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? ... 1
Componentes de AWS Batch ... 1
Jobs ... 1
Definiciones de trabajo ... 2
Colas de trabajo ... 2
Entorno de computación ... 2
El AWS Batch salpicadero ... 3
Tutorial: Añadir el widget de cola de trabajos única ... 3
Tutorial: Añadir el widget CloudWatch Container Insights ... 4
Tutorial: Añadir el widget de registros de trabajos ... 4
Tutoriales de introducción ... 6
Cumplir con los requisitos previos ... 6
Cree una IAM cuenta y un usuario administrativo ... 7
Tutorial: Crear IAM roles ... 9
Tutorial: Crear un key pair ... 9
Tutorial: Crear un VPC ... 12
Tutorial: Crear un grupo de seguridad ... 13
Tutorial: Instale el AWS CLI ... 15
Tutorial: Cómo empezar a usar Amazon EC2 ... 15
Crear un entorno de computación ... 15
Crear una cola de trabajos ... 20
Creación de una definición de trabajo ... 20
Creación de un trabajo ... 25
Revisar y crear ... 25
Tutorial: Cómo empezar a usar Fargate ... 25
Crear un entorno de computación ... 25
Crear una cola de trabajos ... 27
Creación de una definición de trabajo ... 27
Creación de un trabajo ... 30
Revisar y crear ... 31
Tutorial: Cómo empezar con Amazon EKS ... 31
Requisitos previos ... 32
Prepara tu EKS clúster de Amazon para AWS Batch ... 33
Cree un entorno EKS informático de Amazon ... 37

Cree una cola de trabajos y adjunte el entorno informático ... 39
Creación de una definición de trabajo ... 40
Enviar un trabajo ... 41
(Opcional) Envíe un trabajo con cambios ... 41
Tutorial: Cómo empezar a utilizar AWS Batch Amazon EKS Private Clusters ... 42
Requisitos previos ... 43
Prepare su EKS clúster para AWS Batch ... 45
Cree un entorno EKS informático de Amazon ... 49
Cree una cola de trabajos y adjunte el entorno informático ... 51
Creación de una definición de trabajo ... 51
Enviar un trabajo ... 41
(Opcional) Envíe un trabajo con cambios ... 41
Resolución de problemas ... 54
Entornos informáticos para AWS Batch ... 55
Entornos de computación administrados ... 55
Consideraciones a la hora de crear trabajos paralelos de varios nodos ... 58
Entornos de computación no administrados ... 58
Crear un entorno de computación ... 59
Tutorial: Cree un entorno informático gestionado con los recursos de Fargate ... 60
Tutorial: Creación de un entorno informático gestionado con EC2 los recursos de Amazon ... 62
Tutorial: Creación de un entorno informático no gestionado con los recursos de Amazon EC2 ... 68
Tutorial: Creación de un entorno informático gestionado con EKS los recursos de Amazon ... 69
Recurso: plantilla de entorno informático ... 72
Recurso informático AMIs ... 74
Especificación de recursos AMI informáticos ... 76
Tutorial: Crear un recurso de cómputo AMI ... 78
Usa una GPU carga de trabajo AMI ... 81
Obsolescencia de Amazon Linux ... 87
Usa plantillas de EC2 lanzamiento de Amazon ... 88
Datos EC2 de usuario de Amazon en plantillas de lanzamiento ... 90
Referencia: ejemplos de plantillas de lanzamiento ... 91
EC2configuraciones ... 94
Estrategias de asignación de tipos de instancia ... 95
Administración de la memoria ... 96
Reserve memoria del sistema ... 98

Tutorial: Vea la memoria de los recursos de cómputo ... 98
CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS ... 99
Actualizar entornos de computación ... 104
Actualización de los entornos AWS Fargate informáticos ... 104
Actualizar el AMI ID ... 107
Entornos de cómputo Fargate ... 109
Cuándo usar Fargate ... 109
Definiciones de trabajo en Fargate ... 110
Colas de trabajo en Fargate ... 112
Entornos de computación en Fargate ... 112
Entornos de EKS cómputo de Amazon ... 113
Amazon EKS ... 116
Predeterminado AMI ... 116
Compatible Kubernetes control de versiones ... 117
Actualice la Kubernetes versión del entorno informático ... 118
Responsabilidad compartida del Kubernetes nodos ... 119
Ejecuta un DaemonSet en nodos AWS Batch gestionados ... 120
Personalice las plantillas de EKS lanzamiento de Amazon ... 120
Colas de trabajo ... 125
Crear una cola de trabajos ... 125
Tutorial: Crear una cola de EC2 trabajos en Amazon ... 125
Tutorial: Crear una cola de trabajos en Fargate ... 127
Tutorial: Crear una cola de EKS trabajos en Amazon ... 128
Referencia: plantilla de cola de trabajos ... 129
Ver el estado de la cola de trabajos ... 130
Vea la información de la cola de trabajos ... 130
Políticas de programación de reparto justo ... 132
Utilice identificadores de acciones ... 132
Utilice políticas de programación ... 133
Utilice una programación de reparto justo ... 134
Tutorial: Crear una política de programación ... 135
Referencia: plantilla de política de programación ... 137
Definiciones de trabajo ... 138
Crear una definición de trabajo de un solo nodo ... 138
Tutorial: Crear una definición de trabajo de un solo nodo en los recursos de Amazon EC2 .. 139
Tutorial: Cree una definición de trabajo de un solo nodo en los recursos de Fargate ... 145

