Necesario: este parámetro es necesario para entornos informáticos `SPOT`.

`launchTemplate`

Una plantilla de lanzamiento opcional para asociarla a los recursos informáticos. Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No lo especifique. Cualquier otro parámetro del recurso informático que se especifica en una operación CreateComputeEnvironment o UpdateComputeEnvironment de la API invalida el mismo parámetro en la plantilla de lanzamiento. Para utilizar una plantilla de lanzamiento, debe especificar el ID o el nombre de esta en la solicitud, pero no ambos. Para obtener más información, consulte Compatibilidad con las plantillas de lanzamiento.

Al actualizar un entorno informático, para eliminar la plantilla de lanzamiento personalizada y utilizar la plantilla de lanzamiento predeterminada, defina el miembro `launchTemplateId` o `launchTemplateName` de la especificación de la plantilla de lanzamiento en una cadena vacía. La eliminación de la plantilla de lanzamiento de un entorno informático no eliminará la AMI especificada en la plantilla de lanzamiento, si era la que se usaba. Para actualizar la AMI seleccionada de una plantilla de lanzamiento, el parámetro `updateToLatestImageVersion` debe establecerse en `true`. Al actualizar un entorno informático, si cambia la plantilla de inicio, se requiere una actualización de la infraestructura del entorno informático. Para obtener más información, consulte Actualizar entornos informáticos.

Tipo: LaunchTemplateSpecification

objeto

Obligatorio: no

`launchTemplateId`

El ID de la plantilla de lanzamiento.

Tipo: cadena

Obligatorio: no

`launchTemplateName`

El nombre de la plantilla de lanzamiento.

Tipo: cadena

Obligatorio: no

`version`

El número de versión de la plantilla de lanzamiento, `$Latest` o `$Default`.

Si el valor es `$Latest`, se utiliza la versión más reciente de la plantilla de lanzamiento. Si el valor es `$Default`, se utiliza la versión predeterminada de la plantilla de lanzamiento. Durante una actualización de infraestructura, si se especificó `$Latest` o `$Default` para el entorno informático, AWS Batch reevalúa la versión de la plantilla de lanzamiento y puede utilizar una versión diferente de la plantilla de lanzamiento. Esto es incluso si la plantilla de lanzamiento no se especificó en la actualización.

Valor predeterminado: `$Default`.

Tipo: cadena

Obligatorio: no

`ec2Configuration`

Proporciona información utilizada para seleccionar Amazon Machine Images (AMI) para instancias de EC2 en el entorno informático. Si `Ec2Configuration` no se especifica, el valor predeterminado es Amazon Linux 2 (`ECS_AL2`). Antes del 31 de marzo de 2021, el valor predeterminado era Amazon Linux (`ECS_AL1`) para instancias que no fueran GPU ni Graviton AWS.

Al actualizar un entorno informático, el cambio de este parámetro requiere una actualización de la infraestructura del entorno informático. Para obtener más información, consulte Actualizar entornos informáticos.

> **Note**
>
> Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: matriz de objetos Ec2Configuration

Obligatorio: no

`imageIdOverride`

El ID de AMI que se utiliza para instancias lanzadas en el entorno informático que coinciden con el tipo de imagen. Esta configuración invalida el `imageId` establecido en el objeto `computeResource`.

Tipo: cadena

Obligatorio: no

`imageKubernetesVersion`

La versión Kubernetes para el entorno informático. Si no se especifica un valor, se utiliza la versión más reciente que sea compatible con AWS Batch.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

`imageType`

El tipo de imagen que debe coincidir con el tipo de instancia para seleccionar una AMI. Los valores admitidos son diferentes para los recursos de `ECS` y `EKS`.

ECS

Si no se especifica el parámetro `imageIdOverride`, se utiliza una AMI de Amazon Linux 2 optimizada para ECS (`ECS_AL2`). Si se especifica un nuevo tipo de imagen en una actualización, pero no se especifica `imageId` ni un parámetro `imageIdOverride`, se utilizará la última AMI optimizada de Amazon ECS para ese tipo de imagen que sea compatible con AWS Batch.

ECS_AL2

Amazon Linux 2: predeterminado para todas las familias de instancias que no son GPU.

ECS_AL2_NVIDIA

Amazon Linux 2 (GPU): predeterminado para todas las familias de instancias de la GPU (por ejemplo, `P4` y `G4`), y se puede utilizar para todos los tipos de instancia no basados en AWS Graviton.

ECS_AL1

Amazon Linux. Amazon Linux ha alcanzado end-of-life el soporte estándar. Para obtener más información, consulte AMI de Amazon Linux.

EKS

Si no se especifica el parámetro `imageIdOverride`, se utiliza una AMI de Amazon Linux reciente optimizada para EKS (`EKS_AL2`). Si se especifica un tipo de imagen nuevo en una actualización, pero no se especifica `imageId` ni un parámetro `imageIdOverride`, se utiliza la última AMI optimizada de Amazon EKS para ese tipo de imagen que sea compatible con AWS Batch.

EKS_AL2

Amazon Linux 2: predeterminado para todas las familias de instancias que no son GPU.

EKS_AL2_NVIDIA

Amazon Linux 2 (acelerado): predeterminado para todas las familias de instancias de GPU (por ejemplo, `P4` y `G4`) y se puede utilizar para todos los tipos de instancias no basados en AWS Graviton.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

Configuración de Amazon EKS

Configuración del clúster de Amazon EKS que admite el entorno informático de AWS Batch. El clúster debe existir antes de que se cree el entorno informático.

`eksClusterArn`

El nombre de recurso de Amazon (ARN) del clúster de Amazon EKS. Un ejemplo es `arn:aws:eks:`*`us-east-1`*`:`*`123456789012`*`:cluster/`*`ClusterForBatch`*.

Tipo: cadena

Obligatorio: sí

`kubernetesNamespace`

El espacio de nombres del clúster de Amazon EKS. AWS Batch administra los pods en este espacio de nombres. El valor no puede estar vacío ni ser nulo. Debe tener menos de 64 caracteres, no se puede establecer en `default`, no puede empezar con "`kube-`" y debe coincidir con la siguiente expresión regular: `^[a-z0-9]([-a-z0-9]*[a-z0-9])?$`. Para obtener más información, consulte Espacios de nombres en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: sí

Tipo: EksConfigurationobjeto

Obligatorio: no

Rol de servicio

`serviceRole`

El nombre de recurso de Amazon (ARN) completo del rol de IAM que permite a AWS Batch llamar a otros servicios de AWS en su nombre. Para obtener más información, consulte Uso de funciones vinculadas a servicios para AWS Batch. Se recomienda no especificar el rol de servicio. De este modo, AWS Batch usa el rol vinculado al servicio AWSServiceRoleForBatch.

> **Important**
>
> Si la cuenta ya creó el rol vinculado al servicio AWS Batch (AWSServiceRoleForBatch), se usa dicho rol de forma predeterminada para el servicio, a menos que se especifique un rol aquí. Si el rol vinculado al servicio de AWS Batch no existe en la cuenta y no se especifica ningún rol aquí, el servicio intentará crear el rol vinculado al servicio de AWS Batch en la cuenta. Para obtener más información sobre el rol vinculado al servicio de AWSServiceRoleForBatch, consulte Permisos de rol vinculados al servicio para AWS Batch.
>
> Si el entorno de procesamiento se crea con el rol vinculado al servicio AWSServiceRoleForBatch, no se puede cambiar para usar un rol de IAM normal. Del mismo modo, si el entorno de cómputo se crea con un rol de IAM normal, no se puede

cambiar para usar el rol vinculado al servicio AWSServiceRoleForBatch. Para actualizar los parámetros del entorno de cómputo que requieren un cambio en la infraestructura, se debe usar el rol vinculado al servicio AWSServiceRoleForBatch. Para obtener más información, consulte Actualizar entornos informáticos.

Si el rol especificado tiene una ruta distinta de `/`, entonces debe especificar el ARN de rol completo (recomendado) o prefijar el nombre del rol con la ruta.

> **Note**
>
> En función de cómo se haya creado el rol de servicio de AWS Batch, el Nombre de recurso de Amazon (ARN) puede contener el prefijo de ruta `service-role`. Cuando se especifica únicamente el nombre del rol de servicio, AWS Batch supone que el ARN no usa el prefijo de ruta `service-role`. Por eso se recomienda especificar el ARN completo para el rol de servicio al crear entornos informáticos.

Tipo: cadena

Obligatorio: no

Etiquetas

`tags`

Etiquetas de pares clave-valor para asociar con el entorno informático. Para obtener más información, consulte Etiquetado de los recursos de AWS Batch.

Tipo: mapa de cadena a cadena

Obligatorio: no

Configuración de EC2

AWS Batch utiliza AMI optimizadas para Amazon ECS para entornos de cómputo puntual de EC2 y EC2. El predeterminado es Amazon Linux 2 (`ECS_AL2`). Antes del 31 de marzo de 2021, este valor

predeterminado era Amazon Linux (`ECS_AL1`) para las instancias que no eran de GPU ni de AWS Graviton.

> **Note**
>
> AWS Batch también es compatible con Amazon Linux 2023.

La AMI de Amazon Linux (también llamada Amazon Linux 1) llegó al final de su vida útil el 31 de diciembre de 2023. AWS Batch ha dejado de dar soporte a la AMI de Amazon Linux, ya que no recibirá actualizaciones de seguridad ni correcciones de errores a partir del 1 de enero de 2024. Para obtener más información sobre Amazon Linux end-of-life, consulte las preguntas frecuentes de AL.

Le recomendamos que actualice los entornos informáticos actuales basados en Amazon Linux a Amazon Linux 2023 para evitar interrupciones imprevistas en la carga de trabajo y que siga recibiendo actualizaciones de seguridad y de otro tipo.

Es posible que sus entornos informáticos que utilizan la AMI de Amazon Linux sigan funcionando después de la end-of-life fecha del 31 de diciembre de 2023. Sin embargo, estos entornos informáticos ya no recibirán nuevas actualizaciones de software, parches de seguridad ni correcciones de errores AWS. Es su responsabilidad mantener estos entornos de cómputo en la AMI de Amazon Linux después end-of-life. Recomendamos migrar los entornos AWS Batch informáticos a Amazon Linux 2023 o Amazon Linux 2 para mantener un rendimiento y una seguridad óptimos.

Para obtener ayuda para migrar AWS Batch de la AMI de Amazon Linux a Amazon Linux 2023 o Amazon Linux 2, consulte Actualización de entornos informáticos - AWS Batch

Estrategias de asignación

Cuando se crea un entorno informático gestionado, AWS Batch selecciona los tipos de instancia `instanceTypes` especificados que mejor se adapten a las necesidades de los trabajos. La estrategia de asignación define el comportamiento cuando se AWS Batch necesita capacidad adicional. Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No las especifique en ambos lugares.

`BEST_FIT` (predeterminado)

AWS Batch selecciona el tipo de instancia que mejor se adapte a las necesidades de los trabajos y prefiere el tipo de instancia de menor coste. Si no hay disponibles instancias adicionales del tipo

de instancia seleccionado, AWS Batch espera a que estén disponibles las instancias adicionales. Si no hay suficientes instancias disponibles, o si el usuario alcanza los Service Quotas de Amazon EC2, no se ejecutarán trabajos adicionales hasta que finalicen aquellos que se estén ejecutando en ese momento. Esta estrategia de asignación mantiene los costos más bajos, pero puede limitar el escalado. Si está utilizando la flota de spot con `BEST_FIT`, el rol de IAM de la flota spot debe especificarse. `BEST_FIT` no se admite cuando se actualizan los entornos de computación. Para obtener más información, consulte Actualizar entornos informáticos.

> **Note**
>
> AWS Batch administra AWS los recursos de su cuenta. Los entornos informáticos con la estrategia de asignación BEST_FIT utilizaban originalmente configuraciones de lanzamiento de forma predeterminada. Sin embargo, el uso de configuraciones de lanzamiento con AWS cuentas nuevas se restringirá con el tiempo. Por lo tanto, a partir de finales de abril de 2024, los entornos de cómputo BEST_FIT recién creados lanzarán plantillas de forma predeterminada. Si su función de servicio no tiene permisos para administrar las plantillas de lanzamiento, AWS Batch puede seguir utilizando las configuraciones de lanzamiento. Los entornos informáticos existentes seguirán utilizando las configuraciones de lanzamiento.

`BEST_FIT_PROGRESSIVE`

AWS Batch selecciona tipos de instancias adicionales que sean lo suficientemente grandes como para cumplir con los requisitos de los trabajos de la cola. Se prefieren los tipos de instancias con un costo menor para cada unidad de vCPU. Si las instancias adicionales de los tipos de instancia previamente seleccionados no están disponibles, AWS Batch selecciona nuevos tipos de instancia.

`SPOT_CAPACITY_OPTIMIZED`

AWS Batch selecciona uno o más tipos de instancias que son lo suficientemente grandes como para cumplir con los requisitos de los trabajos de la cola. Se prefieren los tipos de instancias que tienen menos probabilidades de interrumpirse. Esta estrategia de asignación solo está disponible para los recursos de computación de instancia de spot.

`SPOT_PRICE_CAPACITY_OPTIMIZED`

La estrategia de asignación optimizada por precio y capacidad analiza tanto el precio como la capacidad para seleccionar los grupos de instancias de spot que tienen menos probabilidades de

interrupción y el precio más bajo posible. Esta estrategia de asignación solo está disponible para los recursos de computación de instancia de spot.

> **Note**
>
> Le recomendamos que utilice `SPOT_PRICE_CAPACITY_OPTIMIZED` en vez de `SPOT_CAPACITY_OPTIMIZED` la mayoría de los casos.

Las estrategias `BEST_FIT_PROGRESSIVE` y `BEST_FIT` utilizan instancias de spot bajo demanda, y las estrategias `SPOT_CAPACITY_OPTIMIZED` y `SPOT_PRICE_CAPACITY_OPTIMIZED` utilizan instancias de spot. Sin embargo, AWS Batch es posible que deba superarlos `maxvCpus` para cumplir con sus requisitos de capacidad. En este caso, AWS Batch nunca supere `maxvCpus` en más de una instancia.

Actualizar entornos informáticos

Tras crear un entorno informático que utilice recursos de EC2, podrá actualizar directamente muchos de los ajustes del entorno informático. Sin embargo, para cambiar algunos de los ajustes es necesario que AWS Batch reemplace las instancias del entorno informático.

Para los entornos informáticos que utilizan recursos de Fargate, puede actualizar lo siguiente.

- `securityGroupIds`
- `subnets`
- `desiredvCpus`
- `maxvCpus`
- `minvCpus`

AWS Batch tiene dos mecanismos de actualización. La primera es una actualización de escalado en la que se añaden o se borran instancias del entorno informático. La segunda es una actualización de infraestructura en la que se sustituyen las instancias del entorno informático. Una actualización de infraestructura lleva mucho más tiempo que una actualización de escalado.

Si actualiza los entornos de procesamiento con AWS Batch, al cambiar solo estas configuraciones, se produce una actualización de escalado: las vCPU deseadas (`desiredvCpus`), las vCPU

máximas (`maxvCpus`), las vCPU mínimas (`minvCpus`), el rol de servicio (`serviceRole`) y el estado (`state`).

> **Note**
>
> Al actualizar la configuración de `desiredvCpus`, el valor debe estar entre los valores `minvCpus` y `maxvCpus`.
> Además, el valor actualizado de `desiredvCpus` debe ser mayor o igual que el valor actual de `desiredvCpus`. Para obtener más información, consulte the section called “Mensaje de error al actualizar la configuración de `desiredvCpus`”.

Si alguna de las siguientes configuraciones se modifica en una acción de la API UpdateComputeEnvironment, AWS Batch inicia una actualización de la infraestructura. Una actualización de la infraestructura requiere que el rol de servicio esté establecido en AWSServiceRoleForBatch (el valor predeterminado) y que la estrategia de asignación sea `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, o `SPOT_PRICE_CAPACITY_OPTIMIZED`. `BEST_FIT` no se admite. A excepción del rol de servicio, todos los ajustes que se pueden cambiar para una actualización de escalado también se pueden cambiar para una actualización de infraestructura.

> **Note**
>
> Se recomienda utilizar `SPOT_PRICE_CAPACITY_OPTIMIZED` en lugar de `SPOT_CAPACITY_OPTIMIZED` en la mayoría de los casos.

Durante una actualización de infraestructura, el estado del entorno informático cambia a `UPDATING`. Las nuevas instancias se lanzan con la configuración actualizada. Los nuevos trabajos están programados en las nuevas instancias. Los trabajos que se están ejecutando actualmente se envían de acuerdo con la política de actualización de la infraestructura. Para obtener más información, consulte UpdateComputeEnvironment y UpdatePolicy en la Referencia de la API de AWS Batch.

En el tipo de datos `UpdatePolicy`, considere las siguientes situaciones:

> **Note**
>
> En estas situaciones, se aplica lo siguiente. Cuando se termina una instancia, se detienen los trabajos en ejecución. De forma predeterminada, estos trabajos no se vuelven a intentar.

Para volver a intentar uno de estos trabajos una vez finalizada una instancia, configura una estrategia de reintento de trabajo. Para obtener más información, consulte the section called "Reintentos automáticos de trabajo" en la Guía del usuario de AWS Batch.

- Si la configuración `terminateJobsOnUpdate` está establecida en `true`, los trabajos en ejecución finalizarán durante una actualización de la infraestructura. La configuración `jobExecutionTimeoutMinutes` se omite.
- Si la configuración `terminateJobsOnUpdate` está establecida en `false`, los trabajos se pueden ejecutar durante más tiempo después de que se haya realizado la actualización de la infraestructura. Este tiempo adicional se configura en la configuración `jobExecutionTimeoutMinutes`. De forma predeterminada, la configuración de `jobExecutionTimeoutMinutes` es 30 minutos.

A medida que se dispone de capacidad en el entorno informático, se lanzan nuevas instancias con la configuración actualizada y se inician los trabajos en las nuevas instancias. A medida que se completan todos los trabajos en las instancias con la configuración anterior, las instancias antiguas se cancelan. Lo que significa que la capacidad disponible es que la cantidad deseada de vCPU está por debajo de la cantidad máxima de vCPU en al menos tantas vCPU como las que requiere el tipo de instancia más pequeño.

Actualizaciones de la infraestructura

Es necesaria una actualización de la infraestructura para cambiar algunos ajustes de un entorno informático. Si se cambia alguna de las siguientes configuraciones, se inicia una actualización de la infraestructura:

Important

El entorno informático debe usar el rol vinculado al servicio AWSServiceRoleForBatch para realizar cambios que requieran una actualización de la infraestructura.
Si el entorno de cómputo usa un rol vinculado a un servicio, no se puede cambiar para usar un rol de IAM normal. Del mismo modo, si el entorno de cómputo tiene un rol de IAM normal, no se puede cambiar para usar un rol vinculado a un servicio. Por lo tanto, solo puede realizar actualizaciones de infraestructura en entornos informáticos que se hayan creado mediante un rol vinculado a un servicio.

- La estrategia de asignación (`allocationStrategy`, debe ser `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, o `SPOT_PRICE_CAPACITY_OPTIMIZED`. Si la estrategia de asignación original es `BEST_FIT`, no se admiten las actualizaciones de infraestructura).

  > **Note**
  >
  > Se recomienda utilizar `SPOT_PRICE_CAPACITY_OPTIMIZED` en lugar de `SPOT_CAPACITY_OPTIMIZED` en la mayoría de los casos.

- Porcentaje de oferta (`bidPercentage`)
- Configuración de EC2 (`ec2Configuration`)
- Par de claves (`ec2KeyPair`)
- ID de imagen (`imageId`)
- Rol de instancia (`instanceRole`)
- Tipos de instancias (`instanceTypes`)
- Plantilla de lanzamiento (`launchTemplate`)
- Grupo de ubicación (`placementGroup`)
- Grupos de seguridad (`securityGroupIds`)
- Subredes de la VPC (`subnets`)
- Etiquetas EC2 (`tags`)
- Tipo de entorno informático (`type`, puede ser uno de `EC2` o`SPOT`)
- Si se debe actualizar a la AMI más reciente compatible AWS Batch durante una actualización de infraestructura `updateToLatestImageVersion`

Actualización del ID de la AMI

Durante una actualización de la infraestructura, el ID de la AMI del entorno informático puede cambiar en función de si las AMI se especifican en alguna de estas tres configuraciones. Las AMI se especifican en la `imageId` (en `computeResources`), `imageIdOverride` (en `ec2Configuration`), o de lanzamiento especificada en `launchTemplate`. Suponga que no se especifica ningún ID de AMI en ninguna de esas configuraciones y que la configuración `updateToLatestImageVersion` es `true`. A continuación, la última AMI optimizada de Amazon ECS compatible con AWS Batch se utiliza para cualquier actualización de la infraestructura.

Si se especifica una ID de AMI en al menos una de estas configuraciones, la actualización depende de la configuración proporcionada por la ID de AMI utilizada antes de la actualización. Al crear un entorno informático, la prioridad a la hora de seleccionar un ID de AMI es primero la plantilla de lanzamiento, después la configuración `imageId` y, por último, la configuración `imageIdOverride`. Sin embargo, si el ID de la AMI que se utiliza proviene de la plantilla de lanzamiento, al actualizar la configuración `imageId` o `imageIdOverride`, no se actualiza el ID de la AMI. La única forma de actualizar un ID de AMI seleccionado en la plantilla de lanzamiento es actualizar la plantilla de lanzamiento. Si el parámetro de versión de la plantilla de lanzamiento es `$Default` o `$Latest`, se evalúa la versión por defecto o la más reciente de la plantilla de lanzamiento especificada. Si se selecciona un ID de AMI diferente de forma predeterminada o se selecciona la última versión de la plantilla de lanzamiento, ese ID de AMI se utiliza en la actualización.

Si la plantilla de lanzamiento no se usó para seleccionar el ID de AMI, se usa el ID de AMI que se especifica en los parámetros `imageId` o `imageIdOverride`. Si se especifican ambos, se utiliza el ID de AMI especificado en el parámetro `imageIdOverride`.

Supongamos que el entorno informático utiliza un ID de AMI especificado por los parámetros `imageId`, `imageIdOverride`, o `launchTemplate`, y usted desea utilizar la última AMI optimizada para Amazon ECS compatible con AWS Batch. A continuación, la actualización debe eliminar la configuración que proporcionaba los ID de AMI. Para `imageId`, es necesario especificar una cadena vacía para ese parámetro. Para `imageIdOverride`, es necesario especificar una cadena vacía para el parámetro de `ec2Configuration`.

Si el ID de la AMI proviene de la plantilla de lanzamiento, puede cambiarlo AWS Batch por la última AMI optimizada para Amazon ECS que sea compatible con una de las siguientes formas:

- Elimine la plantilla de lanzamiento especificando una cadena vacía para el parámetro `launchTemplateId` o `launchTemplateName`. Esto elimina toda la plantilla de lanzamiento, en lugar de solo el ID de la AMI.
- Si la versión actualizada de la plantilla de lanzamiento no especifica un ID de AMI, el parámetro `updateToLatestImageVersion` debe estar establecido en `true`.

Entornos de computación de Amazon EKS

Cómo empezar con AWS Batch Amazon EKS proporciona una breve guía para crear entornos de cómputo de EKS. En esta sección se proporcionan más detalles sobre los entornos de cómputo de Amazon EKS.

Temas

Selección de AMI predeterminada

Al crear un entorno informático Amazon EKS, no necesita especificar una imagen de máquina de Amazon (AMI). AWS Batch selecciona una AMI optimizada para Amazon EKS en función de los tipos de Kubernetes versión e instancia que se especifican en la CreateComputeEnvironmentsolicitud. En general, recomendamos que utilice la selección de AMI predeterminada. Para obtener más información sobre AMI optimizadas para Amazon EKS, consulte AMI de Linux optimizadas para Amazon EKS en la Guía del usuario de Amazon EKS.

Ejecute el siguiente comando para ver qué tipo de AMI AWS Batch está seleccionado para su entorno informático Amazon EKS. El siguiente ejemplo es un tipo de instancia que no es de GPU.

```
# compute CE example: indicates Batch has chosen the AL2 x86 or ARM EKS 1.29 AMI,
 depending on instance types
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
    | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2",
    "imageKubernetesVersion": "1.29"
  }
]
```

El siguiente ejemplo es un tipo de instancia de GPU.

```
# GPU CE example: indicates Batch has choosen the AL2 x86 EKS Accelerated 1.29 AMI
$ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
    | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
```

```
    "imageType": "EKS_AL2_NVIDIA",
    "imageKubernetesVersion": "1.29"
  }
]
```

Versiones de Kubernetes compatibles

AWS Batch en Amazon EKS actualmente es compatible con las siguientes Kubernetes versiones:

- `1.29`
- `1.28`
- `1.27`
- `1.26`
- `1.25`
- `1.24`
- `1.23`

Es posible que aparezca un mensaje de error similar al siguiente cuando utilice la operación de `CreateComputeEnvironment` API o la operación de `UpdateComputeEnvironment` API para crear o actualizar un entorno de computación. Este problema se produce si especifica una versión de Kubernetes no compatible en `EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Para resolver este problema, elimine el entorno de procesamiento y vuelva a crearlo con una versión de Kubernetes compatible.

Puede realizar una actualización de una versión menor en su clúster de Amazon EKS. Por ejemplo, puede actualizar el clúster de `1.xx` a `1.yy` incluso si la versión secundaria no es compatible.

Sin embargo, es posible que el estado del entorno de computación cambie a `INVALID` después de una actualización de la versión principal. Por ejemplo, si realiza una actualización de una versión principal de `1.xx` a `2.yy`. Si la versión principal no es compatible con AWS Batch, aparecerá un mensaje de error similar al siguiente.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Actualización la versión de Kubernetes del entorno de computación

Con AWS Batch, puede actualizar la Kubernetes versión de un entorno informático para que sea compatible con las actualizaciones de clústeres de Amazon EKS. La Kubernetes versión de un entorno de cómputo es la versión AMI de Amazon EKS para los Kubernetes nodos que se AWS Batch lanzan para ejecutar tareas. Puede realizar una actualización de Kubernetes versión en sus nodos de Amazon EKS antes o después de actualizar la versión del plano de control del clúster de Amazon EKS. Le recomendamos que actualice los nodos después de actualizar el plano de control. Para obtener más información, consulte Actualizar la versión de Kubernetes del clúster de Amazon EKS en la Guía del usuario de Amazon EKS.

Para actualizar la versión de Kubernetes de un entorno de computación, utilice la operación UpdateComputeEnvironment API.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

Responsabilidad compartida de los nodos Kubernetes

El mantenimiento de los entornos de computación es una responsabilidad compartida.

- No cambies ni elimines AWS Batch nodos, etiquetas, manchas, espacios de nombres, plantillas de lanzamiento ni grupos de escalado automático. No añada información contaminada a los nodos gestionados. AWS Batch Si realiza alguno de estos cambios, su entorno de computación no será compatible y se producirán errores, incluidas las instancias inactivas.
- No dirija sus pods a los nodos AWS Batch gestionados. Si dirige sus pods a los nodos gestionados, se producirán problemas de escalado y las colas de trabajos se atascarán. Ejecuta cargas de trabajo que no se usen AWS Batch en nodos autogestionados o grupos de nodos gestionados. Para obtener más información, consulte Grupos de nodos administrados en la Guía del usuario de Amazon EKS.
- Puede seleccionar DaemonSet a para que se ejecute en nodos AWS Batch gestionados. Para obtener más información, consulte Ejecuta un DaemonSet nodo AWS Batch gestionado.

AWS Batch no actualiza automáticamente las AMI del entorno de cómputo. Es su responsabilidad actualizarlas. Para actualizar las AMI a la versión más reciente, ejecute el siguiente comando.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources 'updateToLatestImageVersion=true'
```

AWS Batch no actualiza automáticamente la Kubernetes versión. Ejecute el siguiente comando para actualizar la versión Kubernetes de su entorno de computación a la *1.23*.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

Al actualizar a una AMI o a una Kubernetes versión más reciente, puede especificar si desea finalizar los trabajos cuando se actualicen (`terminateJobsOnUpdate`) y cuánto tiempo esperar antes de reemplazar una instancia si los trabajos en ejecución no terminan (`jobExecutionTimeoutMinutes`.) Para obtener más información, consulte Actualizar entornos informáticos y la política de actualización de la infraestructura (UpdatePolicy) establecida en la operación UpdateComputeEnvironment de la API.

Ejecuta un DaemonSet nodo AWS Batch gestionado

AWS Batch perjudica a los Kubernetes nodos AWS Batch gestionados. Puede seleccionar a DaemonSet para que se ejecute en nodos AWS Batch gestionados con lo siguiente`tolerations`.

```
tolerations:
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
```

Otra forma de hacerlo es con lo siguiente `tolerations`.

```
tolerations:
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
  effect: "NoSchedule"
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
  effect: "NoExecute"
```

Personalización con plantillas de lanzamiento

AWS Batch en Amazon EKS admite plantillas de lanzamiento. Existen limitaciones en cuanto a lo que puede hacer su plantilla de lanzamiento.

> **Important**
>
> AWS Batch se ejecuta`/etc/eks/bootstrap.sh`. No ejecute `/etc/eks/bootstrap.sh` en su plantilla de lanzamiento ni en sus scripts cloud-init user-data. Puede añadir parámetros adicionales además del parámetro `--kubelet-extra-args` a bootstrap.sh. Para ello, defina la variable `AWS_BATCH_KUBELET_EXTRA_ARGS` en el archivo `/etc/aws-batch/batch.config`. Consulte el ejemplo siguiente para obtener información detallada.

> **Note**
>
> Si la plantilla de lanzamiento se cambia después de CreateComputeEnvironmentllamarla, UpdateComputeEnvironmenthay que llamarla para evaluar la versión de la plantilla de lanzamiento que se va a reemplazar.

Temas

- Añadir argumentos adicionales kubelet
- Configuración del tiempo de ejecución del contenedor
- Montaje de un volumen de Amazon EFS
- Compatibilidad con IPv6

Añadir argumentos adicionales **`kubelet`**

AWS Batch admite la adición de argumentos adicionales al `kubelet` comando. Para ver una lista completa de parámetros que admite kubelet, consulte la Documentación de Kubernetes. En el siguiente ejemplo, *`--node-labels mylabel=helloworld`* se agrega a la línea de comandos de `kubelet`.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="
```

```
--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo AWS_BATCH_KUBELET_EXTRA_ARGS=\"--node-labels mylabel=helloworld\" >> /etc/aws-
batch/batch.config

--==MYBOUNDARY==--
```

Configuración del tiempo de ejecución del contenedor

Puede utilizar la variable de entorno AWS Batch `CONTAINER_RUNTIME` para configurar el tiempo de ejecución del contenedor en un nodo administrado. En el siguiente ejemplo, se establece el tiempo de ejecución del contenedor en `containerd` cuando se ejecuta `bootstrap.sh`. Para obtener más información, consulte `containerd` en la documentación de Kubernetes.

> **Note**
>
> La variable de entorno de `CONTAINER_RUNTIME` equivale a la opción `--container-runtime` de `bootstrap.sh`. Para obtener más información, consulte Options en la documentación de Kubernetes.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo CONTAINER_RUNTIME=containerd >> /etc/aws-batch/batch.config

--==MYBOUNDARY==--
```

Montaje de un volumen de Amazon EFS

Puede utilizar plantillas de lanzamiento para montar los volúmenes en el nodo. En el siguiente ejemplo, se utilizan los ajustes `cloud-config` `packages` y `runcmd`. Para obtener más información, consulte ejemplos de configuración de nube en la cloud-initdocumentación.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs _netdev,noresvport,tls,iam 0 0"
 >> /etc/fstab
- mount -t efs -o tls ${file_system_id_01}:/ ${efs_directory}

--==MYBOUNDARY==--
```

Para usar este volumen en el trabajo, debe añadirse al parámetro eksProperties a. RegisterJobDefinition El siguiente ejemplo es una gran parte de la definición del trabajo.

```
{
    "jobDefinitionName": "MyJobOnEks_EFS",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["ls", "-la", "/efs"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi"
```

```
                        }
                    },
                    "volumeMounts": [
                        {
                            "name": "efs-volume",
                            "mountPath": "/efs"
                        }
                    ]
                }
            ],
            "volumes": [
                {
                    "name": "efs-volume",
                    "hostPath": {
                        "path": "/mnt/efs"
                    }
                }
            ]
        }
    }
}
```

En el nodo, el volumen de Amazon EFS está montado en el directorio `/mnt/efs`. En el contenedor del trabajo de Amazon EKS, el volumen se monta en el `/efs` directorio.

Compatibilidad con IPv6

AWS Batch admite clústeres de Amazon EKS que tienen direcciones IPv6. No se requieren personalizaciones para obtener AWS Batch soporte. Sin embargo, antes de empezar, le recomendamos que revise las consideraciones y condiciones que se describen en la sección Asignación de direcciones IPv6 a los pods y servicios de la Guía del usuario de Amazon EKS.

Administración de la memoria de los recursos informáticos de las

Cuando el agente de contenedores de Amazon ECS registra un recurso informático en un entorno informático, el agente debe determinar cuánta memoria tiene disponible el recurso informático para reservarla para sus trabajos. Debido a la sobrecarga de memoria de la plataforma y a la memoria ocupada por el núcleo del sistema, este número es diferente de la cantidad de memoria instalada para las instancias de Amazon EC2. Por ejemplo, una instancia `m4.large` tiene 8 GiB de memoria instalada. Sin embargo, esto no siempre se traduce en exactamente 8192 MiB de memoria disponible para los trabajos cuando se registra el recurso informático.

Suponga que especifica 8192 MiB para el trabajo y que ninguno de sus recursos informáticos tiene 8192 MiB o más de memoria disponible para cumplir este requisito. Entonces, el trabajo no se puede colocar en su entorno informático. Si está utilizando un entorno informático gestionado, AWS Batch debe lanzar un tipo de instancia mayor para dar cabida a la solicitud.

La AMI predeterminada de recursos informáticos de AWS Batch también reserva 32 MiB de memoria para el agente de contenedor de Amazon ECS y otros procesos críticos del sistema. Esta memoria no está disponible para la asignación de trabajos. Para obtener más información, consulte Reservar memoria del sistema.

El agente de contenedor de Amazon ECS utiliza la función `ReadMemInfo()` de Docker para consultar la memoria total disponible al sistema operativo. Linux proporciona utilidades de línea de comandos para determinar la memoria total.

Example - Determinar la memoria total en Linux

El comando free devuelve la memoria total reconocida por el sistema operativo.

```
$ free -b
```

A continuación, se muestra un ejemplo de salida para una instancia `m4.large` que ejecuta la AMI de Amazon Linux optimizada para Amazon ECS.

```
             total       used       free     shared    buffers     cached
Mem:    8373026816  348180480 8024846336      90112   25534464  205418496
-/+ buffers/cache:  117227520 8255799296
```

Esta instancia tiene 8373026816 bytes de memoria total. Esto significa que hay 7985 MiB disponibles para tareas.

Reservar memoria del sistema

Si ocupa toda la memoria de un recurso informático con sus trabajos, es posible que estos compitan por la memoria con procesos críticos del sistema y provoquen un fallo del sistema. El agente contenedor Amazon ECS proporciona una variable de configuración que se denomina `ECS_RESERVED_MEMORY`. Puede utilizar esta variable de configuración para eliminar un número determinado de MiB de memoria del pool asignado a sus trabajos. Este es un mecanismo eficaz que permite reservar memoria para los procesos críticos del sistema.

La AMI predeterminada de recursos informáticos de AWS Batch reserva 32 MiB de memoria para el agente de contenedor de Amazon ECS y otros procesos críticos del sistema.

Visualización de la memoria de los recursos informáticos de las

Puede ver con cuánta memoria registra un recurso informático en la consola de Amazon ECS o con la operación de la API DescribeContainerInstances. Si intenta maximizar la utilización de tus recursos proporcionando a tus trabajos tanta memoria como sea posible para un tipo de instancia en particular, puede observar la memoria disponible para ese recurso informático y luego asignar a sus trabajos esa cantidad de memoria.

Para ver la memoria de recurso informático

1. Abra la consola en https://console.aws.amazon.com/ecs/v2.
2. Elija Clústeres y, a continuación, elija el clúster que aloja los recursos informáticos que desee ver.

   El nombre del clúster del entorno informático comienza por el nombre del entorno informático.
3. Elija Infraestructura.
4. En Instancias de contenedor, elija la instancia de contenedor.
5. La sección Recursos y redes muestra la memoria registrada y disponible para el recurso informático.

   El valor de memoria Registrada es lo que el recurso informático registró con Amazon ECS cuando se lanzó por primera vez, y el valor de memoria Disponible es lo que aún no se ha asignado a los trabajos.

Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS

En AWS Batch en Amazon EKS, puede especificar los recursos que se ponen a disposición de un contenedor. Por ejemplo, puede especificar los valores `requests` o `limits` para los recursos vCPU y memoria.

Las siguientes son restricciones para especificar los recursos de vCPU:

- Debe especificarse al menos una vCPU de valor `requests` o `limits`.
- Una unidad vCPU equivale a un núcleo físico o virtual.
- El valor de vCPU debe introducirse en números enteros o en incrementos de 0,25.

- El valor de vCPU válido más bajo es 0,25.
- Si se especifican ambos, el valor `requests` debe ser menor o igual que el valor `limits`. De esta forma, puede configurar configuraciones de vCPU flexibles e invariables.
- Los valores de vCPU no se pueden especificar en formato milliCPU. Por ejemplo, `100m` no es un valor válido.
- AWS Batch utiliza el valor `requests` para tomar decisiones de escalado. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Las siguientes son restricciones para especificar los recursos de memoria:

- Debe especificarse al menos una memoria de valor `requests` o `limits`.
- Los valores de memoria deben estar en mebibytes (MiBs).
- Si se especifican ambos, el valor `requests` debe ser igual que el valor `limits`.
- AWS Batch utiliza el valor `requests` para tomar decisiones de escalado. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Las siguientes son restricciones para especificar los recursos de GPU:

- Si se especifican ambos, el valor `requests` debe ser igual que el valor `limits`.
- AWS Batch utiliza el valor `requests` para tomar decisiones de escalado. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Ejemplos de definiciones de trabajo

La siguiente AWS Batch de Amazon EKS configura los recursos compartidos de vCPU flexibles. Esto permite a AWS Batch de Amazon EKS utilizar toda la capacidad de vCPU para el tipo de instancia. Sin embargo, si hay otros trabajos en ejecución, al trabajo se le asigna un máximo de 2 vCPU. La memoria está limitada a 2 GB.

```
{
    "jobDefinitionName": "MyJobOnEks_Sleep",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
```

```
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["sleep", "60"],
                    "resources": {
                        "requests": {
                            "cpu": "2",
                            "memory": "2048Mi"
                        }
                    }
                }
            ]
        }
    }
}
```

La siguiente AWS Batch de trabajo de Amazon EKS tiene un `request` valor de 1 y asigna un máximo de 4 vCPU al trabajo.

```
{
    "jobDefinitionName": "MyJobOnEks_Sleep",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["sleep", "60"],
                    "resources": {
                        "requests": {
                            "cpu": "1"
                        },
                        "limits": {
                            "cpu": "4",
                            "memory": "2048Mi"
                        }
                    }
                }
            ]
        }
    }
}
```

La siguiente AWS Batch de trabajo de Amazon EKS establece un valor de vCPU `limits` de 1 y un valor de memoria `limits` de 1 GB.

```
{
    "jobDefinitionName": "MyJobOnEks_Sleep",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["sleep", "60"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi"
                        }
                    }
                }
            ]
        }
    }
}
```

Cuando AWS Batch traduce un AWS Batch en Amazon EKS a un pod de Amazon EKS, AWS Batch copia el valor `limits` al valor `requests`. Esto ocurre si no se especifica un valor `requests`. Al enviar la definición de trabajo del ejemplo anterior, el pod `spec` es el siguiente.

```
apiVersion: v1
kind: Pod
...
spec:
  ...
  containers:
    - command:
        - sleep
        - 60
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      resources:
        limits:
          cpu: 1
          memory: 1024Mi
        requests:
          cpu: 1
          memory: 1024Mi
```

```
...
```

Reservas de CPU y memoria de los nodos

AWS Batch se basa en la lógica predeterminada del archivo `bootstrap.sh` para las reservas de vCPU y memoria. Para obtener más información sobre el archivo `bootstrap.sh`, consulte bootstrap.sh. Al ajustar el tamaño de los recursos de vCPU y memoria, tenga en cuenta los ejemplos siguientes.

> **Note**
>
> Si no se está ejecutando ninguna instancia, las reservas de vCPU y memoria pueden afectar inicialmente a la lógica de escalado de AWS Batch y a la toma de decisiones. Una vez ejecutadas las instancias, AWS Batch ajusta las asignaciones iniciales.

Ejemplo de reserva de CPU de nodo

El valor de reserva de CPU se calcula en milinúcleos mediante la cantidad total de vCPU disponibles para la instancia.

Número de vCPU	Porcentaje reservado
1	6 %
2	1 %
3-4	0,5%
4 y superior	0,25 %

Si se utilizan los valores anteriores, se aplica lo siguiente:

- El valor de reserva de CPU para una instancia `c5.large` con 2 vCPU es de 70 m. Se calcula de la siguiente manera: (1*60) + (1*10) = 70 m.
- El valor de reserva de CPU para una instancia `c5.24xlarge` con 96 vCPU es de 310 m. Se calcula de la siguiente manera: (1*60) + (1*10) + (2*5) + (92*2.5) = 310 m.

En este ejemplo, hay 1930 unidades de vCPU milicore (calculadas entre 2000 y 70) disponibles para ejecutar trabajos en una instancia `c5.large`. Supongamos que su trabajo requiere unidades vCPU 2 (2*1000 m), el trabajo no cabe en una sola instancia `c5.large`. Sin embargo, un trabajo que requiere unidades de vCPU `1.75` es adecuado.

Ejemplo de reserva de memoria de nodo

El valor de reserva de memoria se calcula en mebibytes de la siguiente manera:

- Capacidad de la instancia en mebibytes. Por ejemplo, una instancia de 8 GB equivale a 7,748 MiB.
- El valor `kubeReserved`. El valor `kubeReserved` es la cantidad de memoria que se debe reservar para los daemon del sistema. El valor `kubeReserved` se calcula de la siguiente manera: (11 * número máximo de pods que admite el tipo de instancia) + 255). Para obtener información sobre el número máximo de pods que admite un tipo de instancia, consulte eni-max-pods.txt
- El valor `HardEvictionLimit`. Cuando la memoria disponible cae por debajo del valor `HardEvictionLimit`, la instancia intenta expulsar los pods.

La fórmula para calcular la memoria asignable es la siguiente: (*`instance_capacity_in_MiB`*) - (11 * (*`maximum_number_of_pods`*)) - 255 - (*`HardEvictionLimit value.`*)).

Una instancia `c5.large` admite hasta 29 pods. Para una instancia `c5.large` de 8 GB con un valor `HardEvictionLimit` de 100 MiB, la memoria asignable es 7074 MiB. Esto se calcula de la siguiente manera: (7748 - (11 * 29) -255 -100) = 7074 MiB. En este ejemplo, un trabajo MiB de 8192 no cabe en esta instancia aunque sea una instancia 8 gibibyte (GiB).

DaemonSets

Cuando utilice DaemonSets, tenga en cuenta lo siguiente:

- Si no se está ejecutando ninguna instancia AWS Batch de Amazon EKS, DaemonSets puede afectar inicialmente a la lógica del escalado y a la toma de decisiones de AWS Batch. AWS Batch inicialmente asigna 0,5 unidades de vCPU y 500 MiB para DaemonSets previsto. Una vez ejecutadas las instancias, AWS Batch ajusta las asignaciones iniciales.
- Si DaemonSet define los límites de vCPU o memoria, los AWS Batch en Amazon EKS tienen menos recursos. Le recomendamos que mantenga el número de DaemonSets asignados a los AWS Batch trabajos lo más bajo posible.

Políticas de programación

Puede utilizar las políticas de programación para configurar cómo se asignan los recursos informáticos de una cola de trabajos entre los usuarios o las cargas de trabajo. Mediante las políticas de programación, puede asignar diferentes identificadores de reparto justo a las cargas de trabajo o a los usuarios. AWS Batchasigna a cada identificador de reparto justo un porcentaje del total de recursos disponibles durante un período de tiempo.

El porcentaje de reparto justo se calcula utilizando los valores `shareDecaySeconds` y `shareDistribution`. Puede añadir tiempo al análisis del reparto justo asignando un tiempo de caída de participación de las acciones a la política. Agregar tiempo da más peso al tiempo y menos al peso definido. Si especifica una reserva informática, puede reservar los recursos informáticos para los identificadores de reparto justo que no estén activos. Para obtener más información, consulte Parámetros de la política de programación.

Temas

Creación de una política de programación

Antes de crear una cola de trabajos con una política de programación, debe crear una política de programación. Al crear una política de programación, se asocian uno o más identificadores de reparto equitativo o prefijos de identificador de reparto equitativo a las ponderaciones de la cola y, si lo prefiere, se le asigna un período de reducción y se calcula la reserva a la política.

Para crear una política de programación

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la región a utilizar.
3. En el panel de navegación, elija Políticas de programación, Crear.
4. Para Nombre, escriba un nombre único para la política de programación. Se admiten hasta 128 letras (mayúsculas y minúsculas), números, guiones y caracteres de subrayado.
5. (Opcional) En Segundos de decaimiento de cuota, introduzca un valor entero para el tiempo de decaimiento de cuota de la política de programación. Un tiempo de decaimiento de la cuota

más largo utilizará considerar el uso de recursos informáticos durante un tiempo más largo al programar los trabajos. Esto puede permitir que los trabajos que utilizan un identificador de reparto equitativo utilicen temporalmente más recursos de computación de los que permitiría el peso de ese identificador si ese identificador de reparto equitativo no hubiera utilizado recursos de computación recientemente.

6. (Opcional) En Reserva de computación, introduzca un valor entero para la reserva de cálculo de la política de programación. La reserva de computación mantendrá algunas vCPU en reserva para usarlas como identificadores de reparto justo que no estén activos actualmente.

   La proporción reservada es `(`*`computeReservation`*`/100)^`*`ActiveFairShares`*, donde ActiveFairShares es el número de identificadores activos de la cuota justa.

   Por ejemplo, un valor `computeReservation` de 50 indica que AWS Batch debe reservar el 50 % de la VCPU máxima disponible si solo hay un identificador de acciones justas, el 25 % si hay dos identificadores de acciones justas y el 12,5 % si hay tres identificadores de acciones justas. Un valor `computeReservation` de 25 indica que AWS Batch debe reservar el 25 % de la VCPU máxima disponible si solo hay un identificador de acciones justas, el 6,25 % si hay dos identificadores de acciones justas y el 1,56 % si hay tres identificadores de acciones justas.

7. En la sección Atributos de cuota, puede especificar el identificador de reparto equitativo y el peso de cada identificador de reparto equitativo para asociarlos a la política de programación.

   a. Seleccione Añadir identificador de reparto.

   b. En Identificador de reparto, especifique el identificador de reparto justo. Si la cadena termina con «*», se convierte en un prefijo identificador de reparto equitativo que se utiliza para hacer coincidir los identificadores de repartos equitativos de los puestos de trabajo. Todos los identificadores de reparto equitativo y los prefijos de los identificadores de reparto equitativo de una política de programación deben ser únicos y no pueden superponerse. Por ejemplo, los identificadores de reparto equitativa no pueden tener el prefijo «UserA*» y el identificador de reparto equitativo «userA1» en la misma política de programación.

   c. En Factor de ponderación, especifique la ponderación relativa para el identificador de reparto justo. El valor predeterminado es 1.0. Un valor más bajo tiene una mayor prioridad para los recursos informáticos. Si se utiliza un prefijo identificador de reparto equitativo, los trabajos con identificadores de reparto equitativo que comiencen con ese prefijo compartirán el factor de ponderación. De hecho, esto aumenta el factor de ponderación de esos puestos de trabajo, lo que reduce su prioridad individual, a la vez que mantiene el mismo factor de ponderación para el prefijo identificador de reparto equitativo.

8. (Opcional) En la sección Etiquetas, puede especificar la clave y el valor de cada etiqueta que desea asociar a la política de programación. Para obtener más información, consulte Etiquetado de los recursos de AWS Batch.
9. Seleccione Enviar para terminar y crear su política de programación.

Plantilla de política de programación

A continuación, se muestra una plantilla de política de programación vacía. Utilice esta plantilla para crear su política de programación, que posteriormente se puede guardar en un archivo y utilizarse con la opción de la AWS CLI `--cli-input-json`. Para obtener más información sobre estos parámetros, consulte CreateSchedulingPolicy en la AWS Batch Referencia de API.

```
{
    "name": "",
    "fairsharePolicy": {
        "shareDecaySeconds": 0,
        "computeReservation": 0,
        "shareDistribution": [
            {
                "shareIdentifier": "",
                "weightFactor": 0.0
            }
        ]
    },
    "tags": {
        "KeyName": ""
    }
}
```

> **Note**
>
> Puede generar la plantilla de cola de trabajos mostrada anteriormente con el siguiente comando de la AWS CLI.
>
> ```
> $ aws batch create-scheduling-policy --generate-cli-skeleton
> ```

Parámetros de la política de programación

Las políticas de programación se dividen en tres componentes básicos: el nombre, la política de reparto justo y las etiquetas de la política de programación.

Temas

Nombre de la política de programación

`name`

El nombre de su política de programación. Se admiten hasta 128 letras (mayúsculas y minúsculas), números, guiones y caracteres de subrayado.

Tipo: String

Obligatorio: sí

Política de reparto justo

`fairsharePolicy`

La política de reparto justo de la política de programación.

```
"fairsharePolicy": {
   "computeReservation": number,
   "shareDecaySeconds": number,
   "shareDistribution": [
      {
         "shareIdentifier": "string",
         "weightFactor": number
      }
   ]
}
```

Tipo: objeto

Obligatorio: no

`computeReservation`

Un valor que se usa para reservar parte de la VCPU máxima disponible para los identificadores de la parte justa que aún no han sido utilizados.

La proporción reservada es `(`*computeReservation*`/100)^`*ActiveFairShares*, donde ActiveFairShares es el número de identificadores activos de la cuota justa.

Por ejemplo, un valor `computeReservation` de 50 indica que AWS Batch debe reservar el 50 % de la VCPU máxima disponible si solo hay un identificador activo de reparto justo, el 25 % si hay dos identificadores activos de reparto justo y el 12,5 % si hay tres identificadores activos de reparto justos. Un valor `computeReservation` de 25 indica que AWS Batch debe reservar el 25 % de la VCPU máxima disponible si solo hay un identificador activo de reparto justo, el 6,25 % si hay dos identificadores activos de reparto justo y el 1,56 % si hay tres identificadores activos de reparto justo.

Tipo: entero

Rango válido: valor mínimo de 0. Valor máximo de 99.

Obligatorio: no

`shareDecaySeconds`

El período de tiempo que se utilizará para calcular el porcentaje de reparto justo para cada identificador de reparto justo en uso. Un valor de cero (0) indica que solo se debe medir el uso actual. El decaimiento permite que los trabajos más recientes tengan más peso que los anteriores.

Tipo: entero

Rango válido: valor mínimo de 0. Valor máximo de 604800 (1 semana).

Obligatorio: no

`shareDistribution`

Matriz de objetos que contiene las ponderaciones de los identificadores de reparto justo para la política de reparto justo. Los identificadores de reparto justo que no se incluyen tienen un peso predeterminado de `1.0`.

```
"shareDistribution": [
   {
      "shareIdentifier": "string",
      "weightFactor": number
   }
]
```

Tipo: matriz

Obligatorio: no

`shareIdentifier`

Un identificador de reparto justo o un prefijo de identificador de reparto justo. Si la cadena termina con '*', dicha cadena especifica un prefijo de reparto justo para los identificadores de reparto justo que comienzan con ese prefijo. Por ejemplo, si el valor es `UserA*` y el `weightFactor` es 1 y hay dos identificadores de reparto justo que comienzan por `UserA`, cada uno de esos identificadores de reparto justo tendrá una ponderación de 2; si hay cinco identificadores de reparto justo de este tipo, cada uno tendrá una ponderación de 5.

La lista de identificadores de reparto justo y prefijos de identificadores de reparto justo de una política de reparto justo no puede solaparse. Por ejemplo, no puede haber un prefijo identificador de reparto justo de `UserA*` y un identificador de reparto justo de `UserA-1` en la misma política de reparto justo.

Tipo: String

Obligatorio: sí

`weightFactor`

El factor de ponderación para el identificador de reparto justo. El valor predeterminado es 1.0. Un valor más bajo tiene una mayor prioridad para los recursos informáticos. Por ejemplo, los trabajos que utilizan un identificador de recurso compartido con un factor de peso de 0,125 (1/8) obtienen 8 veces los recursos de computación de los trabajos que utilizan un identificador de recurso compartido con un factor de peso de 1.

El valor más pequeño admitido es 0,0001 y el más grande es 999,9999.

Tipo: Flotante

Obligatorio: no

Etiquetas

`tags`

Etiquetas de pares clave-valor para asociarlas a la política de programación. Para obtener más información, consulte Etiquetado de los recursos de AWS Batch.

Tipo: mapa de cadena a cadena

Obligatorio: no

Organice las tareas AWS Batch con las máquinas de estados de Step Functions en la consola de AWS Batch

Puede utilizar la consola de AWS Batch para ver detalles de las máquinas de estado de Step Functions y las funciones que utilizan.

Secciones

Consulta de los detalles de las máquinas de estado

La consola de AWS Batch muestra una lista de las máquinas de estado en la Región de AWS actual que contiene al menos un paso de flujo de trabajo que envía un trabajo AWS Batch.

Elija una máquina de estado para ver una representación gráfica del flujo de trabajo. Los pasos resaltados en azul representan trabajos AWS Batch. Utilice los controles del gráfico para acercar, alejar y centrar el gráfico.

> **Note**
>
> Cuando se hace referencia dinámicamente con JsonPath a un trabajo AWS Batch en la definición de máquina de estado, los detalles del trabajo no se pueden mostrar en la consola de AWS Batch. En su lugar, el nombre del trabajo se muestra como una referencia dinámica y los pasos correspondientes del gráfico aparecen atenuados.

Para ver los detalles de una máquina de estado

1. Abra la consola de AWS Batch de la Página de orquestación del flujo de trabajo con Step Functions.
2. Creación de una máquina de estado.
   <result>

   La consola de AWS Batch abre la página Detalles.

```
</result>
```

Para obtener más información, consulte Step Functions en la Guía para desarrolladores de AWS Step Functions.

Edición de una máquina de estado

Cuando desee editar una máquina de estado, AWS Batch abre la página Editar definición de la consola de Step Functions.

Para editar una máquina de estado

1. Abra la consola de AWS Batch de la Página de orquestación del flujo de trabajo con Step Functions.
2. Creación de una máquina de estado.
3. Elija Editar.

   La consola de Step Functions abre la página Editar definición.
4. Edite la máquina de estado y elija Guardar.

Para obtener más información acerca de la edición de máquinas de estado, consulte Lenguaje de máquinas de estado de Step Functions en la Guía para desarrolladores de AWS Step Functions.

Ejecución de una máquina de estado

Cuando desee ejecutar una máquina de estado, AWS Batch abre la página Nueva ejecución de la consola de Step Functions.

Para ejecutar una máquina de estado

1. Abra la consola de AWS Batch de la Página de orquestación del flujo de trabajo con Step Functions.
2. Creación de una máquina de estado.
3. Elija Ejecutar.

   La consola de Step Functions abre la página Nueva ejecución.
4. (Opcional) Edite la máquina de estado y elija Iniciar ejecución.

Para obtener más información acerca de cómo ejecutar máquinas de estado, consulte Conceptos de ejecución de máquinas de estado de Step Functions en la Guía para desarrolladores de AWS Step Functions.

AWS Batch en AWS Fargate

La tecnología AWS Fargate se puede utilizar con AWS Batch para ejecutar contenedores sin tener que administrar servidores ni clústeres de instancias de Amazon EC2. Con AWS Fargate, ya no tendrá que aprovisionar, configurar ni escalar clústeres de máquinas virtuales para ejecutar los contenedores. De esta manera, se elimina la necesidad de elegir tipos de servidores, decidir cuándo escalar los clústeres u optimizar conjuntos de clústeres.

Al ejecutar los trabajos con los recursos de Fargate, la aplicación se empaqueta en contenedores, se especifican los requisitos de CPU y de memoria, se definen las políticas de IAM y de redes y se lanza la aplicación. Cada trabajo de Fargate tiene su propio límite de aislamiento y no comparte el kernel subyacente, los recursos de CPU, los recursos de memoria ni la interfaz de red elástica con otro trabajo.

Contenido

Cuándo usar Fargate

Recomendamos usar Fargate en la mayoría de los casos. Fargate lanza y escala el cómputo para que se ajuste perfectamente a los requisitos de recursos que especifique para el contenedor. Con Fargate, no es necesario aprovisionar en exceso los servidores ni pagar por ellos. Tampoco tiene que preocuparse por los detalles de los parámetros relacionados con la infraestructura, como el tipo de instancia. Cuando es necesario escalar verticalmente el entorno de cómputo, los trabajos que se ejecutan en los recursos de Fargate pueden comenzar más rápidamente. Por lo general, se tardará unos minutos hasta que se lance la instancia de Amazon EC2. Sin embargo, los trabajos que se ejecutan en Fargate se pueden aprovisionar en unos 30 segundos. El tiempo exacto necesario depende de varios factores, como el tamaño de la imagen del contenedor y la cantidad de trabajos.

No obstante, recomendamos utilizar Amazon EC2 si sus trabajos requieren cualquiera de las siguientes opciones:

- Más de 16 vCPU

- Más de 120 gibibytes (GiB) de memoria
- Una GPU
- Imagen de máquina de Amazon (AMI) personalizada
- Alguno de los parámetros de LinuxParameters

Si tiene un gran número de trabajos, le recomendamos utilizar la infraestructura de Amazon EC2. Por ejemplo, si el número de trabajos que se ejecutan simultáneamente supera la limitación de regulación de Fargate. Esto se debe a que, con EC2, los trabajos se pueden enviar a un ritmo mayor a los recursos de EC2 que a los recursos de Fargate. Además, se pueden ejecutar más trabajos de forma simultánea cuando se utiliza EC2. Para obtener más información, consulte Fargate Service Quotas AWS en la Guía para desarrolladores de Amazon Elastic Container Service.

Definiciones de trabajo en Fargate

Los trabajos de AWS Batch alojados en Fargate no admiten todos los parámetros de definición de tareas disponibles. Algunos parámetros no son compatibles, y otros se comportan de forma distinta para trabajos de Fargate.

La siguiente lista describe los parámetros de definición de trabajo que no son válidos o están restringidos de otro modo en los trabajos de Fargate.

`platformCapabilities`

Debe especificarse como `FARGATE`.

```
"platformCapabilities": [ "FARGATE" ]
```

`type`

Debe especificarse como `container`.

```
"type": "container"
```

Parámetros de `containerProperties`

`executionRoleArn`

Debe especificarse para trabajos que se ejecutan en recursos de Fargate. Para obtener más información, consulte Roles de IAM para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

```
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
```

`fargatePlatformConfiguration`

(Opcional, solo para las definiciones de trabajo de Fargate). Especifica la versión de la plataforma Fargate o `LATEST` para una versión reciente de la plataforma. Los valores posibles de `platformVersion` son `1.3.0`, `1.4.0` y `LATEST` (predeterminado).

```
"fargatePlatformConfiguration": { "platformVersion": "1.4.0" }
```

`instanceType`, `ulimits`

No se aplica a trabajos que se ejecutan en recursos de Fargate.

`memory`, `vcpus`

Esta configuración debe especificarse en `resourceRequirements`

`privileged`

No especifique este parámetro o especifique `false`.

```
"privileged": false
```

`resourceRequirements`

Los requisitos de memoria y vCPU deben especificarse mediante los valores admitidos. Los recursos de GPU no son compatibles con los recursos de Fargate.

Si usa GuardDuty Runtime Monitoring, hay una ligera sobrecarga de memoria para el agente GuardDuty de seguridad. Por lo tanto, el límite de memoria debe incluir el tamaño del agente de GuardDuty seguridad. Para obtener información sobre los límites de memoria del agente de GuardDuty seguridad, consulte los límites de CPU y memoria en la Guía del GuardDuty usuario. Para obtener información sobre las prácticas recomendadas, consulte Cómo corregir los errores

de falta de memoria en mis tareas de Fargate después de activar Runtime Monitoring en la Guía para desarrolladores de Amazon ECS.

```
"resourceRequirements": [
  {"type": "MEMORY", "value": "512"},
  {"type": "VCPU",   "value": "0.25"}
]
```

Parámetros de `linuxParameters`

`devices`, `maxSwap`, `sharedMemorySize`, `swappiness`, `tmpfs`

No se aplica a trabajos que se ejecutan en recursos de Fargate.

Parámetros de `logConfiguration`

`logDriver`

Solo se admiten `awslogs` y `splunk`. Para obtener más información, consulte Uso del controlador de registros awslogs.

Miembros en `networkConfiguration`

`assignPublicIp`

Si la subred privada no tiene una puerta de enlace NAT conectada para enviar tráfico a Internet, `assignPublicIp` debe ser «`ENABLED`». Para obtener más información, consulte AWS Batch función de IAM de ejecución.

Colas de trabajo en Fargate

AWS Batch las colas de trabajo en Fargate prácticamente no han cambiado. La única restricción es que todos los entornos informáticos que aparecen en la lista en `computeEnvironmentOrder` deben ser entornos informáticos de Fargate (`FARGATE` o `FARGATE_SPOT`). Los entornos informáticos de EC2 y Fargate no se pueden mezclar.

Entornos informáticos en Fargate

AWS Batch Lls entornos informáticos de Fargate no admiten todos los parámetros del entorno informático disponibles. Algunos parámetros no son compatibles. Otros tienen requisitos específicos para Fargate.

La siguiente lista describe los parámetros de entorno informático que no son válidos o están restringidos de otro modo en los trabajos de Fargate.

`type`

Este parámetro debe establecerse en `MANAGED`.

```
"type": "MANAGED"
```

Parámetros del objeto `computeResources`

`allocationStrategy`, `bidPercentage`, `desiredvCpus`, `imageId`, `instanceTypes`, `ec2Configuration`, `ec2KeyPair`, `instanceRole`, `launchTemplate`, `minvCpus`, `placementGroup`, `spotIamFleetRole`

No se aplican a los entornos informáticos de Fargate y no se pueden proporcionar.

`subnets`

Si las subredes enumeradas en este parámetro no tienen puertas de enlace NAT conectadas, el parámetro `assignPublicIp` de la definición del trabajo debe estar establecido en `ENABLED`.

`tags`

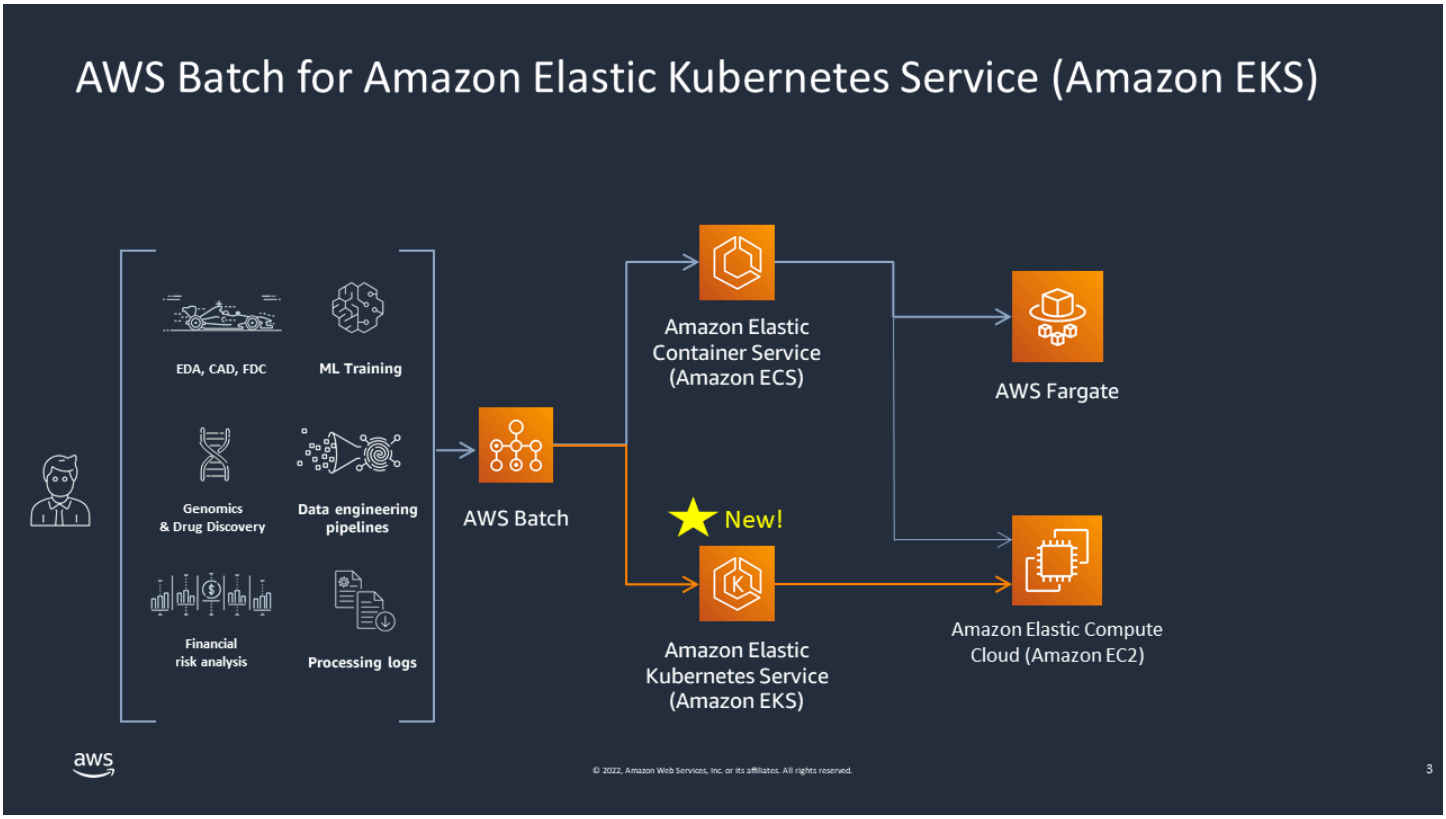
Esto no se aplica a los entornos informáticos de Fargate y no se puede proporcionar. Para especificar etiquetas para los entornos informáticos de Fargate, use el parámetro `tags` que no está en el objeto `computeResources`.

`type`

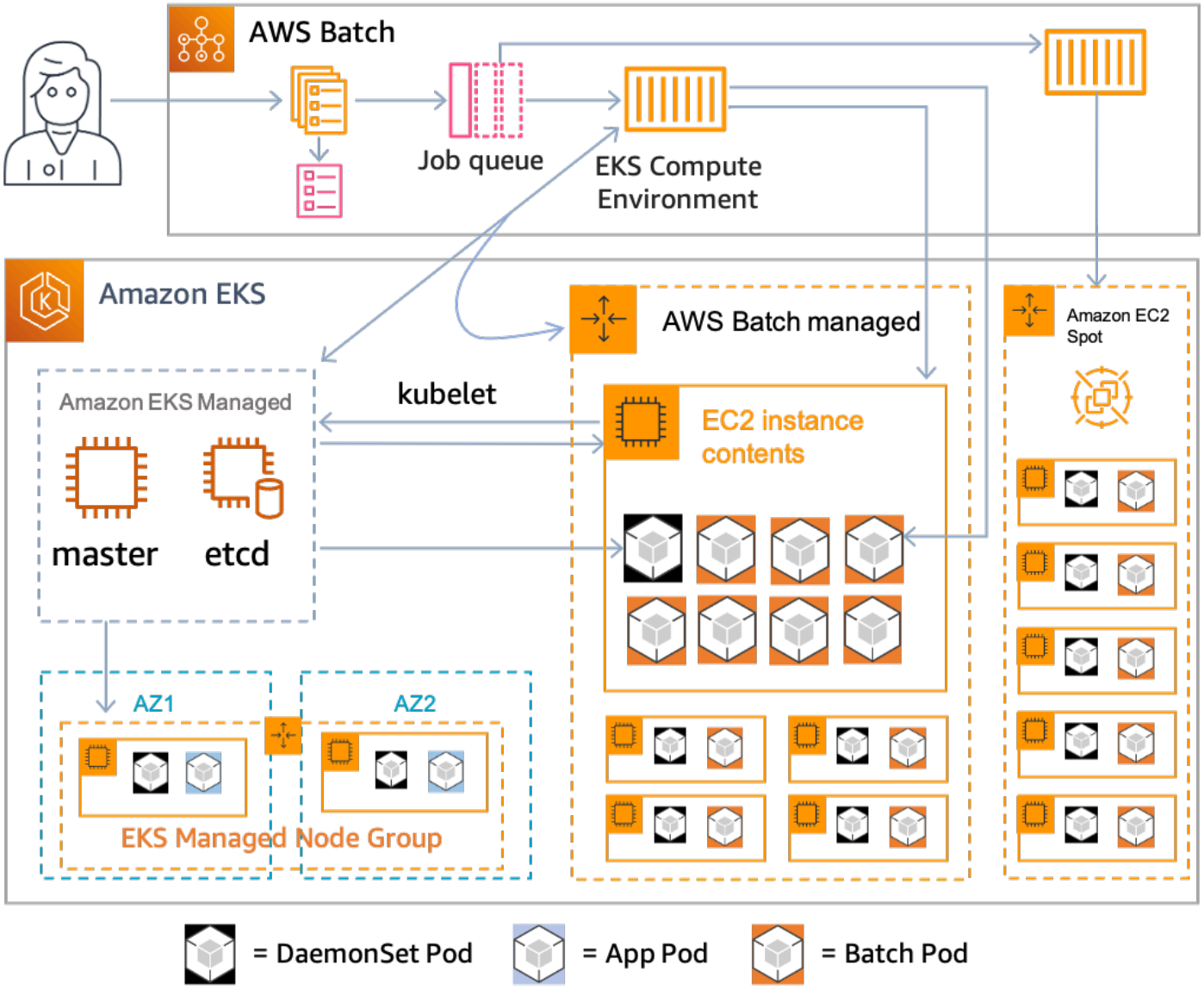
Debe ser `FARGATE` o `FARGATE_SPOT`.

```
"type": "FARGATE_SPOT"
```

AWS Batch en Amazon EKS



AWS Batch simplifica las cargas de trabajo por lotes en los clústeres de Amazon EKS al proporcionar funciones de gestión por lotes. Esto incluye la creación de colas, el seguimiento de las dependencias, la gestión de los reintentos y las prioridades de los trabajos, la gestión de los pods y el escalado de los nodos. AWS Batch puede gestionar varias zonas de disponibilidad y varios tipos y tamaños de instancias de Amazon EC2. AWS Batch integra varias de las prácticas recomendadas de Amazon EC2 Spot para ejecutar sus cargas de trabajo de forma tolerante a errores, lo que permite reducir las interrupciones. Puede utilizar AWS Batch para ejecutar un puñado de trabajos de un día para otro o millones de trabajos de misión crítica con total confianza.