Tutorial: Crear una definición de trabajo de un solo nodo en los recursos de Amazon EKS .. 152
Cree una definición de trabajo paralelo de varios nodos 156
Tutorial: Crear una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2 157
Referencia: plantilla de definición de trabajo que utiliza ContainerProperties 164
Referencia: parámetros de definición de trabajo para ContainerProperties 172
Cree definiciones de trabajo mediante EcsProperties 217
`ContainerPropertiesen` comparación con las definiciones de puestos `EcsProperties` 218
Cambios generales en el AWS Batch APIs 219
Definiciones de trabajos con varios contenedores para Amazon ECS 219
Definiciones de trabajos con varios contenedores para Amazon EKS 220
Referencia: escenarios de AWS Batch trabajo que utilizan EcsProperties 221
Utilice el controlador de registro awslogs 227
opciones del controlador de registro de awslogs en el tipo de datos AWS Batch JobDefiniton 228
Especifique una configuración de registro en la definición de su trabajo 231
Especificar datos confidenciales 232
Uso de Secrets Manager 233
Utilice el almacén de parámetros de Systems Manager 241
Autenticación de registro privado para trabajos 245
IAMPermisos necesarios para la autenticación del registro privado 246
Tutorial: Cree un secreto para la autenticación del registro privado 247
EFSVolúmenes de Amazon 248
Consideraciones sobre EFS el volumen de Amazon 249
Usa los puntos de EFS acceso de Amazon 250
Especifica un sistema de EFS archivos de Amazon en tu definición de trabajo 251
Referencia: ejemplos de definiciones de trabajo 254
Referencia: variables de entorno 254
Referencia: sustitución de parámetros 255
Referencia: Pruebe la GPU funcionalidad 256
Referencia: trabajo paralelo multinodo 257
Jobs 259
Tutorial: enviar un trabajo 259
Estados de trabajo 262
Variables de entorno de trabajo 265

Reintentos automáticos de trabajo 267
Dependencias de trabajos 269
Tiempos de espera de trabajo 269
EKSEmpleos en Amazon 270
Tutorial: Asigne un trabajo en ejecución a un pod y un nodo 271
Tutorial: asigne un pod en ejecución a su función 272
Trabajos paralelos de varios nodos 274
Variables de entorno 275
Grupos de nodos 276
Ciclo de vida del trabajo 276
Consideraciones del entorno de computación 277
Trabajos paralelos de varios nodos en Amazon EKS 279
¿Ejecutando trabajos? MNP 279
Crear una definición de EKS MNP trabajo en Amazon 281
Envía un EKS MNP trabajo en Amazon 283
Anular una definición de EKS MNP trabajo de Amazon 284
Trabajos de matrices 284
Referencia: ejemplo de un flujo de trabajo de tareas matriciales 287
Tutorial: Uso del índice de trabajo de matriz 291
Ejecute GPU trabajos 297
Tutorial: Crear una GPU basada Kubernetes clúster en Amazon EKS 299
Tutorial: Crear una definición de EKS GPU trabajo en Amazon 301
Tutorial: Ejecuta un GPU trabajo en tu EKS clúster de Amazon 302
Busque AWS Batch trabajos en una cola de trabajos 302
Tutorial: Busque y filtre AWS Batch trabajos 303
Tutorial: (Beta) Busque y filtre AWS Batch trabajos 304
Tutorial: envíe los registros de AWS Batch trabajos a CloudWatch Logs 306
Tutorial: Revise la información del AWS Batch trabajo 307
Seguridad en AWS Batch 308
Identity and Access Management 309
Público 309
Autenticación con identidades 310
Administración de acceso mediante políticas 313
¿Cómo AWS Batch funciona con IAM 316
IAMpolíticas, funciones y permisos 339
IAMestructura de políticas 340