AWS Batch es un servicio gestionado que organiza las cargas de trabajo por lotes en Kubernetes los clústeres gestionados por Amazon Elastic Kubernetes Service (Amazon EKS). AWS Batch lleva a cabo esta organización de forma externa a sus clústeres mediante un modelo de «superposición». Como AWS Batch se trata de un servicio gestionado, no hay Kubernetes componentes (por ejemplo, operadores o recursos personalizados) que instalar o gestionar en el clúster. AWS Batch solo necesita que el clúster esté configurado con controles de acceso basados en roles (RBAC) que permitan AWS Batch comunicarse con el servidor de API. Kubernetes AWS Batch llama a Kubernetes las API para crear, monitorear y eliminar Kubernetes pods y nodos.

AWS Batch tiene una lógica de escalado integrada para escalar Kubernetes los nodos en función de la carga de la cola de trabajos con optimizaciones en términos de asignación de la capacidad de trabajo. Cuando la cola de trabajos esté vacía, reduce la AWS Batch escala de los nodos hasta

alcanzar la capacidad mínima que haya establecido, que de forma predeterminada es cero. AWS Batch gestiona todo el ciclo de vida de estos nodos y los decora con etiquetas y manchas. De esta forma, no se colocan otras Kubernetes cargas de trabajo en los nodos gestionados por. AWS Batch La excepción son las `DaemonSets` que pueden dirigirse a AWS Batch los nodos para proporcionar supervisión y otras funciones necesarias para la correcta ejecución de las tareas. Además, AWS Batch no ejecuta trabajos, específicamente pods, en los nodos del clúster que no administra. De esta forma, puede usar una lógica de escalado y servicios independientes para otras aplicaciones del clúster.

Para enviar trabajos AWS Batch, interactúas directamente con la AWS Batch API. AWS Batch traduce los trabajos en `podspecs` y, a continuación, crea las solicitudes para colocar pods en los nodos gestionados por AWS Batch su clúster de Amazon EKS. Puede utilizar herramientas como, por ejemplo, `kubectl` para ver los nodos y los pods en ejecución. Cuando un pod ha completado su ejecución, AWS Batch elimina el pod que creó para mantener una carga menor en el Kubernetes sistema.

Puede empezar conectando un clúster de Amazon EKS válido con AWS Batch. A continuación, adjúntele una cola de AWS Batch trabajos y registre una definición de trabajo de Amazon EKS con atributos `podspec` equivalentes. Por último, envíe los trabajos mediante la operación de SubmitJobAPI que hace referencia a la definición del trabajo. Para obtener más información, consulte Cómo empezar con AWS Batch Amazon EKS.

Elastic Fabric Adapter

Un adaptador Elastic Fabric Adapter (EFA) es un dispositivo de red que acelera las aplicaciones de informática de alto rendimiento (HPC). AWS Batch es compatible con las aplicaciones que utilizan EFA si se cumplen las siguientes condiciones.

- Para obtener una lista de los tipos de instancia que admiten los EFA, consulte Tipos de instancias compatibles en la Guía del usuario para instancias de Linux de Amazon EC2.

  > **Tip**
  >
  > Para ver una lista de los tipos de instancia que admiten las EFA en un Región de AWS, ejecute el siguiente comando. Luego, cruce la lista que se muestra con la lista de tipos de instancias disponibles en la consola AWS Batch.
  >
  > ```
  > $ aws ec2 describe-instance-types  --region us-east-1  --filters Name=network-info.efa-supported,Values=true  --query "InstanceTypes[*].[InstanceType]"  --output text | sort
  > ```

- Para obtener una lista de los sistemas operativos compatibles con EFA, consulte Sistemas operativos compatibles con.
- La AMI tiene cargado el controlador de EFA.
- El grupo de seguridad del adaptador EFA debe permitir todo el tráfico de entrada y salida del propio grupo de seguridad.
- Todas las instancias que utilizan un adaptador EFA deben estar en el mismo grupo con ubicación en clúster.
- La definición del trabajo debe incluir un miembro `devices` con `hostPath` establecido en `/dev/infiniband/uverbs0` para permitir que el dispositivo EFA se transfiera a través del contenedor. Si se especifica `containerPath`, también debe establecerse en `/dev/infiniband/uverbs0`. Si se configura `permissions`, debe establecerse en `READ` | `WRITE` | `MKNOD`.

  La ubicación de los miembros LinuxParameters será diferente en los trabajos paralelos de varios nodos y los trabajos de contenedor de un solo nodo. Los ejemplos siguientes ilustran las diferencias, pero faltan algunos valores necesarios.

Example Ejemplo de un trabajo paralelo con varios nodos

```
{
  "jobDefinitionName": "EFA-MNP-JobDef",
  "type": "multinode",
  "nodeProperties": {
    ...
    "nodeRangeProperties": [
      {
        ...
        "container": {
          ...
          "linuxParameters": {
            "devices": [
              {
                "hostPath": "/dev/infiniband/uverbs0",
                "containerPath": "/dev/infiniband/uverbs0",
                "permissions": [
                    "READ", "WRITE", "MKNOD"
                ]
              },
            ],
          },
        },
      },
    ],
  },
}
```

Example Ejemplo de un trabajo de contenedor con un solo nodo

```
{
  "jobDefinitionName": "EFA-Container-JobDef",
  "type": "container",
  ...
  "containerProperties": {
    ...
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "/dev/infiniband/uverbs0",
        },
```

```
      ],
    },
  },
}
```

Para obtener más información sobre EFA, consulte Elastic Fabric Adapter en la Guía del usuario para instancias de Linux de Amazon EC2.

AWS Batch Políticas, roles y permisos de IAM

Por defecto, los usuarios no tienen permiso para crear o modificar recursos de AWS Batch ni para realizar tareas mediante la API de AWS Batch, la consola de AWS Batch o la aplicación AWS CLI. Para permitir a los usuarios trabajar con estos recursos, debe crear políticas de IAM que concedan permisos para utilizar recursos específicos y acciones de la API. A continuación, asocie esas políticas a los usuarios o grupos que necesiten esos permisos.

Cuando se adjunta una política a un usuario o grupo de usuarios, la política permite o deniega los permisos para realizar tareas específicas en recursos específicos. Para obtener más información, consulte Permisos y políticas en la Guía del usuario de IAM . Para obtener más información sobre cómo crear y administrar políticas personalizadas de IAM, consulte Administración de políticas de IAM.

AWS Batch hace llamadas a otras Servicios de AWS en su nombre. Como resultado, AWS Batch debe autenticarse con sus credenciales. Más específicamente, AWS Batch se autentica mediante la creación de una política y rol de IAM que concede dichos permisos. A continuación, asocia el rol a sus entornos informáticos al crearlos. Para obtener más información, consulte Función de instancia de Amazon ECS, Roles de IAM, Usar roles vinculados a servicios y Creación de un rol para delegar permisos a un servicio de AWS en la Guía del usuario de IAM.

Introducción

Una política de IAM debe conceder o denegar permisos para usar una o varias acciones de AWS Batch.

Temas

Estructura de la política

En los siguientes temas se explica la estructura de una política de IAM.

Temas

Sintaxis de la política

Una política de IAM es un documento JSON que contiene una o varias instrucciones. Cada instrucción tiene la estructura siguiente.

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
    "key":"value"
    }
      }
    }
  ]
}
```

Una instrucción está compuesta por varios elementos:

- Effect: el valor de effect puede ser `Allow` o `Deny`. Por defecto, los usuarios no tienen permiso para utilizar recursos y acciones de la API. De este modo, se deniegan todas las solicitudes. Si se concede un permiso explícito se anula el valor predeterminado. Una denegación explícita invalida cualquier permiso concedido.
- Acción: La acción es la acción específica de la API para la que está concediendo o denegando el permiso. Para obtener instrucciones acerca de cómo especificar la acción, consulte Acciones de AWS Batch.

- Resource: el recurso al que afecta la acción. Con algunas acciones de la API de AWS Batch, puede incluir recursos específicos en su política que pueden ser creados o modificados por la acción. Para especificar un recurso en la instrucción se utiliza el nombre de recurso de Amazon (ARN). Para obtener más información, consulte Permisos de nivel de recurso admitidos para las acciones de la API de AWS Batch y Nombres de recursos de Amazon para AWS Batch. Si la operación de la API AWS Batch, no admite actualmente permisos a nivel de recurso, incluya un comodín (*) para especificar que la acción puede afectar a todos los recursos.
- Condition: las condiciones son opcionales. Se pueden usar para controlar cuándo está en vigor la política.

Para obtener más información sobre ejemplos de declaraciones de política de IAM para AWS Batch, consulte Crear políticas de IAM AWS Batch.

Acciones de AWS Batch

En una instrucción de política de IAM, puede especificar cualquier acción de API de cualquier servicio que sea compatible con IAM. Para AWS Batch, use el prefijo siguiente con el nombre de la acción de API: `batch:` (por ejemplo, `batch:SubmitJob` y `batch:CreateComputeEnvironment`).

Para especificar varias acciones en una única sentencia, separe cada acción con una coma.

```
"Action": ["batch:action1", "batch:action2"]
```

También puede especificar varias acciones mediante un comodín (*). Por ejemplo, puede especificar todas las acciones cuyo nombre empiece por la palabra «Describe».

```
"Action": "batch:Describe*"
```

Para especificar todas las acciones de la API de AWS Batch , incluya un comodín (*).

```
"Action": "batch:*"
```

Para obtener una lista de acciones de AWS Batch, consulte Acciones en la Referencia de la API AWS Batch.

Nombres de recursos de Amazon para AWS Batch

Cada declaración de política de IAM se aplica a los recursos que especifique utilizando su Nombre de recurso de Amazon (ARN).

Un Nombre de recurso de Amazon (ARN) tiene la siguiente sintaxis general:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

El servicio (por ejemplo, `batch`).

region

La Región de AWS para el recurso (por ejemplo, `us-east-2`).

cuenta

El ID de Cuenta de AWS, sin guiones (por ejemplo, `123456789012`).

resourceType

El tipo de recurso (por ejemplo, `compute-environment`).

resourcePath

Una ruta que identifica al recurso. Puede utilizar un comodín (*) en sus rutas.

Algunas de las operaciones de la API de AWS Batch actualmente admiten los permisos de nivel de recursos. Para obtener más información, consulte Permisos de nivel de recurso admitidos para las acciones de la API de AWS Batch. Para especificar todos los recursos, o si una acción de API específica no admite ARN, incluya el comodín asterisco (*) en el elemento `Resource`.

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

Comprobar que los usuarios tienen los permisos necesarios

Antes de poner una política de IAM en producción, asegúrese de que concede a los usuarios los permisos para utilizar las acciones y recursos específicos de la API que necesitan.

Para ello, cree primero un usuario de prueba y adjúntele la política de IAM. A continuación, realice una solicitud como usuario de prueba. Puede realizar solicitudes de prueba en la consola o con la AWS CLI.

> **Note**
>
> También puede probar las políticas con el Simulador de políticas de IAM. Para obtener más información sobre el simulador de políticas, consulte Trabajar con el simulador de políticas de IAM en la Guía del usuario de IAM.

Si la política no concede los permisos previstos al usuario o es demasiado permisiva, puede ajustarla según sea necesario. Repita las pruebas hasta obtener el resultado deseado.

> **Important**
>
> Puede que los cambios en la política tarden varios minutos en propagarse y surtir efecto. Por lo tanto, le recomendamos que deje pasar al menos cinco minutos antes de probar las actualizaciones de sus políticas.

Si se produce un error en la comprobación de autorización, la solicitud devuelve un mensaje codificado con información de diagnóstico. Puede descodificar el mensaje usando la acción `DecodeAuthorizationMessage`. Para obtener más información, consulte DecodeAuthorizationMessage en la Referencia de la API de AWS Security Token Service y decode-authorization-message en la Referencia de comandos de la AWS CLI.

Permisos de nivel de recurso admitidos para las acciones de la API de AWS Batch

El término permisos de nivel de recurso se refiere a la posibilidad de especificar en qué recursos pueden realizar acciones los usuarios. AWS Batch es compatible parcialmente con permisos de nivel de recurso. En algunas acciones de AWS Batch, puede determinar cuándo se permite utilizarlas a los usuarios en función de si se cumplen una serie de condiciones. También puede controlar en función de los recursos específicos que los usuarios pueden usar. Por ejemplo, puede conceder a los usuarios permisos para enviar trabajos, pero solo a una cola de trabajos específica y únicamente con una definición de trabajo determinada.

La lista siguiente describe las acciones de la API de AWS Batch que actualmente admiten permisos de nivel de recursos. La lista también describe los recursos admitidos, los ARN de los recursos y las claves de condición de cada una de ellas.

> **Important**
>
> Si una acción de la API de AWS Batch no aparece en la tabla, significa que no admite los permisos de nivel de recursos. Si una acción de la API de AWS Batch no admite este tipo de permisos de nivel de recursos, usted puede conceder permisos a los usuarios para que la utilicen. Sin embargo, debe incluir un comodín (*) para el elemento de recurso de su declaración de política.

Acciones

CancelJob, CreateComputeEnvironment, CreateJobQueue, CreateSchedulingPolicy, DeleteComputeEnvironment, DeleteJobQueue, DeleteSchedulingPolicy, DeregisterJobDefinition, ListTagsForResource, RegisterJobDefinition, SubmitJob, TagResource, TerminateJob, UntagResource, UpdateComputeEnvironment, UpdateSchedulingPolicy, UpdateJobQueue

CancelJob

Cancela un trabajo en una cola de trabajo de AWS Batch Batch.

Recurso

Tarea

arn:aws:batch:*region*:*account*:job/*jobId*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

CreateComputeEnvironment

Crea un entorno informático de AWS Batch.

Recurso

Entorno informático

arn:aws:batch:*región*:*cuenta*:compute-environment/*nombre-de-entorno-informático*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Claves de condición

`aws:RequestTag/${TagKey}` (Cadena)

Filtra acciones basadas en las etiquetas que se pasan en la solicitud.

`aws:TagKeys` (Cadena)

Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.

CreateJobQueue

Crea una cola de AWS Batch.

Recurso

Entorno informático

arn:aws:batch:*región*:*cuenta*:compute-environment/*nombre-de-entorno-informático*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Cola de trabajos

arn:aws:batch:*region*:*account*:job-queue/*queue-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Claves de condición

`aws:RequestTag/${TagKey}` (Cadena)

Filtra acciones basadas en las etiquetas que se pasan en la solicitud.

`aws:TagKeys` (Cadena)

Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.

DeleteComputeEnvironment

Elimina un entorno informático de AWS Batch.

Recurso

Entorno informático

arn:aws:batch:*región*:*cuenta*:compute-environment/*nombre-de-entorno-informático*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

CreateSchedulingPolicy

Crea una política AWS Batch de programación.

Recurso

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Claves de condición

`aws:RequestTag/${TagKey}` (Cadena)

Filtra acciones basadas en las etiquetas que se pasan en la solicitud.

`aws:TagKeys` (Cadena)

Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.

DeleteJobQueue

Elimina la cola de trabajo especificada. Al eliminar la cola de trabajos, finalmente se eliminan todos los trabajos de la cola. Los trabajos se eliminan a una velocidad de unos 16 trabajos por segundo.

Recurso

Cola de trabajos

arn:aws:batch:*region*:*account*:job-queue/*queue-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

DeleteSchedulingPolicy

Elimina la política de programación especificada.

Recurso

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

DeregisterJobDefinition

Cancela el registro de una definición de trabajo de AWS Batch.

Recurso

Definición de trabajo

arn:aws:batch:*region*:*account*:job-definition/*definition-name*:*revision*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

ListTagsForResource

Muestra una lista de las etiquetas del recurso especificado.

Recurso

Entorno informático

arn:aws:batch:*región*:*cuenta*:compute-environment/*nombre-de-entorno-informático*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Tarea

arn:aws:batch:*region*:*account*:job/*jobId*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Definición de trabajo

arn:aws:batch:*region*:*account*:job-definition/*definition-name*:*revision*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Cola de trabajos

arn:aws:batch:*region*:*account*:job-queue/*queue-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

RegisterJobDefinition

Registra una definición AWS Batch.

Recurso

Definición de trabajo

arn:aws:batch:*region*:*account*:job-definition/*definition-name*

Claves de condición

`batch:AWSLogsCreateGroup` (Booleano)

Cuando este parámetro sea verdadero, se creará `awslogs-group` para los registros.

`batch:AWSLogsGroup` (Cadena)

El grupo de `awslogs` donde se encuentran los registros.

`batch:AWSLogsRegion` (Cadena)

La región a la que se envían los registros.

`batch:AWSLogsStreamPrefix` (Cadena)

Prefijo de flujo de registros `awslogs`.

`batch:Image` (Cadena)

La imagen de Docker que se utiliza para iniciar un trabajo.

`batch:LogDriver` (Cadena)

El controlador de registro utilizado para el trabajo.

`batch:Privileged` (Booleano)

Cuando este parámetro es verdadero, al contenedor para el trabajo se le conceden permisos elevados en la instancia de contenedor de host.

`batch:User` (Cadena)

El nombre de usuario o UID numérico que utilizar dentro del contenedor para el trabajo.

`aws:RequestTag/${TagKey}` (Cadena)

Filtra acciones basadas en las etiquetas que se pasan en la solicitud.

`aws:TagKeys` (Cadena)

Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.

SubmitJob

Envía un trabajo de AWS Batch desde una definición de trabajo.

Recurso

Tarea

arn:aws:batch:*region*:*account*:job/*jobId*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Definición de trabajo

arn:aws:batch:*region*:*account*:job-definition/*definition-name*[:*revision*]

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

> **Note**
>
> Esta clave solo se puede utilizar cuando la definición del trabajo Nombre de recurso de Amazon (ARN) tiene el formato `arn:aws:batch:`*region*`:`*account_number*`:job-definition/`*definition-name*`:`*revision*`.`

Cola de trabajos

arn:aws:batch:*region*:*account*:job-queue/*queue-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

TagResource

Etiqueta el recurso especificado.

Recurso

Entorno informático

arn:aws:batch:*región*:*cuenta*:compute-environment/*nombre-de-entorno-informático*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Tarea

arn:aws:batch:*region*:*account*:job/*jobId*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Definición de trabajo

arn:aws:batch:*region*:*account*:job-definition/*definition-name*:*revision*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Cola de trabajos

arn:aws:batch:*region*:*account*:job-queue/*queue-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Claves de condición

`aws:RequestTag/${TagKey}` (Cadena)

Filtra acciones basadas en las etiquetas que se pasan en la solicitud.

`aws:TagKeys` (Cadena)

Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.

TerminateJob

Termina un trabajo en una cola de trabajo de AWS Batch.

Recurso

Tarea

arn:aws:batch:*region*:*account*:job/*jobId*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

UntagResource

Quita la etiqueta del recurso especificado.

Recurso

Entorno informático

arn:aws:batch:*región*:*cuenta*:compute-environment/*nombre-de-entorno-informático*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Tarea

arn:aws:batch:*region*:*account*:job/*jobId*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Definición de trabajo

arn:aws:batch:*region*:*account*:job-definition/*definition-name*:*revision*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Cola de trabajos

arn:aws:batch:*region*:*account*:job-queue/*queue-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Claves de condición

`aws:TagKeys` (Cadena)

Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.

UpdateComputeEnvironment

Un entorno informático de AWS Batch.

Recurso

Entorno informático

arn:aws:batch:*región*:*cuenta*:compute-environment/*nombre-de-entorno-informático*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

UpdateJobQueue

Actualiza una cola de trabajo.

Recurso

Cola de trabajos

arn:aws:batch:*region*:*account*:job-queue/*queue-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

UpdateSchedulingPolicy

Actualiza una política de programación.

Recurso

Política de programación

arn:aws:batch:*region*:*account*:scheduling-policy/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Claves de condición para acciones de API de AWS Batch

AWS Batch define las siguientes claves de condición que se pueden utilizar en el elemento `Condition` de una política de IAM. Puede utilizar estas claves para ajustar las condiciones en las que se aplica la instrucción de política. Para ver las claves de condición globales que están disponibles para todos los servicios, consulte Claves de condición globales disponibles en la Guía del usuario de IAM.

`batch:AWSLogsCreateGroup` (Booleano)

Cuando este parámetro sea verdadero, se creará `awslogs-group` para los registros.

`batch:AWSLogsGroup` (Cadena)

El grupo de `awslogs` donde se encuentran los registros.

`batch:AWSLogsRegion` (Cadena)

La Región de AWS a la que se envían los registros.

`batch:AWSLogsStreamPrefix` (Cadena)

Prefijo de flujo de registros `awslogs`.

`batch:Image` (Cadena)

La imagen de Docker que se utiliza para iniciar un trabajo.

`batch:LogDriver` (Cadena)

El controlador de registro utilizado para el trabajo.

`batch:Privileged` (Booleano)

Cuando este parámetro es verdadero, al contenedor se le conceden permisos elevados en la instancia de contenedor de host (similares a los de un usuario raíz).

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

`aws:RequestTag/${TagKey}` (Cadena)

Filtra acciones basadas en las etiquetas que se pasan en la solicitud.

`batch:ShareIdentifier` (Cadena)

Filtra las acciones en función del parámetro `shareIdentifier` enviado a SubmitJob.

`aws:TagKeys` (Cadena)

Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.

`batch:User` (Cadena)

El nombre de usuario o ID de usuario numérico (UID) que utilizar dentro del contenedor.

Ejemplos de políticas

Los siguientes ejemplos muestran instrucciones de política que puede utilizar para controlar los permisos que los usuarios de tienen en AWS Batch.

Ejemplos

- Acceso de solo lectura
- Restricción al envío de un trabajo según el usuario de POSIX, la imagen de Docker el nivel de privilegios y el rol en el envío del trabajo
- Restricción del envío de un trabajo según el prefijo de la definición de trabajo
- Restringir a la cola de trabajos
- Denegar la acción cuando todas las claves de condición coincidan con las cadenas
- Deniegue la acción cuando alguna clave de condición coincida con una cadena
- Utilice la clave de condición batch:ShareIdentifier

Acceso de solo lectura

La siguiente política concede a los usuarios permisos para usar todas las acciones de la API de AWS Batch con un nombre que empiece por `Describe` y `List`.

A menos que otra instrucción les otorgue permiso para hacerlo, los usuarios no tienen permiso para llevar a cabo ninguna acción en los recursos. De forma predeterminada, se les niega el permiso para usar las acciones de la API.

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

```
                "batch:Describe*",
                "batch:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Restricción al envío de un trabajo según el usuario de POSIX, la imagen de Docker el nivel de privilegios y el rol en el envío del trabajo

La siguiente política permite a un usuario POSIX administrar su propio conjunto de definiciones de trabajo restringidas.

Utilice las instrucciones primera y segunda para registrar y anular el registro de cualquier definición de trabajo cuyo nombre tenga el prefijo *JobDefA_*.

La primera instrucción también utiliza claves de contexto de condición para restringir los valores de usuario de POSIX, estado de privilegio e imagen de contenedor en las `containerProperties` de una definición de trabajo. Para obtener más información, consulte RegisterJobDefinition en la AWS BatchReferencia de la API. En este ejemplo, las definiciones de trabajo solo se pueden registrar cuando el usuario POSIX está configurado en `nobody`. El indicador privilegiado está establecido en `false`. Por ejemplo, el siguiente comando muestra las herramientas `myImage` de un repositorio de Amazon ECR.

> **Important**
>
> Docker resuelve el parámetro `user` de ese usuario`uid` desde la imagen del contenedor. En la mayoría de los casos, se encuentra en el archivo `/etc/passwd` de la imagen del contenedor. Esta resolución de nombres se puede evitar si usan valores de `uid` directos tanto en la definición de trabajo como en las políticas de IAM asociadas. Tanto las operaciones de API de AWS Batch como las claves condicionales de IAM `batch:User` admiten valores numéricos.

Usar la tercera instrucción para restringir únicamente un rol específico a una definición de trabajo.

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

```
        {
            "Effect": "Allow",
            "Action": [
                "batch:RegisterJobDefinition"
            ],
            "Resource": [
                "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*"
            ],
            "Condition": {
                "StringEquals": {
                    "batch:User": [
                        "nobody"
                    ],
                    "batch:Image": [
                        "<aws_account_id>.dkr.ecr.<aws_region>.amazonaws.com/myImage"
                    ]
                },
                "Bool": {
                    "batch:Privileged": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "batch:DeregisterJobDefinition"
            ],
            "Resource": [
                "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::<aws_account_id>:role/MyBatchJobRole"
            ]
        }
    ]
}
```

Restricción del envío de un trabajo según el prefijo de la definición de trabajo

Usar siguiente política para enviar trabajos a cualquier cola de trabajos con cualquier nombre de definición de trabajo que comience por *JobDefA_*.

> **Important**
>
> Al determinar el ámbito del acceso en el nivel de recursos para el envío de trabajos, debe proporcionar los tipos de recursos de cola de trabajos y de definición de trabajo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/*"
      ]
    }
  ]
}
```

Restringir a la cola de trabajos

Usar la siguiente política para enviar trabajos a una cola de trabajos denominada queue1 con cualquier nombre de definición de trabajo.

> **Important**
>
> Al determinar el ámbito del acceso en el nivel de recursos para el envío de trabajos, debe proporcionar los tipos de recursos de cola de trabajos y de definición de trabajo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/queue1"
      ]
    }
  ]
}
```

Denegar la acción cuando todas las claves de condición coincidan con las cadenas

La siguiente política deniega el acceso a la RegisterJobDefinitionoperación de la API cuando la clave de condición `batch:Image` (ID de imagen del contenedor) es «*string1*» y la clave de condición `batch:LogDriver` (controlador de registro del contenedor) es «*string2*». AWS Batch evalúa las claves de condición de cada contenedor. Cuando un trabajo abarca varios contenedores, como un trabajo paralelo de varios nodos, es posible que los contenedores tengan configuraciones diferentes. Si se evalúan varias claves de condición en una instrucción, se combinan mediante la lógica AND. Por lo tanto, si alguna de las múltiples claves de condición no coincide con un contenedor, el efecto Deny no se aplica a ese contenedor. Por el contrario, es posible que se deniegue un contenedor diferente en el mismo trabajo.

Para ver una lista de las claves de condición para AWS Batch, consulte Claves de condición para AWS Batch en la Referencia de autorización de servicio. A excepción de`batch:ShareIdentifier`, todas las claves de condición `batch` se pueden usar de esta manera. La clave de condición `batch:ShareIdentifier` se define para un trabajo, no para una definición de trabajo.

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

```
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": "batch:RegisterJobDefinition",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "batch:Image": "string1",
          "batch:LogDriver": "string2"
        }
      }
    }
  ]
}
```

Deniegue la acción cuando alguna clave de condición coincida con una cadena

La siguiente política deniega el acceso a la RegisterJobDefinitionoperación de la API cuando la clave de condición `batch:Image` (ID de imagen del contenedor) es «*string1*» o la clave de condición `batch:LogDriver` (controlador de registro del contenedor) es «*string2*». Cuando un trabajo abarca varios contenedores, como un trabajo paralelo de varios nodos, es posible que los contenedores tengan configuraciones diferentes. Si se evalúan varias claves de condición en una instrucción, se combinan mediante la lógica `AND`. Por lo tanto, si alguna de las múltiples claves de condición no coincide con un contenedor, el efecto `Deny` no se aplica a ese contenedor. Por el contrario, es posible que se deniegue un contenedor diferente en el mismo trabajo.

Para ver una lista de las claves de condición para AWS Batch, consulte Claves de condición para AWS Batch en la Referencia de autorización de servicio. Excepto para `batch:ShareIdentifier`, todas las claves de condición `batch` se pueden usar de esta manera. (La clave de condición `batch:ShareIdentifier` se define para un trabajo, no para una definición de trabajo).

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

```
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:Image": [
            "string1"
          ]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:LogDriver": [
            "string2"
          ]
        }
      }
    }
  ]
}
```

Utilice la clave de condición `batch:ShareIdentifier`

Utilice la siguiente política para enviar los trabajos que utilizan la definición de trabajo `jobDefA` con la cola de trabajos `jobqueue1` con el identificador compartido `lowCpu`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-definition/JobDefA",
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-queue/jobqueue1"
      ],
      "Condition": {
        "StringEquals": {
          "batch:ShareIdentifier": [
            "lowCpu"
          ]
        }
      }
    }
  ]
}
```

Política administrada de AWS Batch

AWS Batch ofrece una política administrada que se puede asociar a los usuarios para concederles permisos de utilización de recursos de AWS Batch y operaciones de la API. Puede aplicar esta política directamente o utilizarla como punto de partida para crear sus propias políticas. Para obtener más información sobre cada operación de la API mencionada en estas políticas, consulte Acciones en la AWS BatchReferencia de la API.

AWSBatchFullAccess

Esta política proporciona acceso pleno a AWS Batch como administrador.

```
{
```

```
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource":"*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:PassRole"
      ],
      "Resource":[
        "arn:aws:iam::*:role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/ecsInstanceRole",
        "arn:aws:iam::*:instance-profile/ecsInstanceRole",
        "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/AWSBatchJobRole*"
      ]
    },
    {
```

```
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"arn:aws:iam::*:role/*Batch*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "batch.amazonaws.com"
          }
      }
    }
  ]
}
```

Crear políticas de IAM AWS Batch

Puede crear políticas de IAM específicas para restringir las llamadas y los recursos a los que tienen acceso los usuarios de su cuenta. A continuación, puede asignar esas políticas a los usuarios.

Cuando se adjunta una política a un usuario o grupo de usuarios, esta permite o deniega a los usuarios el permiso para tareas específicas en recursos específicos. Para obtener más información, consulte Permisos y políticas en la Guía del usuario de IAM . Para obtener instrucciones sobre cómo gestionar y crear políticas de IAM personalizadas, consulte Gestión de políticas de IAM.

Función de instancia de Amazon ECS

AWS Batch los entornos informáticos se rellenan con instancias de contenedor de Amazon ECS. Ejecutan el agente de contenedor de Amazon ECS de forma local. El agente de contenedor de Amazon ECS realiza llamadas a diversas operaciones de la API de AWS en su nombre. Por lo tanto, las instancias de contenedor que ejecutan el agente requieren una política de IAM y un rol para que estos servicios reconozcan que el agente le pertenece. Debe crear un rol de IAM y un perfil de instancia para que las instancias de contenedor los usen cuando se lancen. De lo contrario, no puede crear un entorno informático y lanzar instancias de contenedor en él. Este requisito se aplica a las instancias de contenedor lanzadas con o sin la AMI optimizada de Amazon ECS proporcionada por Amazon. Para obtener más información, consulte Rol de IAM de instancia de contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

El perfil de instancia y el rol de instancia de Amazon ECS se crean automáticamente en la experiencia de primer uso de la consola. Sin embargo, puede seguir estos pasos para verificar si la

cuenta ya dispone del rol y el perfil de instancia de Amazon ECS. En los siguientes pasos también se explica cómo adjuntar la política de IAM administrada.

Para verificar el **`ecsInstanceRole`** en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles (Roles) en el panel de navegación.
3. En la lista de roles, busque `ecsInstanceRole`. Si el rol no existe, siga los siguientes pasos para crearlo.
   a. Elija Create Role (Crear rol).
   b. En Trusted entity type (Tipo de entidad de confianza), elija Servicio de AWS.
   c. En Common use cases, elija EC2.
   d. Elija Next (Siguiente).
   e. Para ver Políticas de permisos, busque AmazonEC2ContainerServiceforEC2Role.
   f. Seleccione la casilla situada junto a AmazonEC2ContainerServiceForec2Role y, a continuación, seleccione Siguiente.
   g. En Role Name (Nombre de rol), escriba `ecsInstanceRole` y elija Create role (Crear rol).

> **Note**
>
> Si utiliza la AWS Management Console para crear un rol para Amazon EC2, la consola crea un perfil de instancias y le da el mismo nombre que al rol.

También puede utilizar la AWS CLI para crear un rol de IAM `ecsInstanceRole`. El siguiente ejemplo crea un rol de IAM con las siguientes políticas de confianza y una política administrada de AWS.

Para crear un rol de IAM y un perfil de instancias (AWS CLI)

1. Cree la siguiente política de confianza y guárdela en un archivo de texto que se denomina `ecsInstanceRole-role-trust-policy.json`.

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

```
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Utilice el comando create-role para crear el rol de servicio `ecsInstanceRole`. Especifique la ubicación del archivo de política de confianza en el parámetro `assume-role-policy-document`.

```
$ aws iam create-role \
    --role-name ecsInstanceRole \
    --assume-role-policy-document file://ecsInstanceRole-role-trust-policy.json
```

A continuación, se muestra un ejemplo de respuesta.

```
{
    "Role": {
         "Path": "/",
         "RoleName: "ecsInstanceRole",
         "RoleId": "AROAT46P5RDIY4EXAMPLE",
         "Arn": "arn:aws:iam::123456789012:role/ecsInstanceRole".
        "CreateDate": "2022-12-12T23:46:37.247Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service: "ec2.amazonaws.com"
                    }
                    "Action": "sts:AssumeRole",
                }
            ]
          }
        }
```

3. Utilice el comando create-instance-profile para crear un perfil de instancia llamado `ecsInstanceRole`.

> **Note**
>
> Debe crear roles y perfiles de instancias como acciones independientes en la API AWS CLI y la AWS.

```
$ aws iam create-instance-profile --instance-profile-name ecsInstanceRole
```

A continuación, se muestra un ejemplo de respuesta.

```
{
    "InstanceProfile": {
        "Path": "/",
        "InstanceProfileName": "ecsInstanceRole",
        "InstanceProfileId": "AIPAT46P5RDITREXAMPLE",
        "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
        "CreateDate": "2022-06-30T23:53:34.093Z",
        "Roles": [],    }
}
```

4. Utilice el comando add-role-to-instance-profile para añadir el rol `ecsInstanceRole` al perfil de instancia de `ecsInstanceRole`.

```
aws iam add-role-to-instance-profile \
    --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole
```

5. Use el comando attach-role-policy para adjuntar la política `AmazonEC2ContainerServiceforEC2Role` AWS administrada al rol `ecsInstanceRole`.

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
    --role-name ecsInstanceRole
```

Rol de flota de spot de Amazon EC2

Si crea un entorno informático administrado que utiliza las instancias de flota de spot Amazon EC2, debe crear la política `AmazonEC2SpotFleetTaggingRole`. Esta política concede a la flota de spot

permiso para lanzar, etiquetar y finalizar instancias en su nombre. Especifique el rol en la solicitud de flota de spot. También debe tener los roles vinculados al servicio AWSServiceRoleForEC2Spot y AWSServiceRoleForEC2SpotFleet para Amazon EC2 Spot y flota de spot. Utilice la siguiente instrucción para crear todos estos roles. Para obtener más información, consulte Usar roles vinculados a servicios y Creación de un rol para delegar permisos a un servicio de AWS en la Guía del usuario de IAM.

Temas

- Cómo crear roles de flota de spot de Amazon EC2 en AWS Management Console
- Cree roles de flota de Spot Amazon EC2 con la AWS CLI

Cómo crear roles de flota de spot de Amazon EC2 en AWS Management Console

Para crear el rol vinculado a servicio de IAM denominado **`AmazonEC2SpotFleetTaggingRole`** para la flota de spot de Amazon EC2

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En Gestión de acceso, seleccione Roles,
3. En Roles, seleccione Crear rol.
4. En Seleccione una entidad de confianza para Tipo de entidad de confianza, elija Servicio de AWS.
5. Para Casos de uso para otros Servicios de AWS, elija EC2 y, a continuación, elija EC2 - Etiquetado de flota de spot.
6. Elija Next (Siguiente).
7. En Políticas de permisos para el Nombre de la política, compruebe `AmazonEC2SpotFleetTaggingRole`.
8. Elija Next (Siguiente).
9. Para Nombrar, revisar y crear:
    a. En Nombre de rol, escriba un nombre para identificar el rol.
    b. En Descripción, introduzca una breve explicación de la política.
    c. (Opcional) En Paso 1: seleccione entidades de confianza, elija Editar para modificar el código.

d. (Opcional) En el Paso 2: agregar permisos, seleccione Editar para modificar el código.

e. (Opcional) En Agregar etiquetas, elija Agregar etiqueta para agregar etiquetas al recurso.

f. Elija Create role (Crear rol).

> **Note**
>
> En el pasado, había dos políticas administradas para el rol de la flota de spot de Amazon EC2.
>
> - AmazonEC2SpotFleetRole: esta era la política administrada original para el rol de flota de spot. Sin embargo, ya no se recomienda usarlo con AWS Batch. Esta política no admite el etiquetado de instancias de spot en entornos informáticos, que es obligatorio para utilizar la función vinculada al servicio `AWSServiceRoleForBatch`. Si ya había creado un rol de flota de spot con esta política, aplique la nueva política recomendada a ese rol. Para obtener más información, consulte Instancias de spot no etiquetadas en el momento de su creación.
> - AmazonEC2SpotFleetTaggingRole: este rol proporciona todos los permisos necesarios para etiquetar instancias de spot de Amazon EC2. Utilice este rol para permitir el etiquetado de instancias de spot en sus entornos informáticos de AWS Batch.

Cree roles de flota de Spot Amazon EC2 con la AWS CLI

Para crear el rol de IAM AmazonEC2SpotFleetTaggingRole para sus entornos informáticos de flota de spot

1. Ejecute el siguiente comando con la AWS CLI.

```
$ aws iam create-role --role-name AmazonEC2SpotFleetTaggingRole \
    --assume-role-policy-document '{
    "Version":"2012-10-17",
    "Statement":[
       {
         "Sid":"",
         "Effect":"Allow",
         "Principal": {
           "Service":"spotfleet.amazonaws.com"
          },
```

```
            "Action":"sts:AssumeRole"
          }
        ]
      }'
```

2. Para adjuntar la política de IAM administrada AmazonEC2SpotFleetTaggingRole a su rol AmazonEC2SpotFleetTaggingRole, ejecute el siguiente comando con el AWS CLI.

```
$ aws iam attach-role-policy \
  --policy-arn \
    arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
    AmazonEC2SpotFleetTaggingRole
```

Para crear el rol vinculado a servicio de IAM denominado **`AWSServiceRoleForEC2Spot`** para la flota de spot de Amazon EC2

> **Note**
>
> Si el rol vinculado al servicio de IAM denominado `AWSServiceRoleForEC2Spot` ya existe, aparecerá un mensaje de error similar al siguiente.
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
>  operation:
> Service role name AWSServiceRoleForEC2Spot has been taken in this account,
>  please try a different suffix.
> ```

- Ejecute el siguiente comando con la AWS CLI.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Para crear el rol vinculado a servicio de IAM denominado **`AWSServiceRoleForEC2SpotFleet`** para la flota de spot de Amazon EC2

> **Note**
>
> Si el rol vinculado al servicio de IAM denominado `AWSServiceRoleForEC2SpotFleet` ya existe, aparecerá un mensaje de error similar al siguiente.
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
>  operation:
> Service role name AWSServiceRoleForEC2SpotFleet has been taken in this account,
>  please try a different suffix.
> ```

- Ejecute el siguiente comando con la AWS CLI.

  ```
  $ aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
  ```

Rol de IAM de EventBridge

Amazon EventBridge proporciona un flujo casi en tiempo real de eventos del sistema que describen los cambios en los recursos de AWS. Los trabajos de AWS Batch están disponibles como destinos de EventBridge. Mediante reglas sencillas que puede configurar rápidamente, puede asignar eventos y enviar trabajos de AWS Batch como respuesta a ellos. Para poder enviar trabajos de AWS Batch con reglas y destinos de EventBridge, EventBridge debe tener permisos para ejecutar trabajos de AWS Batch en su nombre.

> **Note**
>
> Cuando se crea una regla en la consola EventBridge que especifica una cola AWS Batch como destino, se puede crear este rol. Para ver un tutorial de ejemplo, consulte AWS Batch puestos de trabajo como EventBridge objetivos. Puede crear un rol de EventBridge manualmente mediante una consola de IAM. Para obtener instrucciones, consulte Creación de un rol mediante políticas de confianza personalizadas (consola) en la Guía del usuario de IAM.

La relación de confianza para su rol de IAM de EventBridge debe proporcionar al principal del servicio `events.amazonaws.com` la capacidad de asumir el rol.

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

Asegúrese de que la política asociada a su rol de IAM en EventBridge permita `batch:SubmitJob` permisos en sus recursos. En el siguiente ejemplo, AWS Batch proporciona la política administrada de `AWSBatchServiceEventTargetRole` para conceder estos permisos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
       ],
      "Resource": "*"
    }
  ]
}
```

AWS Batch Event Stream para Amazon EventBridge

Puedes usar la transmisión de AWS Batch eventos de Amazon EventBridge para recibir notificaciones casi en tiempo real sobre el estado actual de los trabajos en tus colas de trabajos.

Puedes utilizarla EventBridge para obtener más información sobre tu AWS Batch servicio. Más específicamente, puede usarlo para comprobar el progreso de los trabajos, crear flujos de trabajo AWS Batch personalizados, generar informes o métricas de uso o crear sus propios paneles. Con AWS Batch y EventBridge, no necesita un código de programación y monitoreo que sondee continuamente los cambios en AWS Batch el estado de los trabajos. En su lugar, puedes gestionar los cambios de estado de las AWS Batch tareas de forma asíncrona mediante una variedad de objetivos de Amazon. EventBridge Estos incluyen AWS Lambda Amazon Simple Queue Service, Amazon Simple Notification Service o Amazon Kinesis Data Streams.

Se garantiza que los AWS Batch eventos de la transmisión de eventos se entreguen al menos una vez. En caso de que se envíen eventos duplicados, el evento ofrece información suficiente para identificar duplicados. De esta forma, puede comparar la marca horaria del evento y el estado del trabajo.

AWS Batch los puestos de trabajo están disponibles como EventBridge objetivos. Con reglas sencillas, puede hacer coincidir los eventos y enviar AWS Batch trabajos en respuesta a ellos. Para obtener más información, consulte ¿Qué es EventBridge? en la Guía del EventBridge usuario de Amazon. También puedes utilizar EventBridge para programar acciones automatizadas que se activen automáticamente en determinados momentos utilizando cron o puntuando expresiones. Para obtener más información, consulta Cómo crear una EventBridge regla de Amazon que se ejecute según un cronograma en la Guía del EventBridge usuario de Amazon. Para ver un tutorial de ejemplo, consulte AWS Batch puestos de trabajo como EventBridge objetivos. Para obtener información sobre el uso del EventBridge Programador, consulte Configuración de Amazon EventBridge Scheduler en la Guía EventBridge del usuario de Amazon.

Temas

- AWS Batch Eventos
- Uso de las notificaciones de AWS usuario con AWS Batch
- AWS Batch puestos de trabajo como EventBridge objetivos
- Tutorial: Listening for AWS Batch EventBridge
- Tutorial: envío de alertas de Amazon Simple Notification Service Alerts para eventos de trabajos fallidos

AWS Batch Eventos

AWS Batch envía los eventos de cambio de estado del trabajo a. EventBridge AWS Batch rastrea el estado de sus trabajos. Si el estado de un trabajo enviado anteriormente cambia, se invoca un evento. Por ejemplo, si un trabajo en el estado `RUNNING` pasa a al estado `FAILED`. Estos eventos se clasifican como eventos de cambio en el estado de los trabajos.

> **Note**
>
> AWS Batch podría añadir otros tipos de eventos, fuentes y detalles en el futuro. Si va a deserializar datos JSON de eventos mediante programación, asegúrese de que la aplicación esté preparada para tratar propiedades desconocidas. Esto es para evitar problemas si se agregan estas propiedades adicionales y cuando se agregan.

Eventos de cambio de estado de los trabajos

Cada vez que un trabajo existente (enviado previamente) cambia de estado, se crea un evento. Para obtener más información sobre los estados de los AWS Batch puestos, consulteEstados de trabajo.

> **Note**
>
> No se crean eventos para el envío inicial de los trabajos.

Example Evento de cambio de estado de los trabajos

Los eventos de cambio de estado de trabajo se entregan en el siguiente formato. La `detail` sección se parece al JobDetailobjeto devuelto por una operación de DescribeJobsAPI en la Referencia de AWS Batch API. Para obtener más información sobre EventBridge los parámetros, consulte Eventos y patrones de eventos en la Guía del EventBridge usuario de Amazon.

```
{
    "version": "0",
    "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
    "detail-type": "Batch Job State Change",
    "source": "aws.batch",
    "account": "123456789012",
    "time": "2022-01-11T23:36:40Z",
    "region": "us-east-1",
```

```
    "resources": [
        "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    ],
    "detail": {
        "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "jobName": "event-test",
        "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
        "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
        "status": "RUNNABLE",
        "attempts": [],
        "createdAt": 1641944200058,
        "retryStrategy": {
            "attempts": 2,
            "evaluateOnExit": []
        },
        "dependsOn": [],
        "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
        "parameters": {},
        "container": {
            "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
            "command": [
                "sleep",
                "600"
            ],
            "volumes": [],
            "environment": [],
            "mountPoints": [],
            "ulimits": [],
            "networkInterfaces": [],
            "resourceRequirements": [
                {
                    "value": "2",
                    "type": "VCPU"
                }, {
                    "value": "256",
                    "type": "MEMORY"
                }
            ],
            "secrets": []
        },
        "tags": {
```

```
            "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
        },
        "propagateTags": false,
        "platformCapabilities": []
    }
}
```

Eventos bloqueados en la cola de trabajos

Cada vez que se AWS Batch detecta un trabajo en el `RUNNABLE` estado y, por lo tanto, se bloquea una cola, se crea un evento en Amazon CloudWatch Events. Para obtener más información sobre las causas admitidas de colas de trabajo bloqueadas, consulte ejemplos de mensajes de colas de trabajos bloqueadas. El mismo motivo también está disponible en el `statusReason` campo de la acción de la `DescribeJobs` API.

Example Evento de cambio de estado de los trabajos

Los eventos de cambio de estado de trabajo se entregan en el siguiente formato. La `detail` sección se parece al JobDetailobjeto devuelto por una operación de DescribeJobsAPI en la Referencia de AWS Batch API. Para obtener más información sobre EventBridge los parámetros, consulte Eventos y patrones de eventos en la Guía del EventBridge usuario de Amazon.

```
{
    "version": "0",
    "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
    "detail-type": "Batch Job Queue Blocked",
    "source": "aws.batch",
    "account": "123456789012",
    "time": "2022-01-11T23:36:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue"
    ],
    "detail": {
        "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "jobName": "event-test",
        "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
```

```
        "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
        "status": "RUNNABLE",
        "statusReason": "blocked-reason"
        "attempts": [],
        "createdAt": 1641944200058,
        "retryStrategy": {
            "attempts": 2,
            "evaluateOnExit": []
        },
        "dependsOn": [],
        "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
        "parameters": {},
        "container": {
            "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
            "command": [
                "sleep",
                "600"
            ],
            "volumes": [],
            "environment": [],
            "mountPoints": [],
            "ulimits": [],
            "networkInterfaces": [],
            "resourceRequirements": [
                {
                    "value": "2",
                    "type": "VCPU"
                }, {
                    "value": "256",
                    "type": "MEMORY"
                }
            ],
            "secrets": []
        },
        "tags": {
            "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
        },
        "propagateTags": false,
        "platformCapabilities": []
    }
```

```
}
```

Uso de las notificaciones de AWS usuario con AWS Batch

Puede usar notificaciones de usuario de AWS para configurar los canales de entrega a fin de recibir notificaciones sobre los eventos de AWS Batch . Recibirá una notificación cuando un evento coincida con una regla que especifique. Puede recibir notificaciones de eventos a través de varios canales, como correo electrónico, notificaciones por chat de AWS Chatbot o notificaciones de inserción de AWS Console Mobile Application. También puede ver las notificaciones en el Centro de notificaciones de la consola. Las notificaciones de usuario admiten la agregación, lo que puede reducir el número de notificaciones que recibe durante eventos específicos.

Para configurar las notificaciones de usuario en AWS Batch:

1. Abra la consola de AWS CloudFormation.
2. Elija Dashsboard (Panel).
3. Seleccione Configurar notificaciones.
4. En Notificaciones de AWS usuario, elija Crear configuración de notificaciones.

Para obtener más información sobre cómo configurar y ver las notificaciones de usuario, consulte Introducción a las notificaciones AWS de usuario.

AWS Batch puestos de trabajo como EventBridge objetivos

Amazon EventBridge ofrece una transmisión casi en tiempo real de los eventos del sistema que describen los cambios en los recursos de Amazon Web Services. Normalmente, AWS Batch en Amazon Elastic Container Service, Amazon Elastic Kubernetes Service AWS y Fargate, los trabajos están disponibles como destinos. EventBridge Con reglas sencillas, puede hacer coincidir los eventos y enviar AWS Batch trabajos en respuesta a ellos. Para obtener más información, consulte ¿Qué es EventBridge? en la Guía del EventBridge usuario de Amazon.

También se pueden utilizar EventBridge para programar acciones automatizadas que se invoquen en determinados momentos utilizando cron o puntuando expresiones. Para obtener más información, consulta Cómo crear una EventBridge regla de Amazon que se ejecute según un cronograma en la Guía del EventBridge usuario de Amazon.

Para obtener información sobre cómo crear una regla que se ejecute cuando un evento coincida con un patrón de eventos, consulta Cómo crear EventBridge reglas de Amazon que reaccionen a los eventos en la Guía del EventBridge usuario de Amazon.

Los casos de uso más comunes de los AWS Batch trabajos como EventBridge objetivo incluyen los siguientes casos de uso:

- Un trabajo programado se produce a intervalos de tiempo regulares. Por ejemplo, un trabajo de cron solo se realiza durante las horas de bajo uso, cuando las instancias puntuales de spot de Amazon EC2 son menos costosas.
- Un AWS Batch trabajo se ejecuta en respuesta a una operación de API en la que se ha iniciado sesión CloudTrail. Por ejemplo, se envía un trabajo cada vez que se carga un objeto en un bucket de Amazon S3 específico. Cada vez que esto ocurre, el transformador EventBridge de entrada pasa el depósito y el nombre clave del objeto a AWS Batch los parámetros.

> **Note**
>
> En este escenario, todos los AWS recursos relacionados deben estar en la misma región. Esto incluye recursos como el bucket, la EventBridge regla y los CloudTrail registros de Amazon S3.

Para poder enviar AWS Batch trabajos con EventBridge reglas y objetivos, el EventBridge servicio necesita varios permisos para AWS Batch ejecutarlos. Al crear una regla en la EventBridge consola que especifica un AWS Batch trabajo como destino, también puede crear este rol. Para obtener más información sobre la entidad principal de servicio y los permisos de IAM necesarios para este rol, consulte Rol de IAM de EventBridge.

Crear un AWS Batch trabajo programado

El siguiente procedimiento explica cómo crear un AWS Batch trabajo programado y el rol de EventBridge IAM requerido.

Para crear un AWS Batch trabajo programado con EventBridge

> **Note**
>
> Este procedimiento funciona para todos los trabajos AWS Batch de Amazon ECS, Amazon EKS y AWS Fargate.

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En la barra de navegación, selecciona la Región de AWS que deseas usar.
3. En el panel de navegación, seleccione Reglas.
4. Elija Crear regla.
5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 64 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

   > **Note**
   >
   > Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

6. (Opcional) En Descripción, introduzca una descripción para la regla.
7. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un Servicio de AWS elemento de tu cuenta emite un evento, siempre va al bus de eventos predeterminado de tu cuenta.
8. (Opcional) Desactive la regla en el bus seleccionado si no desea ejecutarla inmediatamente.
9. En Tipo de regla, elija Programación.
10. Seleccione Continuar para crear la regla o Siguiente.
11. En Programar patrón, realice una de las siguientes acciones:

    - Elija Un horario detallado que se ejecute a una hora específica, como las 8:00 a. m. PST el primer lunes de cada mes y después introduzca la expresión cron. Para obtener más información, consulte Cron Expressions en la Guía del EventBridge usuario de Amazon.
    - Elija un horario que se ejecute a un ritmo regular, por ejemplo, cada 10 minutos. y, a continuación, introduzca una expresión de frecuencia.

12. Elija Siguiente.
13. En Tipos de destino, elija Servicio de AWS.
14. En Seleccione un objetivo, elija Cola de trabajos por lotes. A continuación, configure lo siguiente:
    - Cola de trabajos: introduzca el nombre de recurso de Amazon (ARN) de la cola de trabajos en la que va a programar el trabajo.
    - Definición de trabajo: introduzca el nombre y la revisión o el ARN completo de la definición de trabajo que va a utilizar para el trabajo.
    - Nombre de trabajo: introduzca un nombre para el trabajo.
    - Tamaño de la matriz: (opcional) introduzca un tamaño de matriz para que el trabajo ejecute más de una copia. Para obtener más información, consulte Trabajos de matrices.
    - Intentos de trabajo: (opcional) introduzca el número de veces que se debe reintentar el trabajo si se produce un error. Para obtener más información, consulte Reintentos automáticos de trabajo.
15. Para los tipos de objetivos de cola de trabajos por lotes, EventBridge necesita permiso para enviar eventos al destino. EventBridge puede crear la función de IAM necesaria para que se ejecute la regla. Realice una de las acciones siguientes:
    - Para crear un rol de IAM automáticamente, elija Crear un nuevo rol para este recurso específico.
    - Para utilizar un rol de IAM que haya creado antes, elija Usar rol existente.
16. (Opcional) Amplíe Configuración adicional.
    a. En Configurar la entrada de destino, elija cómo se procesa el texto de un evento antes de pasarlo al destino.
    b. En Antigüedad máxima del evento, especifique el intervalo de tiempo durante el que se guardan los eventos sin procesar.
    c. En Intentos de reintento, introduzca el número de veces que se volverá a intentar un evento.
    d. En Cola de mensajes fallidos, elija una opción para gestionar los eventos no procesados. Si es necesario, especifique la cola de Amazon SQS que se utilizará como la cola de mensajes fallidos.
17. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.
18. Elija Siguiente.

19. (Opcional) En Etiquetas, elija Añadir nueva etiqueta para añadir una etiqueta de recurso a la regla. Para obtener más información, consulta las EventBridge etiquetas de Amazon.
20. Elija Siguiente.
21. En Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Crear regla.

Para obtener más información sobre la creación de reglas, consulta Cómo crear una EventBridge regla de Amazon que se ejecute según un cronograma en la Guía del EventBridge usuario de Amazon.

Crear una regla con un patrón de evento

El siguiente procedimiento explica cómo crear una regla con un patrón de evento.

Para crear una regla que envíe el evento a un objetivo cuando el evento coincida con un patrón definido

> **Note**
>
> Este procedimiento funciona para todos los trabajos AWS Batch de Amazon ECS, Amazon EKS y AWS Fargate.

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En la barra de navegación, selecciona la Región de AWS que deseas usar.
3. En el panel de navegación, seleccione Reglas.
4. Elija Crear regla.
5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 64 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

   > **Note**
   >
   > Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

6. (Opcional) En Descripción, introduzca una descripción para la regla.
7. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un Servicio de AWS elemento de tu cuenta emite un evento, siempre va al bus de eventos predeterminado de tu cuenta.
8. (Opcional) Desactive la regla en el bus seleccionado si no desea ejecutarla inmediatamente.
9. En Tipo de regla, elija Regla con un patrón de evento.
10. Elija Siguiente.
11. En Origen del evento, selecciona AWS evento o eventos EventBridge asociados.
12. (Opcional) En Ejemplo de evento:

    a. En Ejemplo de tipo de evento, elija eventos de AWS .

    b. En Eventos de muestra, elija Cambiar estado de trabajo por lotes.
13. En Método de creación, elija Usar forma de patrón.
14. En Patrón de eventos:

    a. En Origen del evento, elija Servicios de AWS.

    b. En Servicio de AWS, elija Lote.

    c. En Tipo de evento, elija Cambio de estado de trabajo por lotes.
15. Elija Siguiente.
16. En Tipos de destino, elija Servicio de AWS.
17. En Elegir un tipo de destino, elija un tipo de destino. Por ejemplo, elija Cola de trabajos por lotes. A continuación, especifique lo siguiente:

    - Cola de trabajos: introduzca el nombre de recurso de Amazon (ARN) de la cola de trabajos en la que va a programar el trabajo.
    - Definición de trabajo: introduzca el nombre y la revisión o el ARN completo de la definición de trabajo que va a utilizar para el trabajo.
    - Nombre de trabajo: introduzca un nombre para el trabajo.
    - Tamaño de la matriz: (opcional) introduzca un tamaño de matriz para que el trabajo ejecute más de una copia. Para obtener más información, consulte Trabajos de matrices.
    - Intentos de trabajo: (opcional) introduzca el número de veces que se debe reintentar el trabajo si se produce un error. Para obtener más información, consulte Reintentos automáticos de trabajo.

18. Para los tipos de objetivos de cola de trabajos por lotes, EventBridge necesita permiso para enviar eventos al destino. EventBridge puede crear la función de IAM necesaria para que se ejecute la regla. Realice una de las acciones siguientes:

    - Para crear un rol de IAM automáticamente, elija Crear un nuevo rol para este recurso específico.
    - Para utilizar un rol de IAM que haya creado antes, elija Usar función existente.

19. (Opcional) Amplíe Configuración adicional.

    a. En Configurar la entrada de destino, elija cómo se procesa el texto de un evento.

    b. En Antigüedad máxima del evento, especifique el intervalo de tiempo durante el que se guardan los eventos sin procesar.

    c. En Intentos de reintento, introduzca el número de veces que se volverá a intentar un evento.

    d. En Cola de mensajes fallidos, elija una opción para gestionar los eventos no procesados. Si es necesario, especifique la cola de Amazon SQS que se utilizará como la cola de mensajes fallidos.

20. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.
21. Elija Siguiente.
22. (Opcional) En Etiquetas, elija Añadir nueva etiqueta para añadir una etiqueta de recurso. Para obtener más información, consulta las EventBridge etiquetas de Amazon en la Guía del EventBridge usuario de Amazon.
23. Elija Siguiente.
24. En Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, elija Crear regla.

    Para obtener más información sobre la creación de reglas, consulta Cómo crear EventBridge reglas de Amazon que reaccionen a los eventos en la Guía del EventBridge usuario de Amazon.

Pasar la información del evento a un AWS Batch objetivo según un cronograma mediante el transformador EventBridge de entrada

Puede usar el transformador EventBridge de entrada para pasar la información del evento al AWS Batch enviar un trabajo. Esto puede resultar especialmente valioso si se invocan trabajos como resultado de otra información sobre eventos de AWS . Un ejemplo es la carga de un objeto en un bucket de Amazon S3. También puede usar una definición de trabajo con valores de sustitución

de parámetros en el comando del contenedor. El transformador EventBridge de entrada puede proporcionar los valores de los parámetros en función de los datos del evento.

A continuación, se crea un objetivo de AWS Batch evento que analiza la información del evento que lo inicia y la transforma en un `parameters` objeto. Cuando se ejecuta el trabajo, los parámetros del evento que se desencadena se pasan al comando del contenedor del trabajo.

> **Note**
>
> En este escenario, todos los AWS recursos (como los depósitos, EventBridge las reglas y los CloudTrail registros de Amazon S3) deben estar en la misma región.

Para crear un AWS Batch objetivo que utilice el transformador de entrada

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En la barra de navegación, selecciona la Región de AWS que deseas usar.
3. En el panel de navegación, seleccione Reglas.
4. Elija Crear regla.
5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 64 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

   > **Note**
   >
   > Una regla no puede tener el mismo nombre que otra regla en el mismo bus de eventos Región de AWS y en el mismo.

6. (Opcional) En Descripción, introduzca una descripción para la regla.
7. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un Servicio de AWS elemento de tu cuenta emite un evento, siempre va al bus de eventos predeterminado de tu cuenta.
8. (Opcional) Desactive la regla en el bus seleccionado si no desea ejecutarla inmediatamente.
9. En Tipo de regla, elija Programación.
10. Seleccione Continuar para crear la regla o Siguiente.
11. En Programar patrón, realice una de las siguientes acciones:

- Elija Un horario detallado que se ejecute a una hora específica, como las 8:00 a. m. PST el primer lunes de cada mes y después introduzca la expresión cron. Para obtener más información, consulte Cron Expressions en la Guía del EventBridge usuario de Amazon.
- Elija un horario que se ejecute a un ritmo regular, por ejemplo, cada 10 minutos. y, a continuación, introduzca una expresión de frecuencia.

12. Elija Siguiente.
13. En Tipos de destino, elija Servicio de AWS.
14. En Seleccione un objetivo, elija Cola de trabajos por lotes. A continuación, configure lo siguiente:

    - Cola de trabajos: introduzca el nombre de recurso de Amazon (ARN) de la cola de trabajos en la que va a programar el trabajo.
    - Definición de trabajo: introduzca el nombre y la revisión o el ARN completo de la definición de trabajo que va a utilizar para el trabajo.
    - Nombre de trabajo: introduzca un nombre para el trabajo.
    - Tamaño de la matriz: (opcional) introduzca un tamaño de matriz para que el trabajo ejecute más de una copia. Para obtener más información, consulte Trabajos de matrices.
    - Intentos de trabajo: (opcional) introduzca el número de veces que se debe reintentar el trabajo si se produce un error. Para obtener más información, consulte Reintentos automáticos de trabajo.
15. Para los tipos de objetivos de cola de trabajos por lotes, EventBridge necesita permiso para enviar eventos al destino. EventBridge puede crear la función de IAM necesaria para que se ejecute la regla. Realice una de las acciones siguientes:

    - Para crear un rol de IAM automáticamente, elija Crear un nuevo rol para este recurso específico.
    - Para utilizar un rol de IAM que haya creado antes, elija Usar rol existente.
16. (Opcional) Amplíe Configuración adicional.
17. En la sección Ajustes adicionales, en Configurar entrada de destino, elija Transformador de entrada.
18. Elija Configurar transformador de entrada.
19. (Opcional) En Ejemplo de evento:

    a. En Ejemplo de tipo de evento, elija eventos de AWS .
    b. En Eventos de muestra, elija Cambiar estado de trabajo por lotes.

20. En la sección Transformador de entrada de destino en Ruta de entrada, especifique los valores que se van a analizar del evento que se desencadena. Por ejemplo, para analizar el evento Cambiar estado de trabajo por lotes, utilice el siguiente formato JSON.

```
{
    "instance": "$.detail.jobId",
    "state": "$.detail.status"
}
```

21. En Plantilla, introduzca lo siguiente.

```
{
    "instance": <jobId> ,
    "status": <status>
}
```

22. Seleccione Confirmar.
23. En Antigüedad máxima del evento, especifique el intervalo de tiempo durante el que se guardan los eventos sin procesar.
24. En Intentos de reintento, introduzca el número de veces que se volverá a intentar un evento.
25. En Cola de mensajes fallidos, elija una opción para gestionar los eventos no procesados. Si es necesario, especifique la cola de Amazon SQS que se utilizará como la cola de mensajes fallidos.
26. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.
27. Elija Siguiente.
28. (Opcional) En Etiquetas, elija Añadir nueva etiqueta para añadir una etiqueta de recurso. Para obtener más información, consulta las EventBridge etiquetas de Amazon en la Guía del EventBridge usuario de Amazon.
29. Elija Siguiente.
30. En Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, elija Crear regla.

Tutorial: Listening for AWS Batch EventBridge

En este tutorial, se configura una función simple AWS Lambda que escucha eventos de trabajo AWS Batch y los escribe en un flujo de registro de CloudWatch Logs.

Requisitos previos

En este tutorial se supone que tiene un entorno informático y una cola de trabajos que están listos para aceptar trabajos. Si no dispone de un entorno informático y una cola de trabajos para capturar eventos, siga los pasos de Cómo empezar con AWS Batch para crearlos. Al final de este tutorial, puede enviar opcionalmente un trabajo a esta cola de trabajos para comprobar que ha configurado correctamente su función de Lambda.

Paso 1: crear la función de Lambda

En este procedimiento, creará una función de Lambda sencilla que servirá como destino para los mensajes del flujo de eventos de AWS Batch.

Para crear una función de Lambda de destino

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. Elija Create function (Crear función) y, a continuación, elija Author from scratch (Crear desde cero).
3. En Nombre de función, introduzca batch-event-stream-handler.
4. En Runtime (Tiempo de ejecución), elija Python 3.8.
5. Elija Crear función.
6. En la sección Código fuente, edite el código de muestra para que coincida con el siguiente ejemplo:

```
import json


def lambda_handler(event, _context):
    # _context is not used
    del _context
    if event["source"] != "aws.batch":
        raise ValueError("Function only supports input from events with a source
 type of: aws.batch")

    print(json.dumps(event))
```

Se trata de una función simple de Python 3.8 que imprime los eventos enviados por AWS Batch. Si todo está configurado correctamente, al final de este tutorial verá los detalles de los eventos del flujo de registros de CloudWatch Logs asociado a esta función Lambda.

7. Elija Deploy (Implementar).

Paso 2: Registrar una regla de eventos

En esta sección, se crea una regla de evento de EventBridge que captura eventos de trabajos procedentes de sus recursos de AWS Batch. Esta regla captura todos los eventos procedentes de AWS Batch dentro de la cuenta donde se define. Los propios mensajes de trabajo contienen información sobre el origen del evento, incluida la cola de trabajo en la que se envió. Puede utilizar esta información para filtrar y ordenar eventos de manera programática.

> **Note**
>
> Si utiliza el AWS Management Console para crear una regla de evento, la consola añade automáticamente los permisos IAM para EventBridge para llamar a su función de Lambda. Sin embargo, si crea una regla de evento utilizando la AWS CLI, tiene que otorgar permisos explícitamente. Para obtener más información, consulte Eventos y patrones de eventos en EventBridge en la Guía del usuario de Amazon EventBridge.

Para crear su regla EventBridge

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.
4. Escriba un nombre y una descripción de la regla.

   Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.
5. En Event bus (Bus de eventos), elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija Other (Otro).
9. En Patrón de eventos, seleccione Patrones personalizados (editor de JSON).

10. Pegue el siguiente patrón de eventos en el área de texto.

```
{
  "source": [
    "aws.batch"
  ]
}
```

    Esta regla se aplica a todos los grupos de AWS Batch y a todos los eventos de AWS Batch. También puede crear una regla más específica para filtrar algunos resultados.

11. Elija Next (Siguiente).
12. En Target types (Tipos de destino), elija AWS service.
13. En Seleccionar un objetivo, elija Función de Lambda y seleccione su función de Lambda.
14. (Opcional) En Additional settings (Configuración adicional), haga lo siguiente:

    a. En Maximum age of event (Antigüedad máxima del evento), ingrese un valor entre un minuto (00:01) y 24 horas (24:00).

    b. En Retry attempts (Cantidad de reintentos), ingrese un número entre 0 y 185.

    c. En Dead-letter queue (Cola de mensajes fallidos), elija si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Haga una de las siguientes acciones:

    - Elija None (Ninguno) para no usar una cola de mensajes fallidos.
    - Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para usarla como cola de mensajes fallidos y luego seleccione de la lista desplegable la cola que quiera usar.
    - Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes. Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos en la Guía del usuario de Amazon EventBridge.

15. Elija Next (Siguiente).
16. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte Amazon EventBridge tags (Etiquetas de Amazon EventBridge) en la Guía del usuario de Amazon EventBridge.

17. Elija Next (Siguiente).
18. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 3: Probar la configuración

Ahora puede probar la configuración de EventBridge enviando una tarea a la cola de tareas. Si todo está configurado correctamente, su función de Lambda se activa y escribe los datos de eventos en un flujo de registro de CloudWatch Logs para la función.

Para probar la configuración

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. Envíe un nuevo trabajo de AWS Batch. Para obtener más información, consulte Enviar un trabajo.
3. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
4. En el panel de navegación, elija Logs (Registros) y seleccione el grupo de registros para la función de Lambda (por ejemplo, /aws/lambda/*my-function*).
5. Seleccione una secuencia de registro para ver los datos de los eventos.