Recurso: Ejemplos de políticas ... 344
Recurso: política AWS Batch gestionada ... 351
Rol de ejecución de AWS Batch IAM ... 353
Permisos de nivel de recursos admitidos ... 354
Tutorial: Crea el rol IAM de ejecución ... 367
Tutorial: Compruebe la función IAM de ejecución ... 367
Uso de roles vinculados a servicios ... 368
Función de ECS instancia de Amazon ... 376
Función de Amazon EC2 en la flota spot ... 380
EventBridge IAMrol ... 383
Creación de una nube virtual privada ... 384
Crea un VPC ... 385
Siguientes pasos ... 385
VPCpuntos finales ... 386
Consideraciones ... 386
Creación de un punto de conexión de interfaz ... 388
Creación de una política de punto de conexión ... 388
Validación de conformidad ... 389
Seguridad de la infraestructura ... 391
Prevención de la sustitución confusa entre servicios ... 391
Ejemplo: función para acceder a un solo entorno de cómputo ... 392
Ejemplo: función para acceder a varios entornos de cómputo ... 393
CloudTrail ... 394
AWS Batch información en CloudTrail ... 394
Referencia: Descripción de las entradas de los archivos de AWS Batch registro ... 395
Solucionar problemas AWS Batch IAM ... 397
No tengo autorización para realizar una acción en AWS Batch ... 397
No estoy autorizado a realizar iam: PassRole ... 398
Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS Batch recursos ... 398
AWS Step Functions ... 400
Tutorial: Vea los detalles de las máquinas de estado ... 400
Tutorial: Editar una máquina de estados ... 401
Tutorial: Ejecute una máquina de estados ... 401
Amazon EventBridge ... 402
AWS Batch eventos ... 403

Recurso: Eventos de cambio de estado de trabajo ... 403
Recurso: Eventos bloqueados por la cola de trabajos ... 405
Tutorial: Usa las notificaciones AWS de usuario con AWS Batch ... 407
AWS Batch puestos de trabajo como EventBridge objetivos ... 407
Tutorial: Crear un trabajo programado ... 409
Tutorial: Cree una regla con un patrón de eventos ... 411
Tutorial: Pase el transformador de entrada ... 414
Tutorial: Escuche los eventos AWS Batch laborales ... 417
Requisitos previos ... 417
Tutorial: Creación de la función Lambda ... 417
Tutorial: Registre la regla de eventos ... 418
Tutorial: Pruebe su configuración ... 420
Tutorial: Envío de alertas de Amazon Simple Notification Service sobre eventos de trabajo fallidos ... 421
Requisitos previos ... 421
Tutorial: Crear un SNS tema de Amazon y suscribirse a él ... 421
Tutorial: Registra una regla de evento ... 422
Tutorial: Pon a prueba tu regla ... 424
Regla alternativa: cola de trabajos por lotes bloqueada ... 424
Elastic Fabric Adapter ... 425
Supervisar AWS Batch ... 428
CloudWatch Registros ... 428
Tutorial: Añadir una política de CloudWatch registros IAM ... 429
Instale y configure el CloudWatch agente ... 431
Tutorial: Ver CloudWatch registros ... 432
CloudWatch Información sobre contenedores ... 434
Tutorial: activar Container Insights ... 434
Usa CloudWatch Logs para monitorear AWS Batch los EKS trabajos de Amazon ... 435
Requisitos previos ... 436
Instálelo AWS para Fluent Bit ... 436
Active Fluent Bit para los nodos AWS Batch ... 436
Etiquetar los recursos ... 437
Conceptos básicos de etiquetas ... 437
Etiquetar los recursos ... 438
Restricciones de las etiquetas ... 439
Tutorial: Administra las etiquetas mediante la consola ... 440

Añada etiquetas a un recurso individual al crearlo 440
Agregar y eliminar etiquetas en un recurso individual 440
Gestione las etiquetas mediante CLI o API 441
Prácticas recomendadas 443
¿Cuándo usar AWS Batch 443
Lista de comprobación para ejecutarla a escala 444
Optimice los contenedores y AMIs 445
Elija el recurso de entorno de computación adecuado 446
Amazon EC2 On-Demand o Amazon EC2 Spot 447
Utilice las prácticas recomendadas de Amazon EC2 Spot para AWS Batch 448
Errores comunes y solución de problemas 450
Resolución de problemas 453
AWS Batch 454
Entorno de computación `INVALID` 454
Trabajos bloqueados en estado `RUNNABLE` 457
Instancias de spot no etiquetadas en el momento de su creación 462
Las instancias de spot no se están reduciendo verticalmente 462
No puedo recuperar los secretos de Secrets Manager 464
No se pueden anular los requisitos de recursos de la definición del trabajo 464
Mensaje de error al actualizar la configuración de `desiredvCpus` 466
AWS Batch en Amazon EKS 466
Entorno de computación `INVALID` 467
AWS Batch en Amazon, el EKS trabajo está `RUNNABLE` estancado 470
Compruebe que `aws-auth ConfigMap` se ha configurado correctamente 471
RBAClos permisos o los enlaces no están configurados correctamente 472
Recurso: Cuotas de servicio 474
Historial de documentos 475
..... cdlxxxii

¿Qué es AWS Batch?

AWS Batch le ayuda a ejecutar cargas de trabajo de computación por lotes en el. Nube de AWS La computación por lotes es una forma habitual de que los desarrolladores, científicos e ingenieros accedan a grandes cantidades de recursos informáticos. AWS Batch elimina el trabajo pesado e indiferenciado que supone configurar y administrar la infraestructura requerida, de forma similar a lo que ocurre con el software de computación por lotes tradicional. Este servicio puede aprovisionar recursos de forma eficaz en respuesta a los trabajos enviados para eliminar limitaciones de capacidad, reducir costos de computación y ofrecer resultados con rapidez.

Como servicio totalmente gestionado, le AWS Batch ayuda a ejecutar cargas de trabajo de computación por lotes de cualquier escala. AWS Batch aprovisiona automáticamente los recursos informáticos y optimiza la distribución de la carga de trabajo en función de la cantidad y la escala de las cargas de trabajo. De esta forma AWS Batch, no es necesario instalar ni administrar software de computación por lotes, por lo que puede dedicar su tiempo a analizar los resultados y resolver problemas.

Temas

- Componentes de AWS Batch
- El AWS Batch salpicadero

Componentes de AWS Batch

AWS Batch simplifica la ejecución de trabajos por lotes en varias zonas de disponibilidad de una región. Puede crear entornos de AWS Batch cómputo dentro de una nueva o existenteVPC. Después de que un entorno de computación se ha activado y asociado a una cola de trabajos, puede precisar las definiciones de trabajo que especifican cuáles imágenes de contenedor de 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 .

Jobs

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 contenerizada en EC2 los recursos de AWS Fargate Amazon de su entorno informático, utilizando los parámetros que especifique 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 IAM rol que le dé acceso a otros AWS recursos. También puede especificar tanto la memoria como los CPU requisitos. 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, consulte Definiciones de trabajo

Colas de trabajo

Cuando envías un AWS Batch trabajo, lo envías a una cola de trabajos determinada, donde reside el trabajo hasta que se programa en un entorno informático. Asocie uno o más entornos de computación con una cola de trabajos. También puede asignar valores de prioridad a estos entornos de computación 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 de computación son más económicos.

Entorno de computación

Un entorno de computación es un conjunto de recursos de computación administrados o no administrados que se utilizan para ejecutar trabajos. Con los entornos de procesamiento gestionados, puede especificar el tipo de procesamiento deseado (Fargate oEC2) con varios niveles de detalle. Puede configurar entornos de cómputo que utilicen un tipo concreto de EC2 instancia, un modelo concreto, como `c5.2xlarge` `om5.10xlarge`. O bien, usted puede elegir especificar únicamente que desea utilizar los tipos de instancias más recientes. También puede especificar el número mínimo, deseado y máximo vCPUs para el entorno, así como el importe que está dispuesto a pagar por una instancia puntual como porcentaje del precio de la instancia bajo demanda y del conjunto objetivo de VPC subredes. AWS Batch lanza, administra y termina de manera eficiente los tipos de cómputo según sea necesario. También puede administrar sus propios entornos de computación. Por lo tanto, eres responsable de configurar y escalar las instancias de un ECS

clúster de Amazon que AWS Batch cree para ti. Para obtener más información, consulte Entornos informáticos para AWS Batch.

El AWS Batch salpicadero

A través del AWS Batch panel, puede monitorear los trabajos recientes, las colas de trabajos y los entornos de cómputo. 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 AWS Batch trabajos, consulte. Colas de trabajo
- Descripción general del entorno informático: para obtener más información sobre los entornos AWS Batch informáticos, consulte. Entornos informáticos para AWS Batch

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.

Temas

- Tutorial: Añada el widget de cola de trabajos única al panel AWS Batch
- Tutorial: Añada el widget CloudWatch Container Insights al AWS Batch panel
- Tutorial: Añadir el widget de registros de trabajos al AWS Batch panel

Tutorial: Añada el widget de cola de trabajos única al panel AWS Batch

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

Para añadir este widget, siga estos pasos.

1. Abra la consola de AWS Batch.
2. En la barra de navegación, selecciona la Región de AWS que desees.
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 de computación conectados si no desea mostrar las propiedades de los entornos de computación.
9. En Propiedades del entorno de computación, elija las propiedades que desee.
10. Elija Añadir.

Tutorial: Añada el widget CloudWatch Container Insights al AWS Batch panel

Este widget muestra métricas agregadas para trabajos y entornos de AWS Batch cómputo. Para obtener más información sobre los contenedores, consulte the section called “CloudWatch Información sobre contenedores”.

Para añadir este widget, siga estos pasos.

1. Abra la consola de AWS Batch.
2. En la barra de navegación, selecciona la Región de AWS que desees.
3. En el panel de navegación, elija Panel.
4. Elija Añadir widget.
5. En Información de contenedores, elija Añadir widget.
6. En Entorno de computación, elija el entorno de computación que desee.
7. Elija Añadir.