Tutorial: envío de alertas de Amazon Simple Notification Service Alerts para eventos de trabajos fallidos

En este tutorial, configurará una regla de EventBridge eventos que solo capture los eventos de trabajo en los que el trabajo haya pasado a un `FAILED` estado. Al final de este tutorial, si lo desea, también puede enviar un trabajo a esta cola de trabajos. Esto sirve para comprobar que ha configurado correctamente las alertas de Amazon SNS.

Requisitos previos

En este tutorial se supone que tiene un entorno de computación y una cola de trabajos que están listos para aceptar trabajos. Si no dispone de un entorno de computación y una cola de trabajos para capturar eventos, siga los pasos de Cómo empezar con AWS Batch para crearlos.

Paso 1: Crear y suscribirse a un tema de Amazon SNS

Para este tutorial, se configura un tema de Amazon SNS para utilizarse como destino de eventos para la nueva regla de eventos.

Para crear un tema de Amazon SNS

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. Seleccione Topics (Temas), Create topic (Crear tema).
3. En Tipo, seleccione Estándar.
4. Para Nombre, escriba **`JobFailedAlert`** y, a continuación, elija Crear tema.
5. En la JobFailedAlertpantalla, elija Crear suscripción.
6. En Protocolo, seleccione Correo electrónico.
7. En Endpoint (Punto de conexión), ingrese una dirección de email a la que actualmente tenga acceso y elija Create subscription (Crear suscripción).
8. Consulte su cuenta de correo electrónico y espere para recibir un mensaje de correo electrónico de confirmación de la suscripción. Cuando lo reciba, seleccione Confirmar suscripción.

Paso 2: Registrar una regla de eventos

A continuación, registre una regla de eventos que solo capture los eventos de trabajos con error.

Para registrar tu EventBridge regla

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Seleccione Crear regla.
4. Escriba un nombre y una descripción para la regla.

   Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.
5. En Bus de eventos, seleccione el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS . Cuando un AWS servicio de tu cuenta emite un evento, siempre va al bus de eventos predeterminado de tu cuenta.

6. En Tipo de regla, elija Regla con un patrón de evento.
7. Seleccione Siguiente.
8. En Origen del evento, seleccione Otro.
9. En Patrón de eventos, seleccione Patrones personalizados (editor de JSON).
10. Pegue el siguiente patrón de eventos en el área de texto.

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

Este código define una EventBridge regla que coincide con cualquier evento en el que se encuentre el estado del puesto. `FAILED` Para obtener más información sobre los patrones de eventos, consulta Eventos y patrones de eventos en la Guía del EventBridge usuario de Amazon.

11. Seleccione Siguiente.
12. En Tipos de destino, seleccione Servicio de AWS .
13. En Seleccione un objetivo, elija un tema de SNS y, en Tema, elija JobFailedAlert.
14. (Opcional) En Configuración adicional, haga lo siguiente:

    a. En Antigüedad máxima del evento, indique un valor entre un minuto (00:01) y 24 horas (24:00).

    b. En Cantidad de reintentos, indique un número entre 0 y 185.

    c. En el caso de la cola de cartas sin salida, elija si desea utilizar una cola estándar de Amazon SQS como cola de cartas sin salida. EventBridge envía los eventos que cumplen con esta regla a la lista de espera si no se entregan correctamente al destino. Realice una de las acciones siguientes:

       - Seleccione Ninguno para no usar una cola de mensajes fallidos.

- Elija Seleccione una cola de Amazon SQS en la AWS cuenta corriente para utilizarla como cola de letra muerta y, a continuación, seleccione la cola que desee utilizar en el menú desplegable.
- Elija Seleccione una cola de Amazon SQS en otra AWS cuenta como cola de letra muerta y, a continuación, introduzca el ARN de la cola que desee utilizar. Debe adjuntar a la cola una política basada en recursos que le conceda permiso para enviarle mensajes. EventBridge Para obtener más información, consulta Cómo conceder permisos a la cola de letra muerta en la Guía del usuario de Amazon EventBridge .

15. Seleccione Siguiente.
16. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulta las EventBridge etiquetas de Amazon en la Guía del EventBridge usuario de Amazon.
17. Seleccione Siguiente.
18. Revise los detalles de la regla y seleccione Crear regla.

Paso 3: Comprobación de la regla

Para probar la regla, envíe un trabajo que se cierre poco después de que comience y tenga un código de salida distinto de cero. Si su regla de eventos está configurada correctamente, debería recibir un mensaje de correo electrónico en unos minutos con el texto del evento.

Para probar una regla

1. Abre la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. Envíe un nuevo AWS Batch trabajo. Para obtener más información, consulte Enviar un trabajo. Para el comando del trabajo, sustituya este comando para salir del contenedor con un código de salida de 1.

   ```
   /bin/sh, -c, 'exit 1'
   ```

3. Compruebe su correo electrónico para confirmar que ha recibido una alerta de correo electrónico para la notificación de trabajo fallido.

Regla alternativa: cola de trabajos por lotes bloqueada

Para crear una regla de eventos que supervise el bloqueo de la cola de trabajos por lotes, repita los pasos de este tutorial con las siguientes modificaciones:

1. En el paso 1, *BlockedJobQueue*utilícela como nombre del tema.
2. En el paso 2, usa el siguiente patrón en el editor JSON:

```
{
   "detail-type": [
     "Batch Job Queue Blocked"
   ],
   "source": [
     "aws.batch"
   ]
}
```

Uso de CloudWatch Logs con AWS Batch

Puede configurar sus trabajos AWS Batch en los recursos de EC2 para enviar información de registro y métricas detalladas a CloudWatch Logs. De esta forma, puede ver distintos registros desde sus trabajos en una ubicación cómoda. Para obtener más información acerca de CloudWatch, consulte ¿Qué son los registros de Amazon CloudWatch? en la Guía del usuario de Amazon CloudWatch.

> **Note**
>
> De forma predeterminada, CloudWatch Logs está activado en los contenedores de Fargate AWS.

Para activar y personalizar el registro de CloudWatch Logs, revise las siguientes tareas de configuración únicas:

- Para los entornos informáticos AWS Batch que se basan en recursos de EC2, añada una política de IAM al rol `ecsInstanceRole`. Para obtener más información, consulte the section called “Añadir una política de IAM de CloudWatch Logs”.
- Cree una plantilla de lanzamiento de Amazon EC2 que incluya una supervisión detallada de CloudWatch y, a continuación, especifique la plantilla al crear su entorno informático AWS Batch. También puede instalar el agente de CloudWatch en una imagen existente y, a continuación, especificar la imagen en el asistente de primera ejecución AWS Batch.
- (Opcional) Configure el controlador awslogs. Puede agregar parámetros que cambien el comportamiento predeterminado en los recursos de EC2 y Fargate. Para obtener más información, consulte the section called “Uso del controlador de registros awslogs”.

Añadir una política de IAM de CloudWatch Logs

Para que los trabajos puedan enviar datos de registros y métricas detalladas a CloudWatch Logs, debe crear una política de IAM que utilice las API de CloudWatch Logs. Después de crear la política de IAM, asóciela al rol `ecsInstanceRole`.

> **Note**
>
> Si la política `ECS-CloudWatchLogs` no está asociada al rol `ecsInstanceRole`, se pueden enviar métricas básicas a CloudWatch Logs. Sin embargo, las métricas básicas no incluyen datos de registro ni métricas detalladas, como el espacio libre en disco.

Los entornos de cómputo AWS Batch utilizan recursos de Amazon EC2. Al crear un entorno informático mediante el asistente de primera ejecución AWS Batch, AWS Batch crea el rol `ecsInstanceRole` y configura el entorno con él.

Si no utiliza el asistente de primera ejecución, puede especificar el rol `ecsInstanceRole` al crear un entorno informático en la API AWS Command Line Interface o AWS Batch. Para obtener más información, consulte Referencia de comandos AWS CLI o Referencia de API AWS Batch.

Para crear la política de IAM **`ECS-CloudWatchLogs`**

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Políticas.
3. Elija Crear política.
4. Elija JSON y, a continuación, escriba la política siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        }
    ]
}
```

5. Elija Siguiente: etiquetas.
6. (Opcional) En Añadir etiquetas, elije Añadir etiqueta para añadir una etiqueta a la política.
7. Elija Siguiente: revisar.
8. En la página Revisar política, en Nombre, escriba **`ECS-CloudWatchLogs`**, y después introduzca una Descripción opcional.
9. Elija Crear política.

Asociación de la política **`ECS-CloudWatchLogs`** a **`ecsInstanceRole`**

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación.
3. Elija `ecsInstanceRole`. Si el rol no existe, siga los procedimientos que se indican en Función de instancia de Amazon ECS para crear el rol.
4. Elija Agregar permisos y luego Adjuntar políticas.
5. Elija la política ECS-CloudWatchLogs y haga clic en Adjuntar política.

Instalación y configuración del agente de CloudWatch

Puede crear una plantilla de lanzamiento de Amazon EC2 que incluya la supervisión de CloudWatch. Para obtener más información, consulte, Lanzar una instancia desde una plantilla de lanzamiento y Detalles avanzados en la Guía del usuario de Amazon EC2 para instancias de Linux.

También puede instalar el agente CloudWatch en una AMI de Amazon EC2 existente y, a continuación, especificar la imagen en el asistente de primera ejecución AWS Batch. Para obtener más información, consulte Instalación del agente de CloudWatch y Introducción a AWS Batch.

> **Note**
>
> Los recursos AWS Fargate no admiten plantillas de lanzamiento.

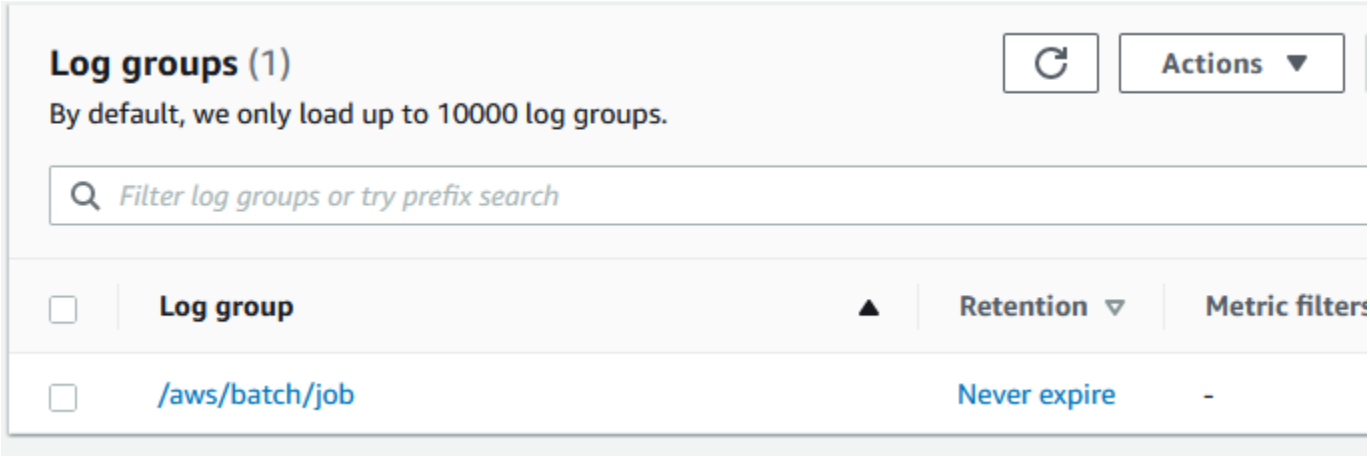
Ver CloudWatch Logs

Puede ver y buscar los registros de CloudWatch Logs en AWS Management Console.

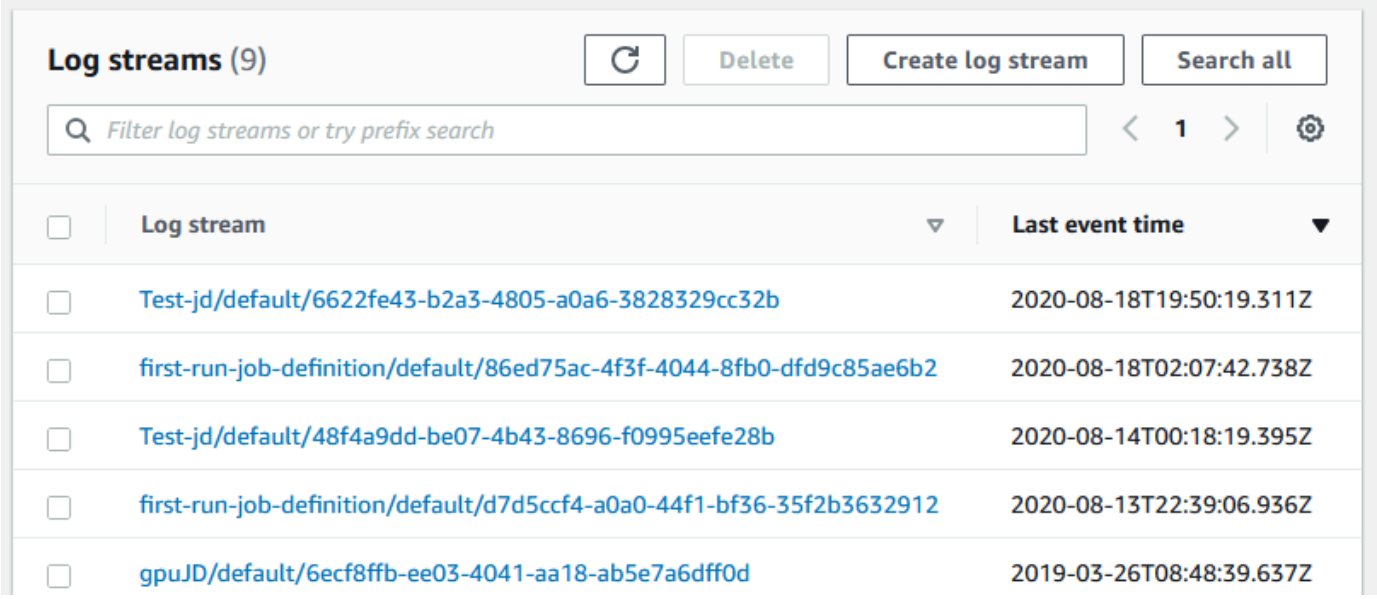
> **Note**
>
> Es posible que los datos tarden unos minutos en mostrarse en CloudWatch Logs.

Para consultar los datos de CloudWatch Logs

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, elija Registros, luego, Grupos de registros.

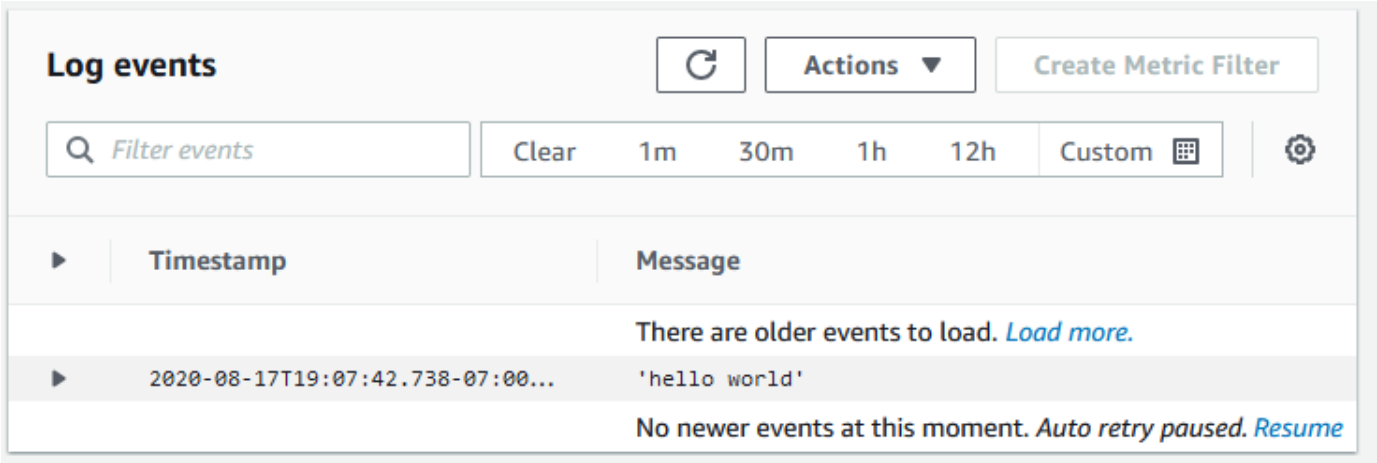


3. Seleccione un grupo de registros que desea ver.



4. Elija el flujo de registros que desea ver. De forma predeterminada, las transmisiones se identifican con los primeros 200 caracteres del nombre del trabajo y el ID de la tarea de Amazon ECS.

> **Tip**
>
> Para descargar los datos del flujo de registro, elija Acciones.



Utilice CloudWatch Logs para supervisar AWS Batch en los trabajos de Amazon EKS

Puede utilizar Registros de Amazon CloudWatch para monitorizar, almacenar y ver todos los archivos de registro en un solo lugar. Con CloudWatch Logs, puede buscar, filtrar y analizar datos de registro de varias fuentes.

Puede descargar una AWS para una imagen Fluent Bit que incluya un complemento para supervisar AWS Batch de Amazon EKS en CloudWatch Logs. Fluent Bit es un procesador y reenviador de registros de código abierto que es compatible con Docker y Kubernetes. Le recomendamos que utilice Fluent Bit como router de registro porque consume menos recursos que Fluentd. Para obtener más información, consulte Uso de AWS para la imagen de Fluent Bit..

Requisitos previos

Adjunte la política `CloudWatchAgentServerPolicy` a la política AWS Identity and Access Management de sus nodos de trabajo. Para obtener más información, consulte Verificación de los requisitos previos.

Instale AWS para Fluent Bit

Para obtener instrucciones sobre cómo instalar AWS para Fluent Bit y crear los grupos de CloudWatch, consulte Configuración de Fluent Bit o Inicio rápido con el agente de CloudWatch y Fluent Bit.

> **Tip**
>
> Recuerde que Fluent Bit utiliza 0,5 CPU y 100 MB de memoria en los nodos AWS Batch. Esto reduce la capacidad total disponible para los trabajos AWS Batch. Tenga esto en cuenta a la hora de dimensionar sus trabajos.

Active Fluent Bit para los nodos AWS Batch

Para garantizar que el registro de DaemonSet Fluent Bit se ejecute en los nodos gestionados AWS Batch, modifique las tolerancias de DaemOnset Fluent Bit:

```
tolerations:
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
```

AWS Batch Información de contenedores de CloudWatch

CloudWatch Container Insights recopila, agrega y resume métricas y registros de sus trabajos y entornos informáticos de AWS Batch. Las métricas incluyen la utilización de recursos como CPU, memoria, disco y red. Puede añadir estas métricas a los paneles de CloudWatch.

Los datos operativos se recopilan como eventos de registro de rendimiento. Son entradas que utilizan un esquema JSON estructurado que permite incorporar y almacenar datos de cardinalidad alta a escala. A partir de estos datos, CloudWatch crea métricas agregadas de nivel superior a nivel de entorno informático y de trabajo como métricas CloudWatch. Para obtener más información, consulte Registros estructurados de Container Insights para Amazon ECS en la Guía del usuario de Amazon CloudWatch.

> **Important**
>
> Las métricas recopiladas por CloudWatch Container Insights se cobran como métricas personalizadas. Para obtener más información, consulte los precios de los Eventos de Amazon CloudWatch .

Active Container Insights.

Puede activar Container Insights para los entornos informáticos de AWS Batch.

1. Abra la consola de AWS Batch.
2. Elija Entornos informáticos.
3. Elija el entorno de desarrollo que quiera.
4. En Container Insights, active Container Insights para el entorno informático.

> **Tip**
>
> Puede seleccionar un intervalo predeterminado para agregar las métricas o crear un intervalo personalizado.

De forma predeterminada, se muestran las siguientes métricas. Para obtener una lista completa de las métricas de Amazon ECS Container Insights, consulte Métricas de Amazon ECS Container Insights en la Guía del usuario de Amazon CloudWatch.

- **`JobCount`** — El número de trabajos que se ejecutan en el entorno informático.
- **`ContainerInstanceCount`** — El número de instancias de Amazon Elastic Compute Cloud que ejecutan el agente de Amazon ECS y están registradas en el entorno informático.
- **`MemoryReserved`** — La memoria que se reserva para los trabajos del entorno informático. Esta métrica se recopila únicamente para las tareas que tienen una reserva de memoria definida en su definición de tarea.
- **`MemoryUtilized`** — La memoria que están utilizando los trabajos del entorno informático. Esta métrica se recopila únicamente para las tareas que tienen una reserva de memoria definida en su definición de tarea.
- **`CpuReserved`** — Las unidades CPU que se reservan para los trabajos del entorno informático. Esta métrica se recopila únicamente para las tareas que tienen una reserva definida de la CPU en la definición de contenedor.
- **`CpuUtilized`** — Las unidades CPU utilizadas por los trabajos en el entorno informático. Esta métrica se recopila únicamente para las tareas que tienen una reserva definida de la CPU en la definición de contenedor.
- **`NetworkRxBytes`**- El número de bytes que se reciben. Esta métrica solo está disponible para los contenedores en tareas que utilizan `awsvpc` o los modos de red puente.
- **`NetworkTxBytes`** — El número de bytes que se transmiten. Esta métrica solo está disponible para los contenedores en tareas que utilizan `awsvpc` o los modos de red puente.
- **`StorageReadBytes`** — El número de bytes que se leen del almacenamiento.
- **`StorageWriteBytes`** — El número de bytes que se escriben en el almacenamiento.

Registro de AWS Batch llamadas a la API de con AWS CloudTrail

AWS Batch se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones hechas por un usuario, un rol o un servicio de AWS en AWS Batch. CloudTrail captura las llamadas a la API de AWS Batch como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de AWS Batch y las llamadas desde el código a las operaciones de la API de AWS Batch. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos para AWS Batch. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a AWS Batch, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc.

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

Información de AWS Batch en CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce una actividad en AWS Batch, esa actividad se registra en un evento de CloudTrail junto con otros eventos de servicio de AWS en Event history (Historial de eventos). Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte Ver eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de eventos en la cuenta de AWS, incluidos los eventos de AWS Batch, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configuración de notificaciones de Amazon SNS para CloudTrail

- Recibir archivos de registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas las acciones de AWS Batch y se documentan en https://docs.aws.amazon.com/batch/latest/APIReference/. Por ejemplo, las llamadas a las secciones `SubmitJob`, `ListJobs` y `DescribeJobs` generan entradas en los archivos de registro de CloudTrail.

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

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM de .
- 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 el Elemento userIdentity de CloudTrail.

Descripción de las entradas de archivos de registro de AWS Batch

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

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateComputeEnvironment`.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-12-20T00:48:46Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::012345678910:role/Admin",
        "accountId": "012345678910",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2017-12-20T00:48:46Z",
  "eventSource": "batch.amazonaws.com",
  "eventName": "CreateComputeEnvironment",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.1",
  "userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
  "requestParameters": {
    "computeResources": {
      "subnets": [
        "subnet-5eda8e04"
      ],
      "tags": {
        "testBatchTags": "CLI testing CE"
      },
      "desiredvCpus": 0,
      "minvCpus": 0,
      "instanceTypes": [
        "optimal"
      ],
      "securityGroupIds": [
        "sg-aba9e8db"
      ],
      "instanceRole": "ecsInstanceRole",
      "maxvCpus": 128,
      "type": "EC2"
    },
    "state": "ENABLED",
    "type": "MANAGED",
    "computeEnvironmentName": "Test"
  },
```

```
  "responseElements": {
    "computeEnvironmentName": "Test",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/
Test"
  },
  "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
  "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "012345678910"
}
```

Creación de una nube virtual privada (VPC)

Los recursos informáticos de sus entornos informáticos necesitan acceso de red externo para comunicarse con AWS Batch y el punto de conexión del servicio de Amazon ECS. Sin embargo, es posible que tenga trabajos que desee ejecutar en subredes privadas. La creación de una VPC con subredes públicas y privadas le proporciona la flexibilidad necesaria para ejecutar trabajos en una subred pública o privada.

Puede usar Amazon Virtual Private Cloud (Amazon VPC) para lanzar recursos de AWS en una red virtual que haya definido. En este tema se proporciona un enlace al asistente de Amazon VPC y una lista de las opciones que se pueden seleccionar.

Creación de una VPC

Para obtener información sobre cómo crear una VPC, consulte Crear una VPC únicamente en la Guía del usuario de Amazon VPC y utilice la siguiente tabla para determinar qué opciones seleccionar.

Opción	Valor
Recursos para crear	VPC solo
Nombre	De manera opcional, indique un nombre para su VPC.
IPv4 CIDR block	Entrada manual de IPv4 CIDR El bloque de CIDR debe ser de un tamaño de entre /16 y /28.
IPv6 CIDR block	No hay bloque de CIDR IPv6
Propiedad	Valor predeterminado

Para obtener más información acerca de Amazon VPC, consulte ¿Qué es Amazon VPC? en la Guía del usuario de Amazon VPC.

Pasos siguientes

Después de crear la VPC, se recomiendan los siguientes pasos:

- Cree grupos de seguridad para los recursos públicos y privados si necesitan acceso de red de entrada. Para obtener más información, consulte Uso de grupos de seguridad en la Guía del usuario de Amazon VPC.
- Cree un entorno informático administrado de AWS Batch que lance recursos informáticos en la nueva VPC. Para obtener más información, consulte Cómo crear un entorno de computación. Si utiliza el asistente de creación de entornos informáticos en la consola de AWS Batch, puede especificar la VPC recién creada y las subredes públicas o privadas en las que lanzar las instancias, en función de su caso de uso.
- Cree una cola de trabajos de AWS Batch que se asigne al nuevo entorno informático. Para obtener más información, consulte Cómo crear de una cola de trabajos.
- Cree una definición de trabajo para ejecutar sus trabajos. Para obtener más información, consulte Creación de una definición de trabajo de un solo nodo .
- Envíe un trabajo con su definición de trabajo a la nueva cola de trabajos. Este trabajo llegará al entorno informático que creó con su nueva VPC y sus subredes. Para obtener más información, consulte Enviar un trabajo.

Seguridad en AWS Batch

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

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

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS Batch, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos vigentes.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS Batch. Los siguientes temas muestran cómo configurarlo AWS Batch para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus AWS Batch recursos.

Temas

Identity and Access Management para AWS Batch

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de

IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. AWS Batch La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice. AWS Batch

Usuario del servicio: si utiliza el AWS Batch servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más AWS Batch funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en AWS Batch, consulte Solución de problemas AWS Batch de identidad y acceso.

Administrador de servicios: si estás a cargo de AWS Batch los recursos de tu empresa, probablemente tengas acceso total a ellos AWS Batch. Su trabajo consiste en determinar a qué AWS Batch funciones y recursos deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar la IAM AWS Batch, consulte¿Cómo AWS Batch funciona con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a AWS. Para ver ejemplos de políticas AWS Batch basadas en la identidad que puede utilizar en IAM, consulte. Ejemplos de políticas basadas en la identidad para AWS Batch

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la

contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS Single Sign-On. Puede crear usuarios y grupos en el Centro de identidades de IAM o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS Single Sign-On.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute

aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.

- Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

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

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que 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 de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

¿Cómo AWS Batch funciona con IAM

Antes de utilizar IAM para gestionar el acceso AWS Batch, infórmese sobre las funciones de IAM disponibles para su uso. AWS Batch

Funciones de IAM que puede utilizar con AWS Batch

Característica de IAM	AWS Batch soporte
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí

Característica de IAM	AWS Batch soporte
Recursos de políticas	Sí
Claves de condición de política	Sí
ACL	No
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	Sí

Para obtener una visión general de cómo AWS Batch funcionan otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en la identidad para AWS Batch

Compatibilidad con las políticas basadas en identidades	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en

una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades de AWS Batch

Para ver ejemplos de políticas AWS Batch basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Batch

Acciones políticas para AWS Batch

Admite acciones de política	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de AWS Batch acciones, consulte las acciones definidas por AWS Batch en la Referencia de autorización del servicio.

Las acciones políticas AWS Batch utilizan el siguiente prefijo antes de la acción:

```
batch
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "batch:action1",
      "batch:action2"
         ]
```

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": "batch:Describe*"
```

Para ver ejemplos de políticas AWS Batch basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Batch

Recursos de políticas para AWS Batch

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de AWS Batch recursos y sus ARN, consulte los recursos definidos por AWS Batch en la Referencia de autorización de servicios. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS Batch.

Para ver ejemplos de políticas AWS Batch basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Batch

Claves de condición de políticas para AWS Batch

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de claves de AWS Batch condición, consulte las claves de condición AWS Batch en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por AWS Batch.

Para ver ejemplos de políticas AWS Batch basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Batch

Control de acceso basado en atributos (ABAC) con AWS Batch

Admite ABAC (etiquetas en las políticas)	Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

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

Uso de credenciales temporales con AWS Batch

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios de AWS Batch

Admite Forward access sessions (FAS)	Sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Funciones de servicio para AWS Batch

Compatible con roles de servicio	Sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir AWS Batch la funcionalidad. Edite los roles de servicio sólo cuando AWS Batch proporcione orientación para hacerlo.

Funciones vinculadas al servicio para AWS Batch

Compatible con roles vinculados al servicio	Sí

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

AWS Batch función de IAM de ejecución

La función de ejecución otorga al contenedor y a los AWS Fargate agentes de Amazon ECS permiso para realizar llamadas a la AWS API en su nombre.

> **Note**
>
> El rol de ejecución de tareas es compatible con la versión 1.16.0 y posteriores del agente de contenedor de Amazon ECS.

El rol de IAM de ejecución de tareas es necesario en función de los requisitos de la tarea. Puede tener varios roles de ejecución de tareas asociados a su cuenta para distintos fines y servicios.

> **Note**
>
> Para obtener más información sobre el rol de instancia de Amazon ECS, consulte Función de instancia de Amazon ECS. Para obtener más información acerca de los roles de servicio, consulte ¿Cómo AWS Batch funciona con IAM.

Amazon ECS proporciona la política administrada `AmazonECSTaskExecutionRolePolicy`. Esta política contiene los permisos necesarios para los casos de uso comunes descritos anteriormente.

Puede ser necesario agregar políticas en línea al rol de ejecución de tareas para los casos de uso especiales que se describen a continuación.

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

Puede utilizar el procedimiento siguiente para comprobar si la cuenta ya dispone del rol de ejecución de tareas y para asociar la política administrada de IAM en caso necesario.

Para verificar el **`ecsTaskExecutionRole`** en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación.
3. En la lista de roles, busque `ecsTaskExecutionRole`. Si no puede encontrar el rol, consulte Creación del rol de IAM de ejecución. Si el rol existe, selecciónelo para ver sus políticas asociadas.
4. En la pestaña Permisos, compruebe que la política TaskExecutionRolePolicy gestionada por AmazonECS esté asociada a la función. Si la política se ha asociado, el rol de ejecución de la tarea está configurado correctamente. En caso contrario, siga los pasos derivados a continuación para asociar la política.

   a. Elija Agregar permisos y luego Adjuntar políticas.

   b. Busque AmazonECS TaskExecutionRolePolicy.

c. Marca la casilla a la izquierda de la política de AmazonECS y selecciona Adjuntar TaskExecutionRolePolicy políticas.

5. Seleccione Trust Relationships.
6. Verifique que la relación de confianza contiene la siguiente política. Si la relación de confianza coincide con la política siguiente, el rol está configurado correctamente. Si la relación de confianza no coincide, elija Editar la política de confianza, introduzca lo siguiente y después elija Actualizar política.

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

Creación del rol de IAM de ejecución

Si su cuenta aún no tiene un rol de ejecución de tareas, siga los pasos siguientes para crear el rol.

Para crear el rol de IAM **`ecsTaskExecutionRole`**

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación.
3. Elija Crear rol.
4. En Tipo de entidad de confianza, elija Servicio de AWS.
5. En Servicio o caso de uso, elija EC2. A continuación, vuelva a elegir EC2.
6. Elija Siguiente.
7. Para ver las políticas de permisos, busca AmazonECS TaskExecutionRolePolicy.
8. Seleccione la casilla de verificación situada a la izquierda de la TaskExecutionRolePolicy política de AmazonECS y, a continuación, seleccione Siguiente.

9. En Nombre de rol, escriba `ecsTaskExecutionRole` y luego elija Crear rol.

Ejemplos de políticas basadas en la identidad para AWS Batch

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de AWS Batch . Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede agregar las políticas de IAM a los roles, y los usuarios pueden asumirlos.

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

Para obtener más información sobre las acciones y los tipos de recursos definidos por cada uno de los tipos de recursos AWS Batch, incluido el formato de los ARN para cada uno de los tipos de recursos, consulte las claves de condición, recursos y acciones de la Referencia de autorización de servicios. AWS Batch

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la consola AWS Batch
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear AWS Batch recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso.

Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola AWS Batch

Para acceder a la AWS Batch consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los AWS Batch recursos de su cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos

necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

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

Para garantizar que los usuarios y los roles puedan seguir utilizando la AWS Batch consola, asocie también la AWS Batch `ConsoleAccess` política `ReadOnly` AWS gestionada a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
```

```
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición `aws:SourceAccount`global `aws:SourceArn`y las claves de contexto en las políticas de recursos para limitar los permisos que se AWS Batch otorgan a otro servicio al recurso. Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un ARN de bucket de Amazon S3, debe utilizar ambas claves de contexto de condición global para limitar los permisos. Si utiliza claves de contexto de condición global y el valor de `aws:SourceArn` contiene el ID de cuenta, el valor de `aws:SourceAccount` y la cuenta en el valor de `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utiliza en la misma instrucción de política. Utilice `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utilice `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

El valor de `aws:SourceArn` debe ser el recurso que se AWS Batch almacena.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si está especificando varios recursos, utilice la clave de condición de

contexto global `aws:SourceArn` con caracteres comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:servicename:*:123456789012:*`.

En los siguientes ejemplos se muestra cómo utilizar las claves de contexto de condición `aws:SourceAccount` global `aws:SourceArn` y las claves contextuales AWS Batch para evitar el confuso problema de los diputados.

Ejemplo 1: Rol para acceder a un solo entorno de computación

El siguiente rol solo se puede usar para acceder a un entorno de computación. El nombre del trabajo debe especificarse como * porque la cola de trabajos se puede asociar a varios entornos de computación.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/testCE",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Ejemplo 2: Rol para acceder a varios entornos de computación

El siguiente rol se puede utilizar para tener acceso a varios entornos de computación. El nombre del trabajo debe especificarse como * porque la cola de trabajos se puede asociar a varios entornos de computación.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/*",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Solución de problemas AWS Batch de identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas habituales que pueden surgir al trabajar con un AWS Batch IAM.

Temas

- No tengo autorización para realizar una acción en AWS Batch
- No estoy autorizado a realizar iam: PassRole
- Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS Batch recursos

No tengo autorización para realizar una acción en AWS Batch

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

En el siguiente ejemplo, el error se produce cuando el usuario `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `batch:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 batch:GetWidget on resource: my-example-widget
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *`my-example-widget`* mediante la acción `batch:`*`GetWidget`*. Para obtener más información sobre la concesión de permisos para transferir un rol, consulte Conceder permisos a un usuario para transferir un rol a un AWS servicio.

No estoy autorizado a realizar iam: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a AWS Batch.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS Batch. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS Batch recursos

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que

asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si AWS Batch es compatible con estas funciones, consulte¿Cómo AWS Batch funciona con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Uso de funciones vinculadas a servicios para AWS Batch

AWS Batch usa roles vinculados al AWS Identity and Access Management servicio (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. AWS Batch Los roles vinculados al servicio están predefinidos AWS Batch e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a un servicio facilita la configuración AWS Batch , ya que no es necesario añadir manualmente los permisos necesarios. AWS Batch define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS Batch puede asumir sus funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

> **Note**
>
> Realice una de las siguientes acciones para especificar una función de servicio para un entorno AWS Batch informático.
>
> - Utilice una cadena vacía para el rol de servicio. Esto permite AWS Batch crear el rol de servicio.
> - Especifique el rol de servicio en el siguiente formato: `arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.
>
> Para obtener más información, consulte the section called “Nombre de rol o ARN incorrectos” la Guía AWS Batch del usuario.

Solo puede eliminar un rol vinculado a servicios después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de AWS Batch , ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

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

Permisos de rol vinculados al servicio para AWS Batch

AWS Batch usa el rol vinculado al servicio denominado. AWSServiceRoleForBatch El AWSServiceRoleForBatchrol permite AWS Batch crear y administrar AWS recursos en su nombre.

El rol AWSServiceRoleForBatchvinculado al servicio confía en que el director del `batch.amazonaws.com` servicio asuma el rol.

La política de IAM denominada BatchServiceRolePolicypermite AWS Batch realizar las siguientes acciones en recursos específicos:

- `autoscaling`— Permite AWS Batch crear y gestionar los recursos de Auto Scaling de Amazon EC2. AWS Batch crea y administra grupos de Auto Scaling de Amazon EC2 para la mayoría de los entornos de cómputo.

- `ec2`— Permite AWS Batch controlar el ciclo de vida de las instancias de Amazon EC2, así como crear y gestionar plantillas y etiquetas de lanzamiento. AWS Batch crea y gestiona las solicitudes de EC2 Spot Fleet para algunos entornos de computación puntual de EC2.
- `ecs`- Permite AWS Batch crear y gestionar clústeres de Amazon ECS, definiciones de tareas y tareas para la ejecución de trabajos.
- `eks`- Permite AWS Batch describir el recurso del clúster Amazon EKS para las validaciones.
- `iam`- Permite AWS Batch validar y transferir las funciones proporcionadas por el propietario a Amazon EC2, Amazon EC2 Auto Scaling y Amazon ECS.
- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch los trabajos.

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

Crear un rol vinculado a un servicio para AWS Batch

No necesita crear manualmente un rol vinculado a servicios. Cuando estás CreateComputeEnvironment en la AWS Management Console AWS CLI, la o la AWS API y no especificas un valor para el `serviceRole` parámetro, se AWS Batch crea automáticamente el rol vinculado al servicio.

> **Important**
>
> Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Además, si utilizabas el AWS Batch servicio antes del 10 de marzo de 2021, cuando comenzó a admitir roles vinculados al servicio, entonces AWS Batch creaste el AWSServiceRoleForBatch rol en tu cuenta. Para obtener más información, consulte Un nuevo rol ha aparecido en mi cuenta de IAM.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando lo haces CreateComputeEnvironment, vuelve a AWS Batch crear el rol vinculado al servicio para ti.

Edición de un rol vinculado a un servicio para AWS Batch

Con AWS Batch, no puedes editar el rol vinculado al AWSServiceRoleForBatch servicio. Después de crear un rol vinculado a un servicio, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia a él. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a un servicio en la Guía del usuario de IAM..

Para permitir que una entidad de IAM edite la descripción del rol vinculado al servicio AWSServiceRoleForBatch

Añada la siguiente instrucción a la política de permisos. Esto permite a una entidad de IAM editar la descripción del rol vinculado a servicio.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
    "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Eliminar un rol vinculado a un servicio para AWS Batch

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

Para permitir que una entidad de IAM elimine el rol vinculado al servicio AWSServiceRoleForBatch

Añada la siguiente instrucción a la política de permisos. Esto permite a una entidad de IAM eliminar un rol vinculado a un servicio.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
```

```
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
    "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Saneamiento de un rol vinculado a servicios

Antes de poder usar IAM para eliminar un rol vinculado a un servicio, primero debe confirmar que el rol no tiene sesiones activas y eliminar todos los entornos de AWS Batch procesamiento que utilizan el rol en todas AWS las regiones en una sola partición.

Para comprobar si el rol vinculado a servicio tiene una sesión activa

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Funciones y, a continuación, el AWSServiceRoleForBatch nombre (no la casilla de verificación).
3. En la página Summary, elija Access Advisor y revise la actividad reciente del rol vinculado a servicio.

> **Note**
>
> Si no sabe si AWS Batch está utilizando el AWSServiceRoleForBatch rol, puede intentar eliminarlo. Si el servicio está utilizando el rol, este no podrá eliminarse. Puede ver las regiones en las que se está utilizando el rol. Si el rol se está utilizando, debe esperar que la sesión finalice para poder eliminarlo. No se puede revocar la sesión de un rol vinculado a un servicio.

Para eliminar AWS Batch los recursos utilizados por el rol AWSServiceRoleForBatch vinculado al servicio

Debe eliminar todos los entornos AWS Batch informáticos que utilizan el AWSServiceRoleForBatch rol en todas AWS las regiones antes de poder eliminar el AWSServiceRoleForBatch rol.

1. Abra la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, seleccione la región a utilizar.
3. En el panel de navegación, elija Entornos de computación.

4. Seleccione el entorno de computación.
5. Elija Deshabilitar. Espere a que Estado cambie a DESHABILITADO.
6. Seleccione el entorno de computación.
7. Elija Eliminar. Confirme que desea eliminar el entorno de computación mediante la elección de Eliminar entorno de computación.
8. Repita los pasos 1 a 7 para todos los entornos de computación que utilizan la función vinculada al servicio en todas las regiones.

Eliminación de un rol vinculado a un servicio en IAM (Consola)

Puede utilizar la consola de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la consola de IAM, elija Roles. A continuación, active la casilla de verificación situada junto a ella AWSServiceRoleForBatch, no el nombre o la fila en sí.
3. Elija Eliminar rol.
4. En el cuadro de diálogo de confirmación, revise los datos del último acceso al servicio, que muestra cuándo cada uno de los roles seleccionados tuvo acceso a un Servicio de AWS por última vez. Esto lo ayuda a confirmar si el rol está actualmente activo. Si desea continuar, seleccione Yes, Delete para enviar la solicitud de eliminación del rol vinculado al servicio.
5. Consulte las notificaciones de la consola de IAM para monitorear el progreso de la eliminación del rol vinculado al servicio. Como el proceso de eliminación del rol vinculado al servicio de IAM es asíncrono, dicha tarea puede realizarse correctamente o fallar después de que envía la solicitud de eliminación.

    - Si la tarea se realiza correctamente, el rol se elimina de la lista y aparece una notificación informando de ello en la parte superior de la página.
    - Si la tarea no se realiza correctamente, puede seleccionar View details (Ver detalles) o View Resources (Ver recursos) desde las notificaciones para obtener información sobre el motivo por el que no se pudo eliminar el rol. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá limpiar los recursos y volver a enviar la solicitud de eliminación.

> **Note**
>
> Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo, se producirá un error y el servicio informará del recurso restante. Un servicio podría informar de todos los recursos, algunos o ninguno.

- Si se produce un error en la tarea y la notificación no incluye una lista de los recursos, el servicio no podría no devolver dicha información. Para obtener información acerca de cómo limpiar los recursos de ese servicio, consulte Servicios de AWS que funcionan con IAM. Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Eliminación de un rol vinculado a un servicio en IAM (AWS CLI)

Puede utilizar los comandos de IAM de AWS Command Line Interface para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (CLI)

1. Como los roles vinculados a servicios no se pueden eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `deletion-task-id` de la respuesta para comprobar el estado de la tarea de eliminación. Ingrese el siguiente comando para enviar una solicitud de eliminación de un rol vinculado a un servicio:

   ```
   $ aws iam delete-service-linked-role --role-name AWSServiceRoleForBatch
   ```

2. Utilice el siguiente comando para comprobar el estado de la tarea de eliminación:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error

para que pueda resolver el problema. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá limpiar los recursos y volver a enviar la solicitud de eliminación.

> **Note**
>
> Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo, se producirá un error y el servicio informará del recurso restante. Un servicio podría devolver todos los recursos, algunos de ellos. O bien, podría no reportar ningún recurso. Para obtener información acerca de cómo limpiar los recursos de un servicio que no está informando de ningún recurso, consulte los Servicios de AWS que funcionan con IAM. Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Eliminación de un rol vinculado a servicio en IAM (API de AWS)

Puede utilizar la API de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (API)

1. Para enviar una solicitud de eliminación de una lista vinculada a un servicio, llama. DeleteServiceLinkedRole En la solicitud, especifique el nombre del AWSServiceRoleForBatch rol.

   Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `DeletionTaskId` de la respuesta para comprobar el estado de la tarea de eliminación.
2. Para comprobar el estado de la eliminación, llame GetServiceLinkedRoleDeletionStatus. En la solicitud, especifique el valor de `DeletionTaskId`.

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error

para que pueda resolver el problema. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá limpiar los recursos y volver a enviar la solicitud de eliminación.

> **Note**
>
> Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo, se producirá un error y el servicio informará del recurso restante. Un servicio podría informar de todos los recursos, algunos o ninguno. Para obtener información acerca de cómo limpiar los recursos de un servicio que no está informando de ningún recurso, consulte Servicios de AWS que funcionan con IAM. Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Regiones compatibles para funciones AWS Batch vinculadas al servicio

AWS Batch admite el uso de funciones vinculadas al servicio en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte Puntos de enlace de AWS Batch.

AWS políticas gestionadas para AWS Batch

Puedes usar políticas AWS administradas para simplificar la administración del acceso a la identidad de tu equipo y de AWS los recursos aprovisionados. AWS las políticas gestionadas cubren una variedad de casos de uso comunes, están disponibles de forma predeterminada en tu AWS cuenta y se mantienen y actualizan en tu nombre. No puedes cambiar los permisos en las políticas AWS gestionadas. Si necesita una mayor flexibilidad, también puede optar por crear políticas de IAM administradas por el cliente. De esta forma, puede proporcionar a su equipo los recursos aprovisionados solo con los permisos exactos que necesitan.

Para obtener más información sobre las políticas AWS administradas, consulte las políticas AWS administradas en la Guía del usuario de IAM.

AWS los servicios mantienen y AWS actualizan las políticas gestionadas en su nombre. Periódicamente, AWS los servicios añaden permisos adicionales a una política AWS gestionada. AWS Lo más probable es que las políticas gestionadas se actualicen cuando se lance una nueva función o cuando esté disponible una nueva operación. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Sin embargo, no eliminan los permisos ni anulan los permisos existentes.

Además, AWS admite políticas administradas para funciones laborales que abarcan varios servicios. Por ejemplo, la política `ReadOnlyAccess` AWS gestionada proporciona acceso de solo lectura a todos los AWS servicios y recursos. Cuando un servicio lanza una nueva función, AWS agrega permisos de solo lectura para nuevas operaciones y recursos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte Políticas administradas de AWS para funciones de trabajo en la Guía del usuario de IAM.

AWS política gestionada: BatchServiceRolePolicy

El BatchServiceRolePolicyrol vinculado al `AWSServiceRoleForBatch`servicio utiliza la política de IAM gestionada. Esto le permite AWS Batch realizar acciones en su nombre. No puede adjuntar esta política a sus entidades de IAM. Para obtener más información, consulte Uso de funciones vinculadas a servicios para AWS Batch.

Esta política permite AWS Batch realizar las siguientes acciones en recursos específicos:

- `autoscaling`— Permite AWS Batch crear y gestionar los recursos de Auto Scaling de Amazon EC2. AWS Batch crea y administra grupos de Auto Scaling de Amazon EC2 para la mayoría de los entornos de cómputo.
- `ec2`— Permite AWS Batch controlar el ciclo de vida de las instancias de Amazon EC2, así como crear y gestionar plantillas y etiquetas de lanzamiento. AWS Batch crea y gestiona las solicitudes de EC2 Spot Fleet para algunos entornos de computación puntual de EC2.
- `ecs`- Permite AWS Batch crear y gestionar clústeres de Amazon ECS, definiciones de tareas y tareas para la ejecución de trabajos.
- `eks`- Permite AWS Batch describir el recurso del clúster Amazon EKS para las validaciones.
- `iam`- Permite AWS Batch validar y transferir las funciones proporcionadas por el propietario a Amazon EC2, Amazon EC2 Auto Scaling y Amazon ECS.

- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch los trabajos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSBatchPolicyStatement1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeImages",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSpotFleetRequestHistory",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:RequestSpotFleet",
                "autoscaling:DescribeAccountLimits",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "eks:DescribeCluster",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:DescribeTaskDefinition",
                "ecs:DescribeTasks",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:ListTaskDefinitionFamilies",
                "ecs:ListTaskDefinitions",
                "ecs:ListTasks",
```

```
                "ecs:DeregisterTaskDefinition",
                "ecs:TagResource",
                "ecs:ListAccountSettings",
                "logs:DescribeLogGroups",
                "iam:GetInstanceProfile",
                "iam:GetRole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AWSBatchPolicyStatement2",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*"
        },
        {
            "Sid": "AWSBatchPolicyStatement3",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*:log-stream:*"
        },
        {
            "Sid": "AWSBatchPolicyStatement4",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateOrUpdateTags"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement5",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
```

```
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn",
                        "ecs-tasks.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement6",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "autoscaling.amazonaws.com",
                        "ecs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement7",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement8",
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:TerminateInstances",
                "ec2:CancelSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "ec2:DeleteLaunchTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement9",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteLaunchConfiguration"
            ],
            "Resource":
"arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement10",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:SuspendProcesses",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:TerminateInstanceInAutoScalingGroup"
            ],
            "Resource":
"arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement11",
            "Effect": "Allow",
            "Action": [
                "ecs:DeleteCluster",
                "ecs:DeregisterContainerInstance",
```

```
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:cluster/AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement12",
            "Effect": "Allow",
            "Action": [
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:task-definition/*"
        },
        {
            "Sid": "AWSBatchPolicyStatement13",
            "Effect": "Allow",
            "Action": [
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:task/*/*"
        },
        {
            "Sid": "AWSBatchPolicyStatement14",
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
                "ecs:RegisterTaskDefinition"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement15",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/*",
```

```
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:launch-template/*",
                "arn:aws:ec2:*:*:placement-group/*",
                "arn:aws:ec2:*:*:capacity-reservation/*",
                "arn:aws:ec2:*:*:elastic-gpu/*",
                "arn:aws:elastic-inference:*:*:elastic-inference-accelerator/*",
                "arn:aws:resource-groups:*:*:group/*"
            ]
        },
        {
            "Sid": "AWSBatchPolicyStatement16",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement17",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "RunInstances",
                        "CreateLaunchTemplate",
                        "RequestSpotFleet"
                    ]
                }
            }
        }
```

```
    ]
}
```

AWS política gestionada: AWSBatchServiceRolepolítica

La política de permisos de roles denominada AWSBatchServiceRole AWS Batch permite realizar las siguientes acciones en recursos específicos:

La política de IAM AWSBatchServiceRolegestionada suele ser utilizada por un rol denominado AWSBatchServiceRolee incluye los siguientes permisos. Siguiendo el consejo de seguridad estándar de conceder el mínimo privilegio, la política AWSBatchServiceRolegestionada se puede utilizar como guía. Si alguno de los permisos que se conceden en la política administrada no resulta necesario para su caso de uso, cree una política personalizada y agregue solo los permisos que necesite. Esta política y esta función AWS Batch gestionadas se pueden utilizar con la mayoría de los tipos de entornos informáticos, pero se prefiere el uso de funciones vinculadas al servicio para disfrutar de una experiencia less-error-prone gestionada mejor y con un alcance mejor.

- `autoscaling`— Permite AWS Batch crear y gestionar los recursos de Auto Scaling de Amazon EC2. AWS Batch crea y administra grupos de Auto Scaling de Amazon EC2 para la mayoría de los entornos de cómputo.
- `ec2`— Permite AWS Batch gestionar el ciclo de vida de las instancias de Amazon EC2, así como crear y gestionar plantillas y etiquetas de lanzamiento. AWS Batch crea y gestiona las solicitudes de EC2 Spot Fleet para algunos entornos de computación puntual de EC2.
- `ecs`- Permite AWS Batch crear y gestionar clústeres de Amazon ECS, definiciones de tareas y tareas para la ejecución de trabajos.
- `iam`- Permite AWS Batch validar y transferir las funciones proporcionadas por el propietario a Amazon EC2, Amazon EC2 Auto Scaling y Amazon ECS.
- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch los trabajos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSBatchPolicyStatement1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
```

```
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeImages",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSpotFleetRequestHistory",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:CreateLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:RequestSpotFleet",
                "ec2:CancelSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "ec2:TerminateInstances",
                "ec2:RunInstances",
                "autoscaling:DescribeAccountLimits",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:SuspendProcesses",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:DescribeTaskDefinition",
                "ecs:DescribeTasks",
                "ecs:ListAccountSettings",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
```

```
                "ecs:ListTaskDefinitionFamilies",
                "ecs:ListTaskDefinitions",
                "ecs:ListTasks",
                "ecs:CreateCluster",
                "ecs:DeleteCluster",
                "ecs:RegisterTaskDefinition",
                "ecs:DeregisterTaskDefinition",
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:UpdateContainerAgent",
                "ecs:DeregisterContainerInstance",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogGroups",
                "iam:GetInstanceProfile",
                "iam:GetRole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AWSBatchPolicyStatement2",
            "Effect": "Allow",
            "Action": "ecs:TagResource",
            "Resource": [
                "arn:aws:ecs:*:*:task/*_Batch_*"
            ]
        },
        {
            "Sid": "AWSBatchPolicyStatement3",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn",
                        "ecs-tasks.amazonaws.com"
                    ]
                }
```

```
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement4",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "autoscaling.amazonaws.com",
                        "ecs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement5",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "RunInstances"
                }
            }
        }
    ]
}
```

AWS política gestionada: AWSBatchFullAccess

La AWSBatchFullAccesspolítica otorga a AWS Batch las acciones pleno acceso a AWS Batch los recursos. También otorga acceso a las acciones de descripción y lista para los servicios de Amazon EC2, Amazon ECS CloudWatch, Amazon EKS e IAM. Esto permite que las identidades de IAM, ya sean usuarios o roles, puedan ver los recursos AWS Batch administrados que se crearon en

su nombre. Por último, esta política también permite transferir determinados roles de IAM a esos servicios.

Puede adjuntarlos AWSBatchFullAccessa sus entidades de IAM. AWS Batch también vincula esta política a un rol de servicio que le permite AWS Batch realizar acciones en su nombre.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource":"*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:PassRole"
      ],
      "Resource":[
        "arn:aws:iam::*:role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
```

```
        "arn:aws:iam::*:role/ecsInstanceRole",
        "arn:aws:iam::*:instance-profile/ecsInstanceRole",
        "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/AWSBatchJobRole*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"arn:aws:iam::*:role/*Batch*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "batch.amazonaws.com"
          }
      }
    }
  ]
}
```

AWS Batch actualizaciones de las políticas AWS gestionadas

Vea los detalles sobre las actualizaciones de las políticas AWS administradas AWS Batch desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase a la fuente RSS de la página del historial del AWS Batch documento.

Cambio	Descripción	Fecha
BatchServiceRolePolicypolítica actualizada	Se actualizó para añadir soporte a la descripción del historial de solicitudes y Amazon EC2 Auto Scaling las actividades de Spot Fleet.	5 de diciembre de 2023
AWSBatchServiceRolepolítica agregada	Se actualizó para agregar identificadores de estados de cuenta, otorgar AWS Batch permisos a `ec2:DescribeSpotFl`	5 de diciembre de 2023

Cambio	Descripción	Fecha
	`eetRequestHistory yautoscaling:DescribeScalingActivities .`	
BatchServiceRolePolicypolítica actualizada	Se actualizó para añadir compatibilidad con la descripción de los clústeres de Amazon EKS.	20 de octubre de 2022
AWSBatchFullAccesspolítica actualizada	Se actualizó para añadir compatibilidad con el listado y la descripción de los clústeres de Amazon EKS.	20 de octubre de 2022
BatchServiceRolePolicypolítica actualizada	Se actualizó para añadir compatibilidad con los grupos de reserva de capacidad de Amazon EC2 gestionados por AWS Resource Groups. Para más información, consulte Trabajar con grupos de reservas de capacidad en la Guía del usuario de Amazon EC2 para instancias de Linux.	18 de mayo de 2022
BatchServiceRolePolicyy AWSBatchServiceRolepolíticas actualizadas	Se ha actualizado para añadir compatibilidad con la descripción del estado de las instancias AWS Batch gestionadas en Amazon EC2, de forma que se sustituyan las instancias en mal estado.	6 de diciembre de 2021
BatchServiceRolePolicypolítica actualizada	Se actualizó para añadir compatibilidad con los recursos de grupos de ubicación , reserva de capacidad, GPU elástica y Elastic Inference en Amazon EC2.	26 de marzo de 2021

Cambio	Descripción	Fecha
BatchServiceRolePolicypolítica agregada	Con la política BatchServiceRolePolicyadministrada para el rol AWSServiceRoleForBatchvinculado al servicio, puede usar un rol vinculado al servicio administrado por. AWS Batch Con esta política, no necesita mantener su propio rol para usarlo en sus entornos de computación.	10 de marzo de 2021
AWSBatchFullAccess- añadir permiso para añadir un rol vinculado al servicio	Añada permisos de IAM para poder añadir el rol AWSServiceRoleForBatchvinculado al servicio a la cuenta.	10 de marzo de 2021
AWS Batch comenzó a rastrear los cambios	AWS Batch comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	10 de marzo de 2021

Acceda AWS Batch mediante un punto final de interfaz

Puede usarlo AWS PrivateLink para crear una conexión privada entre su VPC y. AWS Batch Puede acceder a AWS Batch como si estuviera en su VPC, sin el uso de una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect . Las instancias de la VPC no necesitan direcciones IP públicas para acceder a AWS Batch.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a AWS Batch.

Para obtener más información, consulte Puntos de conexión de VPC en la AWS PrivateLink guía.

Consideraciones para AWS Batch

Antes de configurar un punto final de la interfaz AWS Batch, consulte las propiedades y limitaciones del punto final de la interfaz en la AWS PrivateLink Guía.

AWS Batch permite realizar llamadas a todas sus acciones de API a través del punto final de la interfaz.

Antes de configurar los puntos finales de la VPC de la interfaz AWS Batch, tenga en cuenta las siguientes consideraciones:

- Los trabajos que utilizan el tipo de lanzamiento de recursos de Fargate no requieren los puntos de enlace de la VPC de la interfaz para Amazon ECS, pero es posible que necesite los puntos de enlace de la VPC de la interfaz para Amazon AWS Batch ECR, Secrets Manager o Amazon Logs, que se describen en los puntos siguientes. CloudWatch
  - Para ejecutar trabajos, debe crear los puntos de conexión de VPC de interfaz para Amazon ECS. Para obtener más información, consulte Puntos de conexión de VPC de tipo interfaz (AWS PrivateLink PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.
  - Para permitir que sus tareas extraigan imágenes privadas de Amazon ECR, debe crear los puntos de conexión de VPC de interfaz de Amazon ECR. Para obtener más información, consulte Puntos de enlace de la VPC de interfaz (AWS PrivateLink) en la Guía del usuario de Amazon Elastic Container Registry.
  - Para permitir que sus tareas extraigan información confidencial de Secrets Manager, debe crear los puntos de conexión de VPC de interfaz de Secrets Manager. Para obtener más información, consulte Utilización de Secrets Manager con puntos de enlace de la VPC en la Guía del usuario de AWS Secrets Manager .
  - Si su VPC no tiene una puerta de enlace a Internet y sus trabajos utilizan el controlador de `awslogs` registro para enviar la información de registro a CloudWatch Logs, debe crear un punto de enlace de VPC de interfaz para Logs. CloudWatch Para obtener más información, consulte Uso de CloudWatch registros con puntos de enlace de VPC de interfaz en la Guía del usuario de Amazon CloudWatch Logs.
- Las tareas que utilizan el tipo de lanzamiento de EC2 requieren que las instancias de contenedor en las que se lanzan ejecuten la versión `1.25.1` o posterior del agente de contenedor de Amazon ECS. Para obtener más información, consulte Versiones del agente de contenedores de Linux en Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.
- Los puntos de enlace de la VPC no admiten las solicitudes entre regiones. Asegúrese de crear su punto de conexión en la misma región en la que tiene previsto enviar llamadas a la API de AWS Batch.
- Los puntos de conexión de VPC solo admiten DNS proporcionadas por Amazon a través de Amazon Route 53. Si desea utilizar su propio DNS, puede utilizar el enrutamiento de DNS

condicional. Para obtener más información, consulte Conjuntos de opciones de DHCP en la Guía del usuario de Amazon VPC.

- El grupo de seguridad asociado al punto de conexión de la VPC debe permitir las conexiones entrantes en el puerto 443 desde la subred privada de la VPC.
- AWS Batch no admite los puntos finales de la interfaz de VPC en los siguientes casos: Regiones de AWS
  - Asia-Pacífico (Osaka) (`ap-northeast-3`)
  - Asia-Pacífico (Yakarta) (`ap-southeast-3`)

Cree un punto final de interfaz para AWS Batch

Puede crear un punto final de interfaz para AWS Batch usar la consola de Amazon VPC o AWS Command Line Interface ()AWS CLI. Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

Cree un punto final de interfaz para AWS Batch usar el siguiente nombre de servicio:

```
com.amazonaws.region.batch
```

Por ejemplo:

```
com.amazonaws.us-east-2.batch
```

En la partición `aws-cn`, el formato es diferente:

```
cn.com.amazonaws.region.batch
```

Por ejemplo:

```
cn.com.amazonaws.cn-northwest-1.batch
```

Si habilita el DNS privado para el punto final de la interfaz, puede realizar solicitudes de API AWS Batch utilizando su nombre de DNS regional predeterminado. Por ejemplo, `batch.us-east-1.amazonaws.com`.

Para obtener más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía AWS PrivateLink .

Creación de una política de punto de conexión para el punto de conexión de interfaz

Una política de punto de conexión es un recurso de IAM que puede adjuntar al punto de conexión de su interfaz. La política de punto final predeterminada permite el acceso total a AWS Batch través del punto final de la interfaz. Para controlar el acceso permitido a AWS Batch desde la VPC, adjunte una política de puntos de conexión personalizada al punto de conexión de interfaz.

Una política de punto de conexión especifica la siguiente información:

- Las entidades principales (Cuentas de AWS, usuarios y roles de IAM) que puede realizar acciones
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

Para obtener más información, consulte Control del acceso a los servicios con políticas de punto de conexión en la Guía del usuario de AWS PrivateLink .

Ejemplo: política de puntos finales de VPC para acciones AWS Batch

El siguiente es un ejemplo de una política de un punto de conexión personalizado. Al adjuntar esta política al punto final de la interfaz, se concede acceso a las AWS Batch acciones enumeradas a todos los principales de todos los recursos.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "batch:SubmitJob",
            "batch:ListJobs",
            "batch:DescribeJobs"
         ],
         "Resource":"*"
      }
   ]
}
```

Validación de conformidad para AWS Batch

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

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

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

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

  > **Note**
  >
  > No Servicios de AWS todas cumplen con los requisitos de la HIPAA. Para más información, consulte la Referencia de servicios compatibles con HIPAA.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde la perspectiva del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.

- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS consumo para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Seguridad de la infraestructura en AWS Batch

Como servicio gestionado, AWS Batch está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a AWS Batch través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puede llamar a estas operaciones de API desde cualquier ubicación de la red, pero AWS Batch admite políticas de acceso basadas en los recursos, que pueden incluir restricciones basadas en la dirección IP de origen. También puede usar AWS Batch políticas para controlar el acceso desde puntos de enlace específicos de Amazon Virtual Private Cloud (Amazon VPC) o VPC específicas. En efecto, esto aísla el acceso a la red a un AWS Batch recurso determinado únicamente de la VPC específica de la red. AWS

Etiquetado de los recursos de AWS Batch

Para ayudarle a administrar sus recursos de AWS Batch, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. En este tema se describe qué son las etiquetas y cómo crearlas.

Contenido

Conceptos básicos de etiquetas

Una etiqueta es una marca que se asigna a un recurso de AWS. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario.

Las etiquetas le permiten clasificar los recursos de AWS según, por ejemplo, su finalidad, propietario o entorno. Cuando tenga muchos recursos del mismo tipo, puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. Por ejemplo, puede definir un conjunto de etiquetas para los servicios de AWS Batch para ayudarle a realizar un seguimiento del propietario y del nivel de pila de cada servicio. Le recomendamos que diseñe un conjunto coherente de claves de etiqueta para cada tipo de recurso.

Además, las etiquetas no se asignan a los recursos automáticamente. Después de agregar una etiqueta, puede editar las claves y los valores de las etiquetas o eliminar etiquetas de un recurso en cualquier momento. Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.

Las etiquetas no tienen ningún significado semántico para AWS Batch, por lo que se interpretan estrictamente como cadenas de caracteres. Puede establecer el valor de una etiqueta como una cadena vacía, pero no puede asignarle un valor nulo. Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo.

Puede trabajar con etiquetas utilizando la AWS Management Console, la AWS CLI y la API de AWS Batch.

Si utiliza AWS Identity and Access Management (IAM), puede controlar qué usuarios de su cuenta de AWS tienen permiso para crear, editar o eliminar etiquetas.

Etiquetado de los recursos de

Puede etiquetar entornos informáticos de AWS Batch, trabajos, definiciones de trabajos, colas de trabajos y políticas de programación nuevos o existentes.

Si utiliza la consola de AWS Batch, puede aplicar etiquetas a los recursos de nueva creación o a los recursos existentes cuando lo desee mediante la pestaña Etiquetas en la página de recursos en cuestión.

Si utiliza la API de AWS Batch, la AWS CLI o un SDK de AWS, puede aplicar etiquetas a los recursos nuevos mediante el parámetro de `tags` en la acción de la API pertinente o utilizar la acción de la API de `TagResource` para aplicar etiquetas a los recursos existentes. Para obtener más información, consulte TagResource.

Además, algunas acciones de creación de recursos le permiten especificar etiquetas para un recurso al crearlo. Si no se pueden aplicar etiquetas durante la creación del recurso, el proceso de creación de recursos falla. Esto garantiza que los recursos que pretendía etiquetar en el momento de su creación se creen con etiquetas específicas o no se creen en absoluto. Si etiqueta recursos en el momento de su creación, no es necesario ejecutar scripts de etiquetado personalizados después de la creación del recurso.

En la tabla siguiente se describen los recursos de AWS Batch que se pueden etiquetar y aquellos que se pueden etiquetar en el momento de su creación.

Compatibilidad con el etiquetado de recursos de AWS Batch

Recurso	Admite etiquetas	Admite la propagación de etiquetas	Admite el etiquetado durante la creación (API de AWS Batch, AWS CLI y SDK de AWS)
Entornos informáticos AWS Batch	Sí	No. Las etiquetas del entorno informático no se propagan a ningún otro recurso. Las etiquetas de los	Sí

Recurso	Admite etiquetas	Admite la propagación de etiquetas	Admite el etiquetado durante la creación (API de AWS Batch, AWS CLI y SDK de AWS)
		recursos se especifican en el miembro tags del objeto ComputeResources pasado en la operación de API CreateComputeEnvironment.	
Trabajos de AWS Batch	Sí	Sí	Sí
Definiciones de trabajo de AWS Batch	Sí	No	Sí
Colas de trabajo de AWS Batch	Sí	No	Sí
Políticas de programación de AWS Batch	Sí	No	Sí

Restricciones de las etiquetas

Se aplican las siguientes restricciones básicas a las etiquetas:

- Número máximo de etiquetas por recurso: 50
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.
- Longitud máxima de la clave: 128 caracteres Unicode en UTF-8
- Longitud máxima del valor: 256 caracteres Unicode en UTF-8
- Si se utiliza su esquema de etiquetado en múltiples servicios y recursos de AWS, recuerde que otros servicios pueden tener restricciones sobre caracteres permitidos. Los caracteres permitidos

generalmente son: letras, números y espacios representables en UTF-8, además de los siguientes caracteres: + - = . _ : / @.

- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No utilice `aws:`, `AWS:` ni ninguna combinación de mayúsculas o minúsculas del mismo como prefijo para claves o valores, ya que está reservado para uso de AWS. Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas que tengan este prefijo no cuentan para el límite de etiquetas por recurso.

Uso de etiquetas mediante la consola

Mediante la consola de AWS Batch, puede gestionar las etiquetas asociadas a los entornos informáticos, los trabajos, las definiciones de trabajos y las colas de trabajos nuevos o existentes.

Adición de etiquetas a un recurso individual durante su creación

Puede añadir etiquetas a los entornos informáticos de AWS Batch, los trabajos, las definiciones de trabajos, las colas de trabajos y las políticas de programación al crearlos.

Adición y eliminación de etiquetas en un recurso individual

AWS Batch le permite añadir o eliminar etiquetas asociadas a sus clústeres directamente desde la página del recurso.

Para agregar o eliminar una etiqueta en un recurso individual

1. Abra la consola de AWS Batch en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, elija la región a utilizar.
3. En el panel de navegación, elija un tipo de recurso (por ejemplo, Colas de trabajos).
4. Elija un recurso específico y, a continuación, elija Editar etiquetas.
5. Agregue o elimine sus etiquetas según sea necesario.

   - Para añadir una etiqueta, especifique la clave y el valor en los cuadros de texto vacíos al final de la lista.
   - Para eliminar una etiqueta, seleccione el botón

     Delete icon
     junto a la etiqueta.

6. Repita este proceso para cada etiqueta que desee agregar o eliminar y, a continuación, elija Editar etiquetas para finalizar.

Uso de etiquetas mediante la CLI o la API

Utilice los siguientes comandos de AWS CLI u operaciones de la API de AWS Batch para agregar, actualizar, enumerar y eliminar las etiquetas de sus recursos.

Compatibilidad con el etiquetado de recursos de AWS Batch

Tarea	Acción de la API	AWS CLI	AWS Tools for Windows PowerShell
Agregar o sobrescribir una o varias etiquetas.	TagResource	tag-resource	Add-BATResourceTag
Eliminar una o varias etiquetas.	UntagResource	untag-resource	Remove-BATResourceTag
Enumera las etiquetas de un recurso	ListTagsForResource	list-tags-for-resource	Get-BATResourceTag

Los siguientes ejemplos muestran cómo agregar o quitar etiquetas a los recursos mediante la AWS CLI.

Ejemplo 1: Etiquetar un recurso existente

El siguiente comando etiqueta un recurso existente.