Tutorial: Añadir el widget de registros de trabajos al AWS Batch panel

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 “Tutorial: envíe los registros de AWS Batch trabajos a CloudWatch Logs”.

Para añadir este widget, siga estos pasos.

1. Abra la consola de AWS Batch.
2. En la barra de navegación, selecciona lo Región de AWS que desees.
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.

Cómo empezar con AWS Batch los tutoriales

Puedes usar 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.

Asegúrese de completar los AWS Batch requisitos previos antes de empezar con los tutoriales de AWS Batch introducción.

Después, puede usar el asistente de AWS Batch primera ejecución para crear un entorno informático, una cola de trabajos y enviar un ejemplo de trabajo a Hello World. Si ya tienes una imagen de Docker con la que quieres lanzarla AWS Batch, puedes crear una definición de trabajo con esa imagen y, en su lugar, enviarla a tu cola.

Complete los AWS Batch requisitos previos

Si ya te has registrado en Amazon Web Services (AWS) y utilizas Amazon Elastic Compute Cloud (AmazonEC2) o Amazon Elastic Container Service (AmazonECS), pronto podrás utilizarlos AWS Batch. El proceso de configuración para estos servicios es similar. Esto se debe a que AWS Batch utiliza instancias de ECS contenedor de Amazon en sus entornos de cómputo. Para usarlo AWS CLI con AWS Batch , debe usar una versión del AWS CLI que sea compatible con las AWS Batch funciones más recientes. Si no ves compatibilidad AWS Batch con una función en el AWS CLI, actualiza a la versión más reciente. Para obtener más información, consulta http://aws.amazon.com/cli/.

> **Note**
>
> Como AWS Batch utiliza componentes de AmazonEC2, se utiliza la EC2 consola de Amazon para muchos de estos pasos.

Realiza las siguientes tareas para prepararte AWS Batch. Si ya ha realizado alguno de estos pasos, puede omitirlos y proceder con la instalación de AWS CLI.

Temas

- Cree una IAM cuenta y un usuario administrativo
- Tutorial: Cree IAM roles para sus entornos de cómputo e instancias de contenedores
- Tutorial: Cree un key pair para sus instancias
- Tutorial: Crear un VPC
- Tutorial: Crear un grupo de seguridad
- Tutorial: Instale el AWS CLI

Cree una IAM cuenta y un usuario administrativo

Para empezar, debe crear una AWS cuenta y un usuario único al que normalmente se le concedan derechos administrativos. Para ello, complete los siguientes tutoriales:

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
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.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. 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 y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .

2. Activa la autenticación multifactorial (MFA) para tu usuario root.

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

Creación de un usuario con acceso administrativo

1. Habilite IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .

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

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

Iniciar sesión como usuario con acceso de administrador

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

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

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos con privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

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

Tutorial: Cree IAM roles para sus entornos de cómputo e instancias de contenedores

Sus entornos de AWS Batch cómputo e instancias de contenedores requieren Cuenta de AWS credenciales para realizar llamadas a otras AWS APIs personas en su nombre. Cree un AWS Identity and Access Management rol que proporcione estas credenciales a sus entornos de cómputo e instancias de contenedor y, a continuación, asocie ese rol a sus entornos de cómputo.

> **Note**
>
> Para comprobar que Cuenta de AWS dispone de los permisos necesarios, consulte Configuración inicial del IAM servicio para su cuenta.
> Los roles del entorno de AWS Batch cómputo y de la instancia de contenedor se crean automáticamente en la primera ejecución de la consola. Por lo tanto, si tiene intención de utilizar la AWS Batch consola, puede pasar a la siguiente sección. Si piensa utilizar la AWS CLI en su lugar, complete los procedimientos Función de ECS instancia de Amazon antes de Utilice funciones vinculadas al servicio para AWS Batch crear su primer entorno de cómputo.

Tutorial: Cree un key pair para sus instancias

AWS utiliza criptografía de clave pública para proteger la información de inicio de sesión de la instancia. Una instancia de Linux, como una instancia contenedora de un entorno de AWS Batch cómputo, no tiene contraseña para SSH acceder a ella. Utilice un par de claves para iniciar sesión

de forma segura en la instancia. Especificas el nombre del par de claves cuando creas tu entorno informático y, a continuación, proporcionas la clave privada cuando inicias sesión con élSSH.

Si aún no has creado un key pair, puedes crear uno con la EC2 consola de Amazon. Tenga en cuenta que, si planea lanzar instancias en varias Regiones de AWS, cree un key pair en cada región. Para obtener más información sobre las regiones, consulta Regiones y zonas de disponibilidad en la Guía del EC2 usuario de Amazon.