```
aws batch tag-resource --resource-arn resource_ARN --tags team=devs
```

Ejemplo 2: Eliminar la etiqueta de un recurso existente

El siguiente comando elimina una etiqueta de un recurso existente.

```
aws batch untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Ejemplo 3: enumerar etiquetas de un recurso

El siguiente comando enumera las etiquetas asociadas a un recurso existente.

```
aws batch list-tags-for-resource --resource-arn resource_ARN
```

Algunas acciones de creación de recursos le permiten especificar etiquetas al crear el recurso. Las siguientes acciones admiten etiquetado durante la creación.

Tarea	Acción de la API	AWS CLI	AWS Tools for Windows PowerShell
Crear un entorno informático	CreateComputeEnvironment	create-compute-environment	New-BATComputeEnvironment
Cear una cola de trabajos	CreateJobQueue	create-job-queue	New-BATJobQueue
Crear una política de programación	CreateSchedulingPolicy	create-scheduling-policy	New-BATSchedulingPolicy
Registrar una definición de trabajo	RegisterJobDefinition	register-job-definition	Register-BATJobDefinition
Enviar un trabajo	SubmitJob	submit-job	Submit-BAT Job

Cuotas de servicio de AWS Batch

En la tabla siguiente se indican las Service Quotas para AWS Batch que no pueden modificarse. Cada cuota es específica de la región.

Recurso	Cuota
Número máximo de colas de trabajo. Para obtener más información, consulte Colas de trabajo.	50
Número máximo de entornos informáticos en Amazon ECS y Amazon EKS. Para obtener más información, consulte Entorno de computación.	50
Número máximo de entornos informáticos por clúster de Amazon EKS.	5
Número máximo de entornos informáticos para cada cola de trabajos	3
Cantidad máxima de dependencias de un trabajo	20
Tamaño máximo de la definición del trabajo (para operaciones de la API `RegisterJobDefinition`)	24 KiB
El tamaño máximo de la carga del trabajo (para operaciones de API `SubmitJob`)	30 KiB
Tamaño máximo de matriz para los trabajos de matrices	10000
Cantidad máxima de trabajos en estado `SUBMITTED`	1000000
Número máximo de transacciones por segundo (TPS) para cada cuenta de operaciones `SubmitJob`	50

Dependiendo de cómo utilice AWS Batch, pueden aplicarse cuotas adicionales. Para obtener información sobre las cuotas de Amazon EC2, consulte Amazon EC2 Service Quotas en Referencia general de AWS. Para obtener más información acerca de las cuotas de servicio de Amazon ECS, consulte Amazon ECS Service Quotas en Referencia general de AWS. Para obtener más información acerca de las cuotas de servicio de Amazon EKS, consulte las Amazon EKS Service Quotas en Referencia general de AWS.

Solución de problemas AWS Batch

Es posible que necesite solucionar problemas relacionados con sus entornos de computación, colas de trabajos, definiciones de trabajos o trabajos. En este capítulo se describe cómo solucionar y resolver estos problemas en su entorno. AWS Batch

AWS Batch utiliza políticas, funciones y permisos de IAM y se ejecuta en la infraestructura de Amazon EC2, Amazon ECS y Amazon AWS Fargate Elastic Kubernetes Service. Para solucionar problemas relacionados con estos servicios, consulte lo siguiente:

- Solución de problemas de IAM en la Guía del usuario de IAM
- Solución de problemas de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service
- Solución de problemas de Amazon EKS en la Guía del usuario de Amazon EKS
- Solución de problemas para instancias de EC2 en la Guía del usuario de Amazon EC2 para instancias de Linux

Contenido

AWS Batch

Entorno de computación **`INVALID`**

Es posible que haya configurado incorrectamente un entorno de computación gestionado. Si lo ha hecho, el entorno de computación entra en un estado `INVALID` y no puede aceptar trabajos para ubicarlos. En las siguientes secciones se describen las posibles causas y cómo solucionar los problemas en función de la causa.

Nombre de rol o ARN incorrectos

La causa más común por la que un entorno informático entra en un `INVALID` estado es que el rol de AWS Batch servicio o el rol de Amazon EC2 Spot Fleet tienen un nombre o un nombre de recurso de Amazon (ARN) incorrectos. Esto es más común en los entornos informáticos que se crean con el AWS CLI o los AWS SDK. Cuando crea un entorno informático en AWS Management Console, le AWS Batch ayuda a elegir el servicio o las funciones correctas de Spot Fleet. Sin embargo, supongamos que introduce manualmente el nombre o el ARN y los introduce de forma incorrecta. Luego, el entorno de computación resultante también es `INVALID`.

Sin embargo, supongamos que introduce manualmente el nombre o el ARN de un recurso de IAM en un comando de AWS CLI o en el código del SDK. En este caso, no AWS Batch se puede validar la cadena. En su lugar, AWS Batch debe aceptar el valor incorrecto e intentar crear el entorno. Si AWS Batch no se crea el entorno, el entorno pasa a un `INVALID` estado y aparecen los siguientes errores.

Si se trata de un rol de servicio inválido:

```
CLIENT_ERROR - Not authorized to perform sts:AssumeRole (Service:
AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied;
Request ID: dc0e2d28-2e99-11e7-b372-7fcc6fb65fe7)
```

Si se trata de un rol de flota de spot no válido:

```
CLIENT_ERROR - Parameter: SpotFleetRequestConfig.IamFleetRole
is invalid. (Service: AmazonEC2; Status Code: 400; Error Code:
InvalidSpotFleetRequestConfig; Request ID: 331205f0-5ae3-4cea-
bac4-897769639f8d) Parameter: SpotFleetRequestConfig.IamFleetRole is
invalid
```

Una causa común de este problema es el siguiente escenario. Solo se especifica el nombre de una función de IAM cuando se utilizan el AWS CLI o los AWS SDK, en lugar del nombre completo del recurso de Amazon (ARN). En función de cómo se haya creado el rol de servicio, el ARN puede contener un prefijo de ruta `aws-service-role`. Por ejemplo, si crea manualmente el rol de servicio de AWS Batch utilizando los procedimientos de Uso de funciones vinculadas a servicios para AWS Batch, el ARN del rol de servicio debería ser similar al siguiente.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

Sin embargo, si creó el rol de servicio como parte del asistente de primera ejecución de la consola hoy, el ARN de su rol de servicio podría tener el siguiente aspecto.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Este problema también puede producirse si adjuntas la política de AWS Batch nivel de servicio (`AWSBatchServiceRole`) a una función que no sea de servicio. Por ejemplo, es posible que reciba un mensaje de error similar al siguiente en este escenario:

```
CLIENT_ERROR - User: arn:aws:sts::account_number:assumed-role/batch-replacement-role/
aws-batch is not
     authorized to perform: action on resource ...
```

Para resolver este problema, siga uno de estos pasos.

- Utilice una cadena vacía para la función de servicio al crear el AWS Batch entorno de procesamiento.

- Especifique el rol de servicio en el siguiente formato: `arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Si solo especificas el nombre de un rol de IAM al usar el AWS CLI o los AWS SDK, se AWS Batch supone que tu ARN no usa el prefijo de ruta. `aws-service-role` Por eso se recomienda especificar el ARN completo para los roles de IAM al crear entornos de computación.

Para reparar un entorno de computación configurado de forma incorrecta, consulte Cómo reparar un entorno de computación `INVALID`.

Cómo reparar un entorno de computación **`INVALID`**

Si el entorno de computación se encuentra en estado `INVALID`, actualícelo para reparar el parámetro no válido. Para un Nombre de rol o ARN incorrectos, actualice el entorno de computación con el rol de servicio correcto.

Para reparar un entorno de computación mal configurado

1. Abra la consola en https://console.aws.amazon.com/batch/ AWS Batch .
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. En la página Entornos de computación, marque el botón de opción que se encuentre junto al entorno de computación a editar y, a continuación, seleccione Editar.
5. En la página Actualizar entornos de computación, en Rol de servicio, elija el rol de IAM a utilizar con su entorno de computación. La consola AWS Batch solo muestra roles con la relación de confianza adecuada para entornos de computación.
6. Seleccione Guardar para actualizar el entorno de computación.

Trabajos bloqueados en estado **`RUNNABLE`**

Suponga que su entorno de computación contiene recursos de computación, pero sus trabajos no progresan más allá del estado `RUNNABLE`. Entonces, es probable que algo impida que los trabajos se coloquen en un recurso informático y que se bloqueen las colas de trabajos. A continuación, te explicamos cómo saber si tu trabajo está esperando su turno o si está atascado y bloqueando la cola.

Si AWS Batch detecta que tienes un RUNNABLE trabajo a la cabeza y estás bloqueando la cola, recibirás un evento de Amazon CloudWatch Events sobre el bloqueo de la cola de trabajos con el motivo. El mismo motivo también se actualiza en el `statusReason` campo como parte de las llamadas a `ListJobs` la `DescribeJobs` API.

Si lo desea, puede configurar el `jobStateTimeLimitActions` parámetro mediante `CreateJobQueue` acciones `UpdateJobQueue`de API.

> **Note**
>
> Actualmente, la única acción que puede realizar `jobStateLimitActions.action` es cancelar un trabajo.

El `jobStateTimeLimitActions` parámetro se usa para especificar un conjunto de acciones que se AWS Batch realizan en trabajos en un estado específico. Puede establecer un umbral de tiempo en segundos a través del `maxTimeSeconds` campo.

Cuando un trabajo ha estado en un RUNNABLE estado con los definidos`statusReason`, AWS Batch realiza la acción especificada una vez `maxTimeSeconds` que hayan transcurrido.

Por ejemplo, puede configurar el `jobStateTimeLimitActions` parámetro para que espere hasta 4 horas para que cualquier trabajo en el RUNNABLE estado esté esperando a que haya suficiente capacidad disponible. `statusReason`Para ello, establezca el valor en 144000 `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY` y `maxTimeSeconds` antes de cancelar el trabajo y permitir que el siguiente pase a ocupar el primer lugar de la lista de trabajos.

Los siguientes son los motivos que se utilizan AWS Batch cuando detecta que una cola de trabajos está bloqueada. Esta lista proporciona los mensajes devueltos por las acciones de la `DescribeJobs` API `ListJobs` y las acciones de la API. También son los mismos valores que puede definir para el `jobStateLimitActions.statusReason` parámetro.

1. Motivo: todos los entornos informáticos conectados tienen errores de capacidad insuficientes. Cuando se solicita, AWS Batch detecta las instancias de Amazon EC2 que experimentan errores de capacidad insuficiente. Si se cancela el trabajo, ya sea manualmente o activando el `jobStateTimeLimitActions` parámetro`statusReason`, se permite que el trabajo siguiente pase al principio de la cola.

- **statusReason**mensaje mientras el trabajo está atascado: `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY - Service cannot fulfill the capacity requested for instance type [instanceTypeName]`
- **reason**utilizado para**jobStateTimeLimitActions**: `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY`
- **statusReason**mensaje después de cancelar el trabajo: `Canceled by JobStateTimeLimit action due to reason: CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY`

Nota:

a. El rol AWS Batch de servicio requiere `autoscaling:DescribeScalingActivities` permiso para que esta detección funcione. Si utilizas el rol AWSServiceRoleForBatchvinculado al servicio (SLR) o la política AWSBatchServiceRolePolicyadministrada, no necesitas realizar ninguna acción porque sus políticas de permisos están actualizadas.

b. Si utilizas la SLR o la política gestionada, debes añadir los `ec2:DescribeSpotFleetRequestHistory` permisos `autoscaling:DescribeScalingActivities` y para poder recibir los eventos de las colas de trabajos bloqueadas y el estado actualizado de los trabajos cuando estés dentro. `RUNNABLE` Además, AWS Batch necesita estos permisos para realizar `cancellation` acciones a través del `jobStateTimeLimitActions` parámetro, incluso si están configurados en la cola de trabajos.

c. En el caso de un trabajo paralelo de varios nodos (MNP), si el entorno informático Amazon EC2 de alta prioridad adjunto experimenta `insufficient capacity` errores, bloquea la cola incluso si un entorno informático de prioridad inferior sufre este error.

2. Motivo: todos los entornos informáticos tienen un `maxvCpus`parámetro inferior a los requisitos del trabajo. La cancelación del trabajo, ya sea manualmente o activando el `jobStateTimeLimitActions` parámetro`statusReason`, permite que el trabajo siguiente pase al principio de la lista. Si lo desea, puede aumentar el `maxvCpus` parámetro del entorno informático principal para satisfacer las necesidades del trabajo bloqueado.

   - **statusReason**mensaje mientras el trabajo está atascado: `MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE - CE(s) associated with the job queue cannot meet the CPU requirement of the job.`
   - **reason**utilizado para**jobStateTimeLimitActions**: `MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE`

- **statusReason**mensaje después de cancelar el trabajo: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE`

3. Motivo: ninguno de los entornos de procesamiento tiene instancias que cumplan con los requisitos del trabajo. Cuando un trabajo solicita recursos, AWS Batch detecta que ningún entorno informático adjunto puede alojar el trabajo entrante. La cancelación del trabajo, ya sea manualmente o activando el `jobStateTimeLimitActions` parámetro`statusReason`, permite que el trabajo siguiente pase al principio de la cola. Si lo desea, puede redefinir los tipos de instancias permitidos en el entorno de cómputo para añadir los recursos de trabajo necesarios.

   - **statusReason**mensaje mientras el trabajo está atascado: `MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT - The job resource requirement (vCPU/memory/GPU) is higher than that can be met by the CE(s) attached to the job queue.`
   - **reason**utilizado para**jobStateTimeLimitActions**: `MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
   - **statusReason**mensaje después de cancelar el trabajo: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`

4. Motivo: todos los entornos informáticos tienen problemas con las funciones de servicio. Para resolver este problema, compare los permisos de su rol de servicio con los permisos del rol de servicio AWS Batch administrado y solucione cualquier brecha.

   Se recomienda utilizar la AWS Batch SLR en entornos informáticos para evitar errores similares.

   La cancelación del trabajo, ya sea manualmente o activando el `jobStateTimeLimitActions` parámetro`statusReason`, permite que el trabajo siguiente pase al principio de la lista. Si no se resuelven los problemas relacionados con las funciones de servicio, es probable que también se bloquee el siguiente trabajo. Lo mejor es investigar y resolver este problema manualmente.

   - **statusReason**mensaje mientras el trabajo está atascado: `MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS – Batch service role has a permission issue.`
   - **reason**utilizado para**jobStateTimeLimitActions**: `MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`

- **`statusReason`**mensaje después de cancelar el trabajo: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`

5. Motivo: todos los entornos informáticos no son válidos. Para obtener más información, consulte entorno de `INVALID` cómputo. Nota: No se puede configurar una acción programable a través del `jobStateTimeLimitActions` parámetro para resolver este error.
   - **`statusReason`**mensaje mientras el trabajo está atascado: `ACTION_REQUIRED - CE(s) associated with the job queue are invalid.`
6. Motivo: AWS Batch ha detectado una cola bloqueada, pero no ha podido determinar el motivo. Nota: No se puede configurar una acción programable a través del `jobStateTimeLimitActions` parámetro para resolver este error. Para obtener más información sobre la solución de problemas, consulta ¿Por qué mi AWS Batch trabajo está atascado en RUNNABLE? en AWS Re:post.
   - **`statusReason`**mensaje mientras el trabajo está atascado: `UNDETERMINED - Batch job is blocked, root cause is undetermined.`

En caso de que no hayas recibido un evento de CloudWatch Events o hayas recibido un evento con un motivo desconocido, estas son algunas de las causas más comunes de este problema.

El controlador de **`awslogs`** registro no está configurado en sus recursos informáticos

AWS Batch los trabajos envían su información de registro a CloudWatch Logs. Para activarlo, debe configurar sus recursos de computación para utilizar el controlador de registro `awslogs`. Supongamos que basa la AMI de sus recursos de computación en la AMI optimizada para Amazon ECS (o Amazon Linux). A continuación, este controlador se registra de forma predeterminada en el paquete `ecs-init`. Ahora supongamos que usa una AMI base diferente. Luego, debe verificar que el controlador de registro `awslogs` esté especificado como un controlador de registro disponible con la variable de entorno `ECS_AVAILABLE_LOGGING_DRIVERS` cuando se inicia el agente de contenedor de Amazon ECS. Para obtener más información, consulte Especificaciones de AMI de recursos de computación y Cómo crear una AMI de recursos informáticos.

Recursos insuficientes

Si sus definiciones de trabajo especifican más recursos de memoria o CPU de lo que pueden asignar los recursos de computación, los trabajos no se asignarán. Por ejemplo, supongamos que su trabajo especifica 4 GiB de memoria y que sus recursos de computación tienen menos de los

disponibles. Entonces, se da el caso de que el trabajo no se puede asignar a esos recursos de computación. En ese caso, debe reducir la memoria especificada en la definición del trabajo o añadir más recursos de computación en su entorno. Una parte de la memoria se reserva para el agente de contenedor de Amazon ECS y otros procesos críticos del sistema. Para obtener más información, consulte Administración de la memoria de los recursos informáticos de las .

No hay acceso a Internet para los recursos informáticos

Los recursos de computación de las deben obtener acceso para comunicarse con el punto de conexión del servicio de Amazon ECS. Esto puede ser a través de un punto de conexión de la VPC de la interfaz o a través de recursos de computación de las con direcciones IP públicas.

Para obtener más información acerca de los puntos de enlace de la VPC de la interfaz, consulte Puntos de enlace de la VPC de la interfaz de Amazon ECS (AWS PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.

Si no tiene configurado un punto de conexión de la VPC de la interfaz y los recursos de computación de las no tienen direcciones IP públicas, deberán utilizar traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulte Gateways NAT en la Guía del usuario de Amazon VPC. Para obtener más información, consulte the section called “Creación de una VPC”.

Se alcanzó el límite de instancias de Amazon EC2

La cantidad de instancias de Amazon EC2 en las que puede lanzar su cuenta Región de AWS viene determinada por su cuota de instancias EC2. Algunos tipos de instancias también tienen una per-instance-type cuota. Para obtener más información sobre la cuota de instancias de Amazon EC2 de su cuenta, incluida la forma de solicitar un aumento del límite, consulte los Límites de servicio de Amazon EC2 en la Guía del usuario de Amazon EC2 para instancias de Linux.

El agente de contenedores Amazon ECS no está instalado

El agente contenedor Amazon ECS debe estar instalado en la Imagen de máquina de Amazon (AMI) para permitir a AWS Batch ejecutar los trabajos. El agente de contenedor de Amazon ECS se instala de forma predeterminada en AMI optimizadas para Amazon ECS. Para obtener más información sobre el agente de contenedor de Amazon ECS, consulte Agente de contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

Para obtener más información, consulte ¿Por qué mi AWS Batch trabajo está RUNNABLE estancado? en Re:post.

Instancias de spot no etiquetadas en el momento de su creación

A partir del 25 de octubre de 2017, se admite el etiquetado de instancias puntuales para los recursos AWS Batch informáticos. Antes de esa fecha, la política administrada por IAM recomendada (`AmazonEC2SpotFleetRole`) para el rol de flota de spot de Amazon EC2 no contenía permisos para etiquetar las instancias de spot en el momento del lanzamiento. La nueva política administrada de IAM recomendada se denomina `AmazonEC2SpotFleetTaggingRole`. Admite el etiquetado de instancias de spot en el momento del lanzamiento.

Para corregir el etiquetado de las instancias de spot al crearlas, siga el siguiente procedimiento para aplicar la política gestionada de IAM actualmente recomendada a su rol de flota de spot de Amazon EC2. De esta forma, todas las instancias de spot futuras que se creen con ese rol tendrán permisos para aplicar etiquetas de instancia cuando se creen.

Para aplicar la política administrada de IAM actual al rol de flota de spot de Amazon EC2

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Elija Roles y elija el rol de flota de spot de Amazon EC2.
3. Elija Asociar política.
4. Seleccione AmazonEC2 SpotFleetTaggingRole y elija Adjuntar política.
5. Elija de nuevo el rol de flota de spot de Amazon EC2 para quitar la política anterior.
6. Seleccione la x situada a la derecha de la SpotFleetRole política de AmazonEC2 y elija Separar.

Las instancias de spot no se están reduciendo verticalmente

AWS Batch introdujo la función AWSServiceRoleForBatchvinculada al servicio el 10 de marzo de 2021. Si no se especifica ningún rol en el parámetro del entorno de computación de `serviceRole`, este rol vinculado al servicio se usa como rol de servicio. Sin embargo, supongamos que la función vinculada al servicio se utiliza en un entorno informático puntual de EC2, pero la función puntual utilizada no incluye la política gestionada de AmazonEC2. SpotFleetTaggingRole Entonces, la instancia de spot no se reduce verticalmente. Como resultado, recibirá un error con el siguiente mensaje: “No está autorizado a realizar esta operación”. Siga estos pasos para actualizar la función de flota de spot que utiliza en el parámetro `spotIamFleetRole`. Para obtener más información, consulte Uso de roles vinculados a servicios y Creación de un rol para delegar permisos a un AWS servicio en la Guía del usuario de IAM.

Temas

- Adjunte la política SpotFleetTaggingRole gestionada de AmazonEC2 a su función de Spot Fleet en el AWS Management Console
- Adjunte la política SpotFleetTaggingRole gestionada de AmazonEC2 a su función de Spot Fleet con la AWS CLI

Adjunte la política SpotFleetTaggingRole gestionada de AmazonEC2 a su función de Spot Fleet en el AWS Management Console

Para aplicar la política administrada de IAM actual al rol de flota de spot de Amazon EC2

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Elija Roles y elija el rol de flota de spot de Amazon EC2.
3. Elija Asociar política.
4. Seleccione AmazonEC2 SpotFleetTaggingRole y elija Adjuntar política.
5. Elija de nuevo el rol de flota de spot de Amazon EC2 para quitar la política anterior.
6. Seleccione la x situada a la derecha de la SpotFleetRole política de AmazonEC2 y elija Separar.

Adjunte la política SpotFleetTaggingRole gestionada de AmazonEC2 a su función de Spot Fleet con la AWS CLI

Los comandos de ejemplo asumen que su rol de Amazon EC2 Spot Fleet se denomina AmazonEC2. SpotFleetRole Si su rol usa un nombre diferente, ajuste los comandos para que coincidan.

Para asociar la política SpotFleetTaggingRole gestionada de AmazonEC2 a su función de Spot Fleet

1. Para adjuntar la política de IAM SpotFleetTaggingRole gestionada de AmazonEC2 a su SpotFleetRole función de *AmazonEC2*, ejecute el siguiente comando con la. AWS CLI

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole
 \
    --role-name AmazonEC2SpotFleetRole
```

2. Para separar la política de IAM SpotFleetRole gestionada de AmazonEC2 de su función de *SpotFleetRoleAmazonEC2*, ejecute el siguiente comando mediante. AWS CLI

```
$ aws iam detach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetRole \
    --role-name AmazonEC2SpotFleetRole
```

No puedo recuperar los secretos de Secrets Manager

Si utiliza una AMI con un agente de Amazon ECS anterior a la versión 1.16.0-1, debe utilizar la variable de configuración del agente de Amazon ECS `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` para utilizar esta característica. Puede agregarlo al archivo `./etc/ecs/ecs.config` a una nueva instancia de contenedor al crear esa instancia. O bien, puede añadirlo a una instancia existente. Si lo agrega a una instancia existente, debe reiniciar el agente de ECS después de agregarlo. Para obtener más información, consulte Configuración del agente de contenedor de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

No se pueden anular los requisitos de recursos de la definición del trabajo

Las anulaciones de memoria y vCPU que se especifican en `vcpus` los miembros `memory` y de la estructura ContainerOverrides, que se pasan a SubmitJob, no pueden anular los requisitos de memoria y vCPU que se especifican en la estructura ResourceRequirements de la definición del trabajo.

Si intenta anular estos requisitos de recursos, puede aparecer el siguiente mensaje de error:

“Este valor se envió en una clave obsoleta y puede entrar en conflicto con el valor proporcionado por los requisitos de recursos de la definición del trabajo”.

Para corregir esto, especifique los requisitos de memoria y vCPU en el miembro ResourceRequirements de ContainerOverrides. Por ejemplo, si las anulaciones de memoria y vCPU se especifican en las siguientes líneas.

```
"containerOverrides": {
   "memory": 8192,
   "vcpus": 4
}
```

Cámbielas a lo siguiente:

```
"containerOverrides": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "8192"
      },
      {
         "type": "VCPU",
         "value": "4"
      }
   ],
}
```

Realice el mismo cambio en los requisitos de memoria y vCPU que se especifican en el objeto ContainerProperties de la definición del trabajo. Por ejemplo, si los requisitos de memoria y vCPU se especifican en las siguientes líneas.

```
{
   "containerProperties": {
      "memory": 4096,
      "vcpus": 2,
}
```

Cámbielos a lo siguiente:

```
"containerProperties": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "4096"
      },
      {
         "type": "VCPU",
         "value": "2"
      }
   ],
}
```

Mensaje de error al actualizar la configuración de **desiredvCpus**

Aparece el siguiente mensaje de error cuando usa la AWS Batch API para actualizar la configuración de vCPU () `desiredvCpus` deseada.

`Manually scaling down compute environment is not supported. Disconnecting job queues from compute environment will cause it to scale-down to minvCpus.`

Este problema se produce si el valor actualizado `desiredvCpus` es inferior al valor actual `desiredvCpus`. Al actualizar el valor `desiredvCpus`, deben cumplirse las dos condiciones siguientes:

- El valor `desiredvCpus` debe estar entre los valores `minvCpus` y `maxvCpus`.
- El valor actualizado `desiredvCpus` debe ser igual o mayor que el valor actual `desiredvCpus`.

AWS Batch en Amazon EKS

Temas

Entorno de computación **INVALID**

Es posible que haya configurado incorrectamente un entorno de computación gestionado. Si lo ha hecho, el entorno de computación entra en un estado `INVALID` y no puede aceptar trabajos para ubicarlos. En las siguientes secciones se describen las posibles causas y cómo solucionar los problemas en función de la causa.

La versión del Kubernetes no es compatible

Es posible que aparezca un mensaje de error similar al siguiente cuando utilice la operación de `CreateComputeEnvironment` API o la operación de `UpdateComputeEnvironment` API para crear o actualizar un entorno de computación. Este problema se produce si especifica una versión de Kubernetes no compatible en `EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Para resolver este problema, elimine el entorno de procesamiento y vuelva a crearlo con una versión de Kubernetes compatible.

Puede realizar una actualización de una versión menor en su clúster de Amazon EKS. Por ejemplo, puede actualizar el clúster de `1.xx` a `1.yy` incluso si la versión secundaria no es compatible.

Sin embargo, es posible que el estado del entorno de computación cambie a `INVALID` después de una actualización de la versión principal. Por ejemplo, si realiza una actualización de una versión principal de `1.xx` a `2.yy`. Si la versión principal no es compatible con AWS Batch, aparecerá un mensaje de error similar al siguiente.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Para resolver este problema, especifique una versión Kubernetes compatible cuando utilice una operación de API para crear o actualizar un entorno de computación.

AWS Batch en Amazon EKS actualmente es compatible con las siguientes Kubernetes versiones:

- `1.29`
- `1.28`
- `1.27`
- `1.26`
- `1.25`
- `1.24`
- `1.23`

El perfil de instancia no existe

Si el perfil de instancia especificado no existe, el estado del entorno AWS Batch de cómputo de Amazon EKS cambia a`INVALID`. Aparece un error establecido en el parámetro `statusReason` similar al siguiente.

```
CLIENT_ERROR - Instance profile arn:aws:iam::...:instance-profile/<name> does not exist
```

Para resolver este problema, especifique o cree un perfil de instancia que funcione. Para obtener más información, consulte Rol de IAM de nodo Amazon EKS en la Guía del usuario de Amazon EKS.

Espacio de nombres de Kubernetes no válido

Si AWS Batch en Amazon EKS no puede validar el espacio de nombres del entorno de cómputo, el estado del entorno de cómputo cambia a. `INVALID` Por ejemplo, este problema puede producirse si el espacio de nombres no existe.

Aparece un mensaje de error establecido en el parámetro `statusReason` similar al siguiente.

```
CLIENT_ERROR - Unable to validate Kubernetes Namespace
```

Este problema puede producirse si se cumple cualquiera de las siguientes condiciones:

- La cadena de espacio Kubernetes de nombres de la llamada `CreateComputeEnvironment` no existe. Para obtener más información, consulte. CreateComputeEnvironment
- Los permisos de control de acceso basado en roles (RBAC) necesarios para administrar el espacio de nombres no están configurados correctamente.
- AWS Batch no tiene acceso al punto final del servidor de la Kubernetes API Amazon EKS.

Para resolver este problema, consulte Compruebe que `aws-auth ConfigMap` se ha configurado correctamente. Para obtener más información, consulte Cómo empezar con AWS Batch Amazon EKS.

Entorno de computación eliminado

Suponga que elimina un clúster de Amazon EKS antes de eliminar el entorno informático adjunto AWS Batch en Amazon EKS. A continuación, el estado del entorno de computación cambia a `INVALID`. En este escenario, el entorno de computación no funciona correctamente si se vuelve a crear el clúster de Amazon EKS con el mismo nombre.

Para resolver este problema, elimine y, a continuación, vuelva a crear AWS Batch el entorno informático de Amazon EKS.

Los nodos no se unen al clúster de Amazon EKS

AWS Batch en Amazon EKS reduce la escala de un entorno informático si determina que no todos los nodos se han unido al clúster de Amazon EKS. Cuando AWS Batch en Amazon EKS reduce la escala del entorno de cómputo, el estado del entorno de cómputo cambia a`INVALID`.

> **Note**
>
> AWS Batch no cambia el estado del entorno informático de forma inmediata para que pueda solucionar el problema.

Aparece un mensaje de error establecido en el parámetro `statusReason` similar a uno de los siguientes:

```
Your compute environment has been INVALIDATED and scaled down because
none of the instances joined the underlying ECS Cluster. Common issues
preventing instances joining are the following: VPC/Subnet configuration
preventing communication to ECS, incorrect Instance Profile policy
preventing authorization to ECS, or customized AMI or LaunchTemplate
configurations affecting ECS agent.
```

```
Your compute environment has been INVALIDATED and scaled down because
none of the nodes joined the underlying Amazon EKS Cluster. Common issues
preventing nodes joining are the following: networking configuration
preventing communication to Amazon EKS Cluster, incorrect Amazon EKS
Instance Profile or Kubernetes RBAC policy preventing authorization
to Amazon EKS Cluster, customized AMI or LaunchTemplate configurations
affecting Amazon EKS/Kubernetes node bootstrap.
```

Cuando se utiliza una AMI Amazon EKS predeterminada, las causas más comunes de este problema son las siguientes:

- El rol de la instancia no está configurado correctamente. Para obtener más información, consulte Rol de IAM de nodo Amazon EKS en la Guía del usuario de Amazon EKS.
- Las subredes no están configuradas correctamente. Para obtener más información, consulte los Requisitos y consideraciones de la VPC y las subredes de Amazon EKS en la Guía del usuario de Amazon EKS.
- El grupo de seguridad no está configurado correctamente. Para obtener más información, consulte Requisitos y consideraciones del grupo de seguridad de Amazon EKS en la Guía del usuario de Amazon EKS.

> **Note**
>
> También puede aparecer una notificación de error en el Personal Health Dashboard (PHD).

AWS Batch en Amazon EKS, el trabajo está **RUNNABLE** estancado

Un `aws-auth ConfigMap` se crea y aplica de forma automática al clúster cuando crea un grupo de nodos administrados o cuando crea un grupo de nodos mediante `eksctl`. Un `aws-auth ConfigMap` se crea inicialmente para permitir que los nodos se unan a su clúster. Sin embargo, también se utiliza `aws-auth ConfigMap` para agregar acceso de control de acceso basado en roles (RBAC) a los usuarios y roles.

Compruebe que `aws-auth ConfigMap` se ha configurado correctamente:

1. Recupere los roles mapeados en `aws-auth ConfigMap`:

   ```
   $ kubectl get configmap -n kube-system aws-auth -o yaml
   ```

2. Compruebe que `roleARN` está configurado de la siguiente manera.

   `rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch`

   > **Note**
   >
   > También puede revisar los registros del plano de control de Amazon EKS. Para obtener más información, consulte Registros del plano de control del clúster de Amazon EKS en la Guía del usuario de Amazon EKS.

Para resolver un problema en el que un trabajo está atascado en un estado `RUNNABLE`, le recomendamos que use `kubectl` para volver a aplicar el manifiesto. Para obtener más información, consulte Paso 1: Preparar el clúster de Amazon EKS para AWS Batch. O bien, puede usar `kubectl` para editar manualmente el `aws-authConfigMap`. Para obtener más información, consulte Habilitación del usuario de IAM y el acceso de rol a un clúster en la Guía del usuario de Amazon EKS.

Compruebe que **aws-auth ConfigMap** se ha configurado correctamente

Compruebe que `aws-auth ConfigMap` se ha configurado correctamente:

1. Recupere los roles mapeados en el `aws-auth ConfigMap`.

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Compruebe que `roleARN` está configurado de la siguiente manera.

```
rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

> **Note**
>
> La ruta `aws-service-role/batch.amazonaws.com/` se ha eliminado del ARN del rol vinculado a un servicio. Esto se debe a un problema con el mapa de configuración de `aws-auth`. Para obtener más información, consulte Roles con rutas que no funcionan cuando la ruta está incluida en su ARN en el aws-authconfigmap.

> **Note**
>
> También puede revisar los registros del plano de control de Amazon EKS. Para obtener más información, consulte Registros del plano de control del clúster de Amazon EKS en la Guía del usuario de Amazon EKS.

Para resolver un problema en el que un trabajo está atascado en un estado `RUNNABLE`, le recomendamos que use `kubectl` para volver a aplicar el manifiesto. Para obtener más información, consulte Paso 1: Preparar el clúster de Amazon EKS para AWS Batch. O bien, puede usar `kubectl` para editar manualmente el `aws-authConfigMap`. Para obtener más información, consulte Habilitación del usuario de IAM y el acceso de rol a un clúster en la Guía del usuario de Amazon EKS.