Crear un par de claves

1. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
2. En la barra de navegación, seleccione una Región de AWS para el key pair. 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 utilizas un SSH cliente de un ordenador Mac o Linux para conectarte a tu instancia de Linux, utiliza el siguiente comando para configurar los permisos del archivo de clave privada. De esa forma, solo usted podrá leerlo.

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

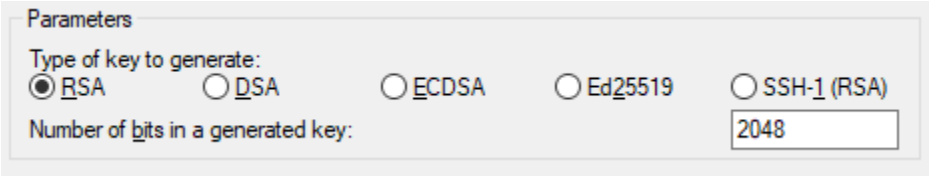
Para obtener más información, consulte Amazon EC2 Key Pairs en la Guía del EC2 usuario de Amazon.

Para conectarse a la instancia mediante el par de claves

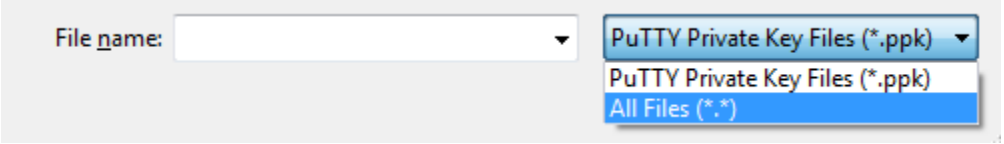
Para conectarse a su instancia de Linux desde un ordenador con Mac o Linux, especifique el `.pem` archivo para su SSH cliente con la `-i` opción y la ruta de acceso a su clave privada. Para conectarte a tu instancia de Linux desde un ordenador con Windows, usa Pu MindTerm o PuTTY. Si planea usar PuTTY,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

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

1. Descargue e instale Pu TTY desde http://www.chiark.greenend.org.uk/~sgtatham/putty/. Asegúrese de instalar el conjunto completo.
2. Inicie P uTTYgen (por ejemplo, en el menú Inicio, seleccione Todos los programasTTY, Pu y P uTTYgen).
3. En Tipo de clave que se va a generar, elija RSA. Si utilizas una versión anterior de PuTTYgen, selecciona SSH-2 RSA.



4. Elija Load (Cargar). De forma predeterminada, P uTTYgen muestra solo los 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). P uTTYgen muestra una advertencia acerca de cómo guardar la clave sin contraseña. Seleccione Yes.
7. Especifique para la clave el mismo nombre que utilizó para el par de claves. Pu añade TTY automáticamente la extensión del archivo `.ppk`.

Tutorial: Crear un VPC

Con Amazon Virtual Private Cloud (AmazonVPC), puede lanzar AWS recursos en una red virtual que haya definido. Le recomendamos encarecidamente que lance sus instancias de contenedor en unVPC.

Si tienes una opción predeterminadaVPC, también puedes saltarte esta sección y pasar a la siguiente tareaTutorial: Crear un grupo de seguridad. Para determinar si tiene un valor predeterminadoVPC, consulte Plataformas compatibles en Amazon EC2 Console en la Guía del EC2 usuario de Amazon

Para obtener información sobre cómo crear un AmazonVPC, consulta Crear un Amazon VPC únicamente en la Guía del VPC usuario de Amazon. Consulte la siguiente tabla para determinar qué opciones seleccionar.

Opción	Valor
Recursos para crear	VPCúnicamente
Nombre	Si lo desea, proporcione un nombre para suVPC.
IPv4CIDRbloquear	IPv4CIDRentrada manual El tamaño del CIDR bloque debe estar comprendido entre /16 y /28.
IPv6CIDRbloque	Sin IPv6 CIDR bloqueo
Propiedad	Predeterminado

Para obtener más información sobre AmazonVPC, consulta ¿Qué es AmazonVPC? en la Guía del VPC usuario de Amazon.

Tutorial: Crear un grupo de seguridad

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

Puede agregar reglas a un grupo de seguridad que le permitan conectarse a su instancia de contenedor desde su dirección IP medianteSSH. También puedes añadir reglas que permitan la entrada y la salida HTTP y el HTTPS acceso 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, consulta Regiones y zonas de disponibilidad en la Guía del EC2 usuario de Amazon.

> **Note**
>
> Necesitará la dirección IP pública de su equipo local, que puede obtener mediante un servicio. Por ejemplo, ofrecemos 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 desde un firewall sin una dirección IP estática, averigüe el rango de direcciones IP que utilizan los ordenadores cliente.

Para crear un grupo de seguridad con la consola

1. Abre la VPC consola de Amazon en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Security Groups.
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. Desde VPC, eligeVPC.
6. (Opcional) 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.

AWS Batch las instancias de contenedor no requieren que ningún puerto de entrada esté abierto. Sin embargo, es posible que desees añadir una SSH regla. De ese modo, puede iniciar sesión en la instancia de contenedor y examinar los contenedores en los trabajos con comandos de Docker. Si quieres que tu instancia contenedora aloje un trabajo que ejecute un servidor web, también puedes añadir reglas para elloHTTP. 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, elija HTTP. En Fuente, elija Cualquier lugar (`0.0.0.0/0`).
- Seleccione Add Rule (Agregar regla). En Tipo, elija SSH. En Fuente, elija IP personalizada y especifique la dirección IP pública de su equipo o red en la notación Classless Inter-Domain Routing (CIDR). Si su empresa asigna direcciones de un rango, especifíquelo; por ejemplo, `203.0.113.0/24`. Para especificar una dirección IP individual en CIDR notación, elija Mi IP. Esto añade el prefijo de enrutamiento `/32` a la dirección IP pública.

> **Note**
>
> Por motivos de seguridad, no recomendamos que permitas el SSH acceso desde todas las direcciones IP (`0.0.0.0/0`) a tu instancia, sino solo con fines de prueba y solo durante un período breve.

7. Puede agregar etiquetas ahora o más adelante. Para agregar una etiqueta, elija Agregar nueva etiqueta y, a continuación, ingrese la clave y el valor de la etiqueta.
8. Elija 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.

Tutorial: Instale el AWS CLI

Para usarlo AWS CLI con AWS Batch, instale la última AWS CLI versión. Para obtener información sobre la instalación AWS CLI o la actualización a la última versión, consulte Instalación de la interfaz de línea de AWS comandos en la Guía del AWS Command Line Interface usuario.

Tutorial: Cómo empezar a utilizar Amazon EC2 Orchestration

Amazon Elastic Compute Cloud (AmazonEC2) proporciona una capacidad informática escalable en el Nube de AWS. El uso de Amazon EC2 elimina la necesidad de invertir en hardware por adelantado, por lo que puede desarrollar e implementar aplicaciones más rápido.

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

Crear un entorno de computación

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

1. Abra el Asistente de primer uso de la consola AWS Batch.
2. En Seleccione el tipo de orquestación, elija Amazon Elastic Compute Cloud (AmazonEC2).
3. Elija Next (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 el caso del rol de instancia, elige un perfil de instancia existente que tenga adjuntos IAM los permisos necesarios. Este perfil de instancia permite a las instancias ECS contenedoras de Amazon de tu entorno informático realizar llamadas a las AWS API operaciones requeridas. Para obtener más información, consulte Función de ECS instancia de Amazon.
6. (Opcional) Una etiqueta es una marca que se asigna a un recurso. Para añadir una etiqueta o una EC2 etiqueta de Amazon, expande Etiquetas y, a continuación, selecciona Añadir 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 de Usar instancias de Amazon EC2 Spot, active Habilitar el uso de instancias de 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) Para el rol de flota puntual, elija un rol existente en Amazon EC2 Spot Fleet IAM para aplicarlo a su entorno de cómputo puntual. Si aún no tienes un IAM rol en Amazon EC2 Spot Fleet, primero debes crear uno. Para obtener más información, consulte Función de Amazon EC2 en la flota spot.

> **Important**
>
> Para etiquetar sus instancias puntuales al crearlas, su IAM rol en Amazon EC2 Spot Fleet debe usar la nueva política EC2SpotFleetTaggingRole gestionada por Amazon. La política EC2SpotFleetRole gestionada por Amazon no tiene los permisos necesarios para etiquetar instancias puntuales. Para obtener más información, consulte Instancias de spot no etiquetadas en el momento de su creación y Etiquetar los recursos.

10. vCPUsEn Mínimo, elija el número mínimo EC2 vCPUs que mantenga su entorno informático, independientemente de la demanda de colas de trabajos.
11. En Desired vCPUs, elija el número con el EC2 vCPUs que se inicia su entorno de cómputo. A medida que aumente la demanda de colas de trabajos, AWS Batch aumente el número deseado de EC2 instancias vCPUs y agréguelas. El número de vCPUs puede aumentar hasta el número máximo devCPUs. A medida que la demanda AWS Batch disminuya, disminuya la cantidad deseada de instancias vCPUs y las elimine. El número de se reduce completamente hasta el número mínimo devCPUs.
12. vCPUsEn Máximo, elija el número máximo al EC2 vCPUs que puede ampliarse su entorno informático, independientemente de la demanda de colas de trabajos.
13. En Tipos de instancias permitidos, selecciona los tipos de EC2 instancias de Amazon 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 puedes elegir seleccionar `optimal` los tipos de instancia (de C4, M4, y R4 familias de instancias) que se adapten a la demanda de sus 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 puedes mezclar instancias x86 e ARM instancias en el mismo entorno de cómputo.