Los permisos o enlaces de RBAC no están configurados correctamente

Si tiene algún problema con los permisos o los enlaces del RBAC, compruebe que el rol `aws-batch` de Kubernetes pueda acceder al espacio de nombres Kubernetes:

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

También puede usar el comando **`kubectl describe`** para ver las autorizaciones de un rol de clúster o espacio de nombres Kubernetes.

```
$ kubectl describe clusterrole aws-batch-cluster-role
```

A continuación, se muestra un ejemplo del resultado.

```
Name:         aws-batch-cluster-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources                                       Non-Resource URLs  Resource Names
 Verbs
  ---------                                       -----------------  --------------
 -----
  configmaps                                      []                 []
 [get list watch]
  nodes                                           []                 []
 [get list watch]
  pods                                            []                 []
 [get list watch]
  daemonsets.apps                                 []                 []
 [get list watch]
  deployments.apps                                []                 []
 [get list watch]
  replicasets.apps                                []                 []
 [get list watch]
  statefulsets.apps                               []                 []
 [get list watch]
  clusterrolebindings.rbac.authorization.k8s.io   []                 []
 [get list]
  clusterroles.rbac.authorization.k8s.io          []                 []
 [get list]
  namespaces                                      []                 []
 [get]
```

```
$ kubectl describe role aws-batch-compute-environment-role -n my-aws-batch-namespace
```

A continuación, se muestra un ejemplo del resultado.

```
Name:         aws-batch-compute-environment-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources                                  Non-Resource URLs  Resource Names  Verbs
  ---------                                  -----------------  --------------  -----
  pods                                       []                 []              [create
 get list watch delete patch]
  serviceaccounts                            []                 []              [get list]
  rolebindings.rbac.authorization.k8s.io     []                 []              [get list]
  roles.rbac.authorization.k8s.io            []                 []              [get list]
```

Para resolver este problema, vuelva a aplicar los permisos RBAC y comandos `rolebinding`. Para obtener más información, consulte Paso 1: Preparar el clúster de Amazon EKS para AWS Batch.

Prácticas recomendadas para AWS Batch

Puede utilizar AWS Batch para ejecutar diversas cargas de trabajo computacionales exigentes a escala sin administrar una arquitectura compleja. Los trabajos AWS Batch se pueden utilizar en una amplia gama de casos de uso en áreas como la epidemiología, los juegos y el machine learning.

En este tema se describen las prácticas recomendadas que se deben tener en cuenta a la hora de utilizar AWS Batch y las directrices sobre cómo ejecutar y optimizar las cargas de trabajo al utilizar AWS Batch.

Temas

Cuándo se debe usarAWS Batch

AWS Batch ejecuta trabajos a escala y a bajo coste, y proporciona servicios de colas y escalado optimizados en función de los costes. Sin embargo, no todas las cargas de trabajo son adecuadas para su ejecución utilizando AWS Batch.

- Trabajos cortos: si un trabajo se ejecuta solo durante unos segundos, la sobrecarga necesaria para programar el trabajo por lotes puede llevar más tiempo que el tiempo de ejecución del trabajo en sí. Como solución alternativa, binpack sus tareas antes de enviarlas en AWS Batch. A continuación, configure los trabajos AWS Batch para que se repitan sobre las tareas. Por ejemplo, coloque los argumentos de cada tarea en una tabla de Amazon DynamoDB o en un archivo en un bucket de Amazon S3. Considere la posibilidad de agrupar las tareas de forma que las tareas se ejecuten durante 3 a 5 minutos cada una. Después de binpack los trabajos, repase los grupos de tareas dentro de su trabajo AWS Batch.
- Trabajos que deben ejecutarse inmediatamente: AWS Batch puede procesar trabajos rápidamente. Sin embargo, AWS Batch es un programador y optimiza buscando la rentabilidad, la prioridad

del trabajo y el rendimiento. AWS Batch podría requerir tiempo para procesar sus solicitudes. Si necesita una respuesta en menos de unos segundos, lo más adecuado es un enfoque basado en servicios que utilice Amazon ECS o Amazon EKS.

Lista de comprobación para ejecutarla a escala

Antes de ejecutar una gran carga de trabajo en 50 000 o más vCPU, tenga en cuenta la siguiente lista de comprobación.

> **Note**
>
> Si planea ejecutar una gran carga de trabajo en un millón o más de vCPU o necesita orientación para ejecutar a gran escala, póngase en contacto con su equipo de AWS.

- Compruebe sus cuotas de Amazon EC2: compruebe su cuota de servicio de Amazon EC2 (también conocidas como límites) en el panel Service Quotas de la AWS Management Console. Si es necesario, solicite un aumento de cuota para el número máximo de instancias de Amazon EC2. Recuerde que las instancias de Amazon EC2 Spot y las instancias bajo demanda de Amazon tienen cuotas distintas. Para obtener más información, consulte Introducción a Service Quotas.
- Verifique su cuota de Amazon Elastic Block Store para cada región: cada instancia utiliza un volumen GP2 o GP3 para el sistema operativo. De forma predeterminada, la cuota de cada Región de AWS es de 300 TiB. Sin embargo, cada instancia utiliza conteos como parte de esta cuota. Por lo tanto, asegúrese de tener esto en cuenta cuando verifique su cuota de Amazon Elastic Block Store para cada región. Si alcanza su cuota, no podrá crear más instancias. Para obtener más información, consulte Puntos de conexión y cuotas de Amazon Elastic Block Store.
- Utilice Amazon S3 para el almacenamiento: Amazon S3 ofrece un alto rendimiento y ayuda a eliminar las conjeturas sobre la cantidad de almacenamiento que se debe aprovisionar en función del número de trabajos e instancias de cada zona de disponibilidad. Para obtener más información, consulte Patrones de diseño de prácticas recomendadas: optimización del rendimiento de Amazon S3.
- Amplíe gradualmente para identificar los cuellos de botella en una fase temprana: en el caso de un trabajo que se ejecute en un millón o más de vCPU, comience con un nivel más bajo y aumente gradualmente para poder identificar los cuellos de botella en una fase temprana. Por ejemplo, comience por ejecutar en 50 mil vCPU. A continuación, aumente el conteo a 200 mil

vCPU y, después, a 500 mil vCPU, y así sucesivamente. En otras palabras, continúe aumentando gradualmente el número de vCPU hasta alcanzar la cantidad deseada de vCPU.

- Supervise para identificar los posibles problemas con antelación: para evitar posibles interrupciones y problemas al ejecutar a gran escala, asegúrese de supervisar tanto la aplicación como la arquitectura. Es posible que se produzcan interrupciones incluso al escalar de 1000 a 5000 vCPUs. Puede utilizar Registros de Amazon CloudWatch para revisar los datos de registro o utilizar CloudWatch Embedded Metrics mediante una biblioteca cliente. Para obtener más información, consulte Referencia del agente de CloudWatch Logs y `aws-embedded-metrics`.

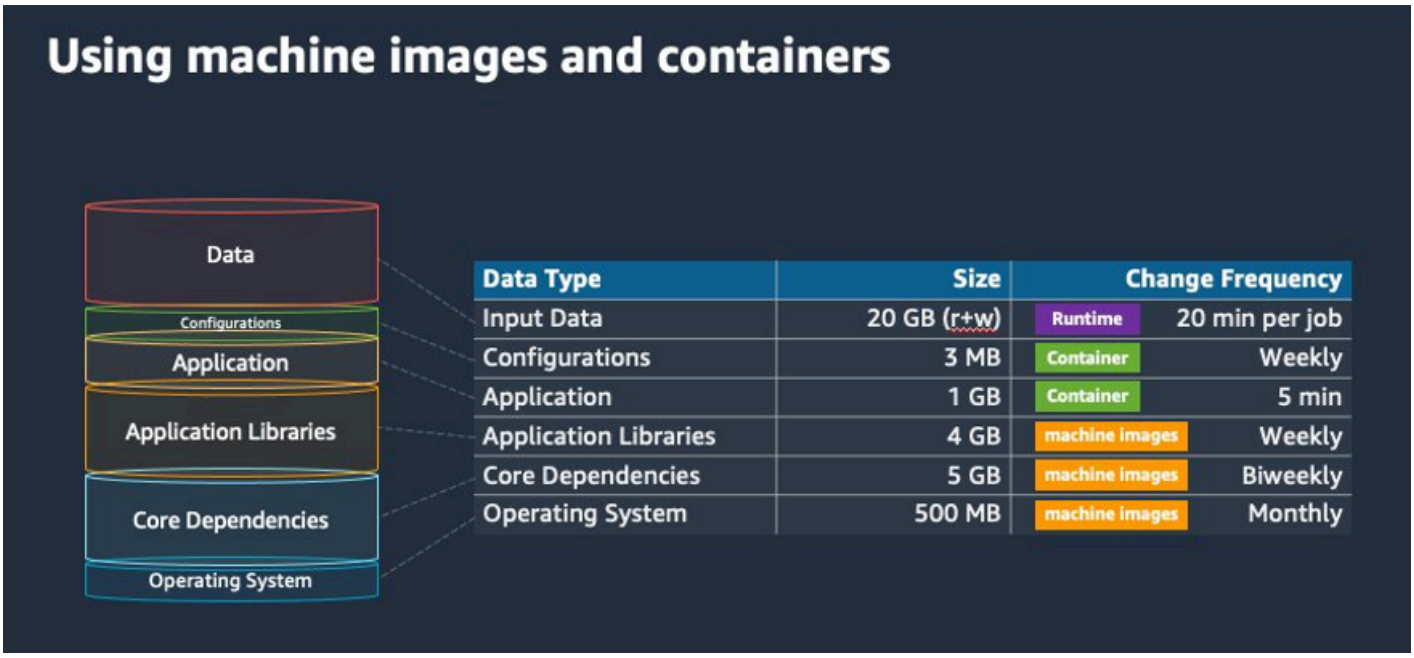
Optimice los contenedores y las AMI

El tamaño y la estructura del contenedor son importantes para la primera serie de trabajos que ejecute. Esto es especialmente cierto si el contenedor tiene más de 4 GB. Las imágenes del contenedor están integradas en capas. Docker recupera las capas en paralelo mediante tres subprocesos simultáneos. Puede aumentar el número de subprocesos simultáneos mediante el parámetro `max-concurrent-downloads`. Para obtener más información, consulte la documentación de Docker.

Aunque puede utilizar contenedores más grandes, le recomendamos que optimice la estructura y el tamaño de los contenedores para acelerar los tiempos de startup.

- Los contenedores más pequeños se obtienen más rápido: los contenedores más pequeños pueden acelerar los tiempos de inicio de las aplicaciones. Para reducir el tamaño del contenedor, descargue las bibliotecas o los archivos que se actualizan con poca frecuencia en Amazon Machine Image (AMI). También puede utilizar soportes de encuadernación para dar acceso a sus contenedores. Para obtener más información, consulte Montajes de enlace.
- Cree capas que tengan un tamaño uniforme y divida las capas grandes: cada capa se recupera mediante un hilo. Por lo tanto, una capa grande podría afectar significativamente el tiempo de startup de su trabajo. Recomendamos un tamaño máximo de capa de 2 GB como compensación entre un tamaño de contenedor más grande y tiempos de startup más rápidos. Puede ejecutar el comando `docker history your_image_id` para comprobar la estructura de la imagen del contenedor y el tamaño de la capa. Para obtener más información, consulte la documentación de Docker.
- Utilice Amazon Elastic Container Registry como repositorio de contenedores: cuando ejecuta miles de trabajos en paralelo, un repositorio autogestionado puede fallar o reducir el rendimiento.

Amazon ECR funciona a escala y puede gestionar cargas de trabajo con hasta más de un millón de vCPU.



Elija el recurso de entorno informático adecuado

AWS Fargate requiere menos instalación y configuración iniciales que Amazon EC2 y probablemente sea más fácil de usar, especialmente si es la primera vez. Con Fargate, no necesita administrar servidores, gestionar la planificación de la capacidad ni aislar las cargas de trabajo de contenedores por seguridad.

Si tiene los siguientes requisitos, le recomendamos que utilice las instancias de Fargate:

- Sus trabajos deben comenzar rápidamente, específicamente en menos de 30 segundos.
- Los requisitos de sus trabajos son 16 vCPU o menos, ninguna GPU y 120 GiB de memoria o menos.

Para obtener más información, consulte Cuándo usar Fargate.

Si tiene los siguientes requisitos, le recomendamos que utilice las instancias de Amazon EC2:

- Necesita un mayor control sobre la selección de instancias o necesita utilizar tipos de instancias específicos.

- Sus trabajos requieren recursos que AWS Fargate no puede proporcionar, como GPU, más memoria, una AMI personalizada o el adaptador Amazon Elastic Fabric.
- Necesita un alto nivel de rendimiento o simultaneidad.
- Debe personalizar la AMI, la plantilla de lanzamiento de Amazon EC2 o el acceso a parámetros especiales de Linux.

Con Amazon EC2, puede ajustar con mayor precisión su carga de trabajo a sus requisitos específicos y ejecutar a escala si es necesario.

Amazon EC2 bajo demanda o Amazon EC2 Spot

La mayoría de clientes de AWS Batch utilizan las instancias de spot de Amazon EC2 porque ahorran en comparación con las instancias bajo demanda. Sin embargo, si su carga de trabajo dura varias horas y no puede interrumpirse, las instancias bajo demanda podrían ser más adecuadas para usted. Siempre puede probar primero las instancias de spot y cambiar a las instancias bajo demanda si es necesario.

Si tiene los siguientes requisitos y expectativas, utilice las instancias bajo demanda de Amazon EC2:

- El tiempo de ejecución de sus trabajos es de más de una hora y no puede tolerar las interrupciones de su carga de trabajo.
- Tiene un SLO (objetivo de nivel de servicio) estricto para su carga de trabajo general y no puede aumentar el tiempo computacional.
- Las instancias que necesita tienen más probabilidades de sufrir interrupciones.

Si tiene los siguientes requisitos y expectativas, utilice las instancias de spot de Amazon EC2:

- El tiempo de ejecución de sus trabajos suele ser de 30 minutos o menos.
- Puede tolerar posibles interrupciones y la reprogramación de los trabajos como parte de su carga de trabajo. Para obtener más información, consulte Asistente de instancias de spot.
- Los trabajos de larga duración se pueden reiniciar desde un punto de control si se interrumpen.

Puede combinar ambos modelos de compra enviándolos primero en una instancia de spot y, a continuación, utilizando la instancia bajo demanda como opción alternativa. Por ejemplo, envíe sus trabajos en una cola que esté conectada a entornos informáticos que se ejecuten en instancias de spot de Amazon EC2. Si se interrumpe un trabajo, busca el evento en Amazon EventBridge y

correlaciónalo con una recuperación de una instancia de spot. A continuación, vuelva a enviar el trabajo a una cola bajo demanda mediante una función AWS Lambda o AWS Step Functions. Para obtener más información, consulte Tutorial: envío de alertas de Amazon Simple Notification Service Alerts para eventos de trabajos fallidos, Prácticas recomendadas para gestionar las interrupciones de las instancias de Amazon EC2 Spot y Gestionar AWS Batch con Step Functions.

> **Important**
>
> Utilice distintos tipos de instancias, tamaños y zonas de disponibilidad para su entorno informático bajo demanda a fin de mantener la disponibilidad del grupo de instancias de Amazon EC2 Spot y reducir la tasa de interrupciones.

Prácticas recomendadas para instancias de spot de Amazon EC2 para AWS Batch

Si elige las instancias de spot de Amazon Elastic Compute Cloud (EC2), es probable que pueda optimizar su flujo de trabajo para ahorrar costos, a veces de forma significativa. Para obtener más información, consulte Prácticas recomendadas de seguridad para Amazon EC2 Spot.

Para optimizar su flujo de trabajo y ahorrar costos, tenga en cuenta las siguientes prácticas recomendadas de Amazon EC2 Spot para AWS Batch:

- Elija la estrategia de asignación de **`SPOT_CAPACITY_OPTIMIZED`**: AWS Batch elige las instancias de Amazon EC2 de los grupos de capacidad de Amazon EC2 Spot más amplios. Si le preocupan las interrupciones, esta es una opción adecuada. Para obtener más información, consulte Estrategias de asignación.
- Diversifique los tipos de instancias: para diversificar sus tipos de instancias, considere tamaños y familias compatibles y, a continuación, deje que AWS Batch elija en función del precio o la disponibilidad. Por ejemplo, considere `c5.24xlarge` como una alternativa a `c5.12xlarge` o a las familias `c5a`, `c5n`, `c5d`, `m5` y `m5d`. Para más información, consulte Ser flexible con respecto a los tipos de instancia y las zonas de disponibilidad.
- Reduzca el tiempo de ejecución o el punto de control de los trabajos: recomendamos no ejecutar trabajos que tarden una hora o más cuando se utilizan instancias de spot de Amazon EC2 para evitar interrupciones. Si divide o pone puntos de control a sus trabajos en partes más pequeñas de 30 minutos o menos, puede reducir considerablemente la posibilidad de interrupciones.

- Utilice reintentos automatizados: para evitar interrupciones en los trabajos AWS Batch, configure los reintentos automatizados para los trabajos. Los trabajos por lotes pueden interrumpirse por cualquiera de los siguientes motivos: se devuelve un código de salida distinto de cero, se produce un error de servicio o se produce la recuperación de una instancia. Puede configurar hasta 10 reintentos automáticos. Para empezar, le recomendamos que establezca al menos de 1 a 3 reintentos automatizados. Para obtener información sobre el seguimiento de las interrupciones puntuales de Amazon EC2, consulte Avisos de interrupción de spot.

  Para AWS Batch, si establece el parámetro de reintento, el trabajo se coloca al principio de la cola de trabajos. Es decir, se da prioridad al trabajo. Al crear la definición de trabajo o al enviar el trabajo en la AWS CLI, puede configurar una estrategia de reintento. Para obtener más información, consulte Enviar un trabajo.

```
$ aws batch submit-job --job-name MyJob \
    --job-queue MyJQ \
    --job-definition MyJD \
    --retry-strategy attempts=2
```

- Utilice reintentos personalizados: puede configurar una estrategia de reintento de trabajo para el código de salida de una aplicación específica o la recuperación de instancias. En el siguiente ejemplo, si el host provoca el error, se puede volver a intentar el trabajo hasta cinco veces. Sin embargo, si el trabajo falla por un motivo diferente, el trabajo se cierra y el estado se establece en `FAILED`.

```
"retryStrategy": {
    "attempts": 5,
    "evaluateOnExit":
    [{
        "onStatusReason" :"Host EC2*",
        "action": "RETRY"
    },{
      "onReason" : "*"
        "action": "EXIT"
    }]
}
```

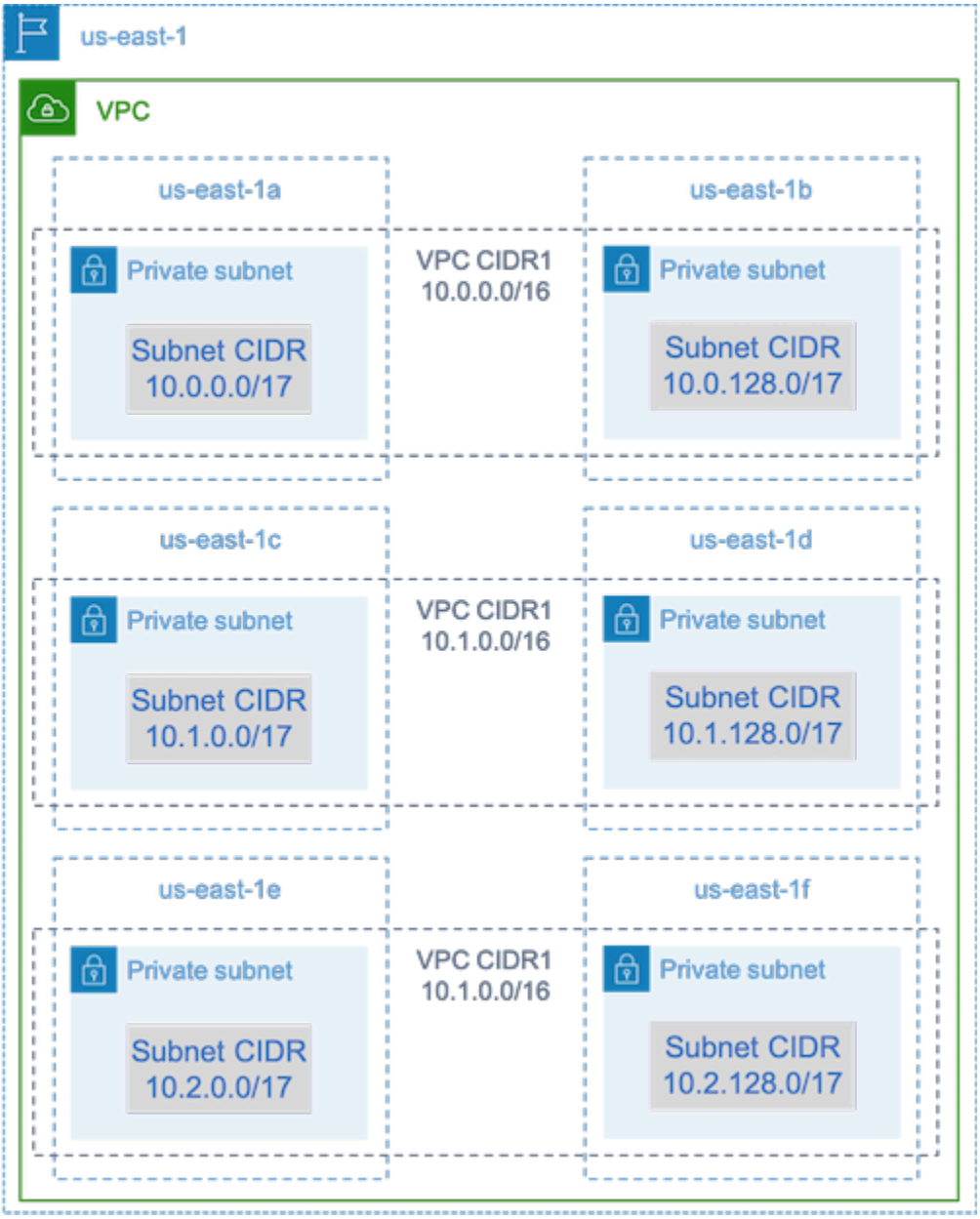
- Utilice el panel de control de interrupciones puntuales: puede utilizar el panel de control de interrupciones puntuales para realizar un seguimiento de las interrupciones puntuales. La aplicación proporciona métricas sobre las instancias de spot de Amazon EC2 que se recuperan

y las zonas de disponibilidad en las que se encuentran las instancias de spot. Para obtener más información, consulte Interrupciones de instancias de spot.

Errores comunes y solución de problemas

Los errores en AWS Batch suelen producirse en el nivel de la aplicación o se deben a configuraciones de instancias que no cumplen los requisitos específicos de su trabajo. Otros problemas incluyen que los trabajos se atasquen en un estado `RUNNABLE` o que los entornos informáticos se queden atascados en un estado `INVALID`. Para obtener más información sobre la solución de problemas de los trabajos que se quedan atascados en el estado `RUNNABLE`, consulte Trabajos bloqueados en estado `RUNNABLE`. Para obtener información sobre cómo solucionar problemas de entornos informáticos en un estado `INVALID`, consulte Entorno de computación `INVALID`.

- Compruebe las cuotas de vCPU puntuales de Amazon EC2: compruebe que sus cuotas de servicio actuales cumplen los requisitos del trabajo. Por ejemplo, supongamos que su cuota de servicio actual es de 256 vCPU y que el trabajo requiere 10 000 vCPU. Entonces, la cuota de servicio no cumple con el requisito del trabajo. Para obtener más información e instrucciones de solución de problemas, consulte Service quotas de Amazon EC2 y ¿Cómo puedo aumentar la cuota de servicio de mis recursos de Amazon EC2?.
- Los trabajos fallan antes de que se ejecute la aplicación: algunos trabajos pueden fallar debido a un error `DockerTimeoutError` o a un error `CannotPullContainerError`. Para obtener información sobre la solución de problemas, consulte ¿Cómo se resuelve el error «DockerTimeoutError» en AWS Batch?.
- Direcciones IP insuficientes: la cantidad de direcciones IP de la VPC y las subredes puede limitar la cantidad de instancias que se pueden crear. Utilice enrutamiento entre dominios sin clases (CIDR) para proporcionar más direcciones IP de las necesarias para ejecutar sus cargas de trabajo. Si es necesario, también puede crear una VPC dedicada con un gran espacio de direcciones. Por ejemplo, puede crear una VPC con varios CIDR en `10.x.0.0/16` y una subred en cada zona de disponibilidad con un CIDR de `10.x.y.0/17`. En este ejemplo, x está entre 1 y 4 e y es 0 o 128. Esta configuración proporciona 36 000 direcciones IP en cada subred.



- Compruebe que las instancias estén registradas en Amazon EC2: si ve sus instancias en la consola de Amazon EC2, pero no ve ninguna instancia de contenedor de Amazon Elastic Container Service en su clúster de Amazon ECS, es posible que el agente de Amazon ECS no esté instalado en una Imagen de máquina de Amazon (AMI). Es posible que el agente de Amazon ECS, los datos de Amazon EC2 de su AMI o la plantilla de lanzamiento tampoco estén configurados correctamente. Para aislar la causa raíz, cree una instancia Amazon EC2 independiente o conéctese a una instancia existente mediante SSH. Para obtener más información, consulte Configuración del agente contenedor de Amazon ECS, Ubicaciones de los archivos de registro de Amazon ECS y AMI de recursos de computación.

- Revise el panel de AWS: revise el panel de AWS para comprobar el estado del trabajo esperado y que el entorno informático se escala según lo esperado. También puede revisar los registros de trabajos en CloudWatch.
- Compruebe que la instancia esté creada: si se crea una instancia, significa que su entorno informático se ha escalado según lo previsto. Si sus instancias no se crean, busque las subredes asociadas en su entorno informático para cambiarlas. Para obtener más información, consulte Verificar una actividad de escalado para un grupo de escalado automático.

  También le recomendamos que verifique que sus instancias pueden cumplir con los requisitos de trabajo relacionados. Por ejemplo, un trabajo puede requerir 1 TiB de memoria, pero el entorno informático usa un tipo de instancia C5 que está limitado a 192 GB de memoria.
- Verifique que sus instancias estén siendo solicitadas por AWS Batch: consulte el historial del grupo de escalado automático para verificar que sus instancias estén siendo solicitadas por AWS Batch. Esto indica cómo Amazon EC2 intenta adquirir instancias. Si recibe un error que indica que Amazon EC2 Spot no puede adquirir una instancia en una zona de disponibilidad específica, es posible que la zona de disponibilidad no ofrezca una familia de instancias específica.
- Compruebe que las instancias se registren en Amazon ECS: si ve instancias en la consola de Amazon EC2, pero no hay instancias de contenedor de Amazon ECS en su clúster de Amazon ECS, es posible que el agente de Amazon ECS no esté instalado en la Imagen de máquina de Amazon (AMI). Además, es posible que el agente de Amazon ECS, los datos de Amazon EC2 de su AMI o la plantilla de lanzamiento tampoco estén configurados correctamente. Para aislar la causa raíz, cree una instancia Amazon EC2 independiente o conéctese a una instancia existente mediante SSH. Para obtener más información, consulte Archivo de configuración del agente de CloudWatch: sección de registros, Ubicaciones de archivos de registro de Amazon ECS y AMI de recursos de computación.
- Abra un ticket de soporte: si aún tiene problemas después de solucionar algunos problemas y tiene un plan de soporte, abra un ticket de soporte. En el ticket de soporte, asegúrese de incluir información sobre el problema, las especificaciones de la carga de trabajo, la configuración y los resultados de las pruebas. Para obtener más información, consulte Comparar AWS Support planes.
- Revise los foros de AWS Batch y HPC: para obtener más información, consulte los foros de AWS Batch y HPC.
- Revise el panel de monitoreo del tiempo de ejecución de AWS Batch: este panel utiliza una arquitectura sin servidor para capturar eventos de Amazon ECS AWS Batch, y Amazon EC2 a fin de proporcionar información sobre los trabajos y las instancias. Para obtener más información, consulte Solución de paneles de monitoreo de tiempo de ejecución de AWS Batch.

Historial del documento

En la siguiente tabla se describen los cambios importantes en la documentación desde la publicación inicial de AWS Batch. Actualizamos la documentación con frecuencia para dar respuesta a los comentarios que se nos envía.

Cambio	Descripción	Fecha
Se AWS Batch actualizaron las versiones compatibles de Amazon EKS	Se actualizaron las versiones de Amazon EKS AWS Batch compatibles para eliminar la versión 1.22.	11 de marzo de 2024
Se AWS Batch actualizaron las versiones compatibles de Amazon EKS	Se actualizaron las versiones de Amazon EKS AWS Batch compatibles para incluir la versión 1.29.	29 de febrero de 2024
Reintentos automáticos de trabajo	Se corrigió el ejemplo de código.	29 de febrero de 2024
Añade compatibilidad con trabajos con varios contenedores para AWS Batch	Añade compatibilidad con trabajos de varios contenedores AWS Batch para Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y. AWS Fargate	28 de febrero de 2024
Se AWS Batch actualizaron las versiones compatibles de Amazon EKS	Se actualizaron las versiones de Amazon EKS AWS Batch compatibles para incluir la versión 1.28	27 de enero de 2024
Actualizado BatchServiceRolePolicy y AWSBatchServiceRole	BatchServiceRolePolicy Se actualizó para añadir soporte a la descripción	5 de diciembre de 2023

	del historial de solicitudes y Amazon EC2 Auto Scaling las actividades de Spot Fleet. AWSBatchServiceRole Se actualizó para añadir identificadores de estados de cuenta, conceder AWS Batch permisos a `ec2:DescribeSpotFleetRequestHistory` y `autoscaling:DescribeScalingActivities` .	
AWS Batch en Amazon EKS	AWS Batch añade compatibilidad para ejecutar trabajos en clústeres de Amazon EKS.	25 de octubre de 2022
La prevención policial confusa entre servicios para AWS Batch	AWS Batch ahora ofrece una solución alternativa al confuso problema de seguridad adjunto, que surge cuando una entidad (un servicio o una cuenta) es obligada por otra entidad a realizar una acción por parte de otra entidad (un servicio o una cuenta).	6 de junio de 2022

Puntos de conexión de VPC de interfaz (AWS PrivateLink)	Se agregó soporte para configurar puntos finales de VPC de interfaz alimentados por. AWS PrivateLink Esto significa que puede crear una conexión privada entre su VPC y AWS Batch sin necesidad de acceso a través de una instancia de NAT, una conexión VPN o. AWS Direct Connect	15 de abril de 2022
Actualizaciones mejoradas del entorno de computación	AWS Batch actualizaciones de soporte mejoradas para entornos informáticos.	14 de abril de 2022
AWS actualizaciones de políticas gestionadas: actualización de las políticas existentes	AWS Batch actualizó las políticas gestionadas existente s.	6 de diciembre de 2021
Programación de reparto justo	AWS Batch añade soporte para añadir políticas de programación a las colas de trabajos.	9 de noviembre de 2021
Amazon EFS	AWS Batch añade compatibi lidad para añadir sistemas de archivos Amazon EFS a las definiciones de trabajo.	1 de abril de 2021
Rol vinculado al servicio añadido	AWS Batch añade el rol AWSServiceRoleForB atchvinculado al servicio.	10 de marzo de 2021
AWS Fargate soporte	AWS Batch añade soporte para ejecutar trabajos en los recursos de Fargate.	3 de diciembre de 2020

Compatibilidad con Amazon Linux 2	AWS Batch añade compatibilidad con la selección automática de las AMI de Amazon Linux 2 en el entorno informático mediante los parámetros de configuración de EC2.	24 de noviembre de 2020
Estrategia de reintento mejorada	AWS Batch mejora la estrategia de reintentos de los trabajos. Ahora se pueden volver a intentar los trabajos o impedir que se vuelvan a intentar realizar nuevos intentos haciendo coincidir el `ExitCodeReason`, o `StatusReason` de un trabajo con los patrones.	20 de octubre de 2020
Etiquetado de recursos	AWS Batch añade compatibilidad para añadir etiquetas de metadatos a sus entornos informáticos, definiciones de trabajos, colas de trabajos y trabajos.	7 de octubre de 2020
Secretos	AWS Batch añade soporte para transmitir secretos a los trabajos.	1 de octubre de 2020
Registro	AWS Batch añade compatibilidad para especificar controladores de registro adicionales para los trabajos.	1 de octubre de 2020

Estrategias de asignación	AWS Batch añade compatibilidad con varias estrategias para elegir tipos de instancias.	16 de octubre de 2019
Compatibilidad con EFA	AWS Batch añade soporte para dispositivos Elastic Fabric Adapter (EFA).	2 de agosto de 2019
Programación de GPU	AWS Batch añade la programación de la GPU. Con esta característica, puede especificar la cantidad de GPU que requiere cada trabajo y AWS Batch escalar las instancias en consecuencia.	4 de abril de 2019
Trabajos paralelos de varios nodos	AWS Batch añade soporte para trabajos paralelos de varios nodos. Puede utilizar esta característica para ejecutar trabajos individuales que abarquen varias instancias de Amazon EC2.	19 de noviembre de 2018
Permisos de nivel de recursos	AWS Batch admite permisos a nivel de recursos en varias operaciones de API.	12 de noviembre de 2018
Compatibilidad con las plantillas de lanzamiento de Amazon EC2	AWS Batch añade compatibilidad con el uso de plantillas de lanzamiento con entornos informáticos.	12 de noviembre de 2018

AWS Batch tiempos de espera de los trabajos	AWS Batch añade compatibilidad con el tiempo de espera del trabajo. Con este soporte, puede configurar un tiempo de espera específico para sus trabajos de modo que, si un trabajo dura más de lo debido, AWS Batch finalice el trabajo.	5 de abril de 2018
AWS Batch trabajos como objetivos EventBridge	AWS Batch los puestos de trabajo están disponibles como EventBridge objetivos . Al crear reglas sencillas , puede hacer coincidir los eventos y enviar trabajos de AWS Batch en respuesta a ellos.	1 de marzo de 2018
CloudTrail auditar para AWS Batch	CloudTrail puede auditar las llamadas realizadas a las acciones AWS Batch de la API.	10 de enero de 2018
Trabajos de matrices	AWS Batch añade soporte para trabajos de matriz. Puede utilizar trabajos de matrices para el barrido de parámetros y para las cargas de trabajo de Monte Carlo.	28 de noviembre de 2017

AWS Batch Etiquetado ampliado	AWS Batch amplía el soporte para la función de etiquetado. Puede utilizar esta función para especificar etiquetas para las instancias de spot de Amazon EC2 lanzadas en entornos de computación gestionados.	26 de octubre de 2017
AWS Batch flujo de eventos para EventBridge	AWS Batch añade el flujo de eventos para EventBridge. Puede utilizar la transmisión de AWS Batch eventos para recibir notificaciones casi en tiempo real sobre el estado de los trabajos que se envían a sus colas de trabajos.	24 de octubre de 2017
Reintentos automáticos de trabajo	AWS Batch añade soporte para reintentos de trabajo. Con esta actualización, puede aplicar una estrategia de reintento a sus trabajos y definiciones de trabajo, que les permita reintentar ejecutarse automáticamente si fallan.	28 de marzo de 2017
AWS Batch disponibilidad general	AWS Batch se presenta, diseñado como un medio para ejecutar cargas de trabajo de computación por lotes en el Nube de AWS.	5 de enero de 2017

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.