> **Note**
>
> AWS Batch escala GPUs en función de la cantidad requerida en sus colas de trabajos. Para utilizar la GPU programación, el entorno informático debe incluir tipos de instancias de la familia`p2`,`p3`,`p4`,`p5`, `g3` `g3sg4`, o de la `g5` familia.

> **Note**
>
> Actualmente, `optimal` utiliza tipos de instancias de C4, M4, y R4 familias de instancias. Regiones de AWS Eso no tiene tipos de instancias de esas familias de instancias, tipos de instancia de C5, M5, y R5 se utilizan familias de 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) Para el par de EC2 claves, elige un par de claves pública y privada como credenciales de seguridad cuando te conectes a la instancia. Para obtener más información sobre los pares de EC2 claves de Amazon, consulta Pares de EC2 claves de Amazon 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 cómputo EC2 bajo demanda y SPOT_ CAPACITY _ OPTIMIZED para los entornos de cómputo EC2 puntual. Para

obtener más información, consulte Estrategias de asignación de tipos de instancia para AWS Batch.

18. (Opcional) Para EC2la configuración, elija Agregar EC2 configuración. Elija los valores de anulación de tipo de imagen e ID de imagen AWS Batch para proporcionar información y seleccionar Amazon Machine Images (AMIs) 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 imagen ECSoptimizada recientemente para AMI Amazon. Si no se especifica ningún tipo de imagen, el valor predeterminado es una instancia de Amazon Linux 2 para instancias que no GPU sean AWS Graviton o no.

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

Amazon Linux 2

Es el valor predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo`C6g`,`M6g`,`R6g`, y`T4g`) y se puede usar para todos los tipos que no sean de instancias. GPU

Amazon Linux (2GPU)

Es el valor predeterminado para todas las familias de GPU instancias (por ejemplo, `P4` y`G4`) y se puede usar para todos los tipos de instancias que no estén AWS basados en Graviton.

Amazon Linux

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

> **Note**
>
> La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que desee utilizar para ese entorno informático. Por ejemplo, si su entorno informático usa A1 tipos de instancias, el recurso de cómputo AMI que elijas debe ser compatible Arm instancias. Amazon ECS vende ambos x86 y Arm

versiones de Amazon Linux 2 ECS optimizadas para AmazonAMI. Para obtener más información, consulte Amazon Linux 2 ECS optimizado para Amazon AMI en la Guía para desarrolladores de Amazon Elastic Container Service.

19. (Opcional) Para la plantilla de lanzamiento, selecciona una plantilla de EC2 lanzamiento de Amazon existente para configurar tus recursos informáticos. La versión predeterminada de la plantilla se rellena automáticamente. Para obtener más información, consulte Usa la plantilla de EC2 lanzamiento de Amazon con AWS Batch.

> **Note**
>
> En una plantilla de lanzamiento, puedes especificar una personalización AMI que hayas 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. Para el ID de Virtual Private Cloud (VPC), elige un AmazonVPC.

    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 informáticos deben comunicarse con el ECS VPC punto de conexión de Amazon a través de un VPC punto de conexión o de varias direcciones IP públicas. Para obtener más información, consulte los VPCpuntos finales de ECS la interfaz de Amazon (AWS PrivateLink). Si tu instancia no tiene un VPC punto

de conexión configurado o una dirección IP pública, puedes usar la traducción de direcciones de red (NAT). Para obtener más información sobreNAT, consulte NATlas puertas de enlace yCreación de una nube virtual privada .

c. En el caso de los grupos de EC2 seguridad, elige los grupos de seguridad de Amazon que quieres 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 Next (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 EC2 organización de Amazon, 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 Next (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 (#). El parámetro se asigna a `Image` la sección Crear un contenedor del Docker Remote API y al `IMAGE` parámetro de docker run.

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

ejemplo: Arm basado Docker las imágenes solo pueden ejecutarse en Arm recursos de cómputo basados.

- Las imágenes de los repositorios ECR públicos de Amazon utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o de `registry/repository[@digest]` nomenclatura (por ejemplo,`public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de ECR los repositorios de Amazon utilizan la convención de `registry/repository:tag` nomenclatura completa (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` la sección Crear un contenedor del Docker Remote API y el `COMMAND` parámetro a. docker run Para obtener más información sobre el `CMD` parámetro de Docker, consulta 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 Función de ejecución, especifica una IAM función que conceda permiso a los agentes de ECS contenedores de Amazon para realizar AWS API llamadas en tu nombre. Esta función utiliza los ECS IAM roles de Amazon para las tareas. Para obtener más información, consulte las IAMfunciones de ejecución de ECS tareas de Amazon en la Guía para desarrolladores de Amazon Elastic Container Service.

d. (Opcional) Para configurar el rol de trabajo, elija un IAM rol que tenga permisos para AWS APIs. Esta función utiliza los ECS IAM roles de Amazon para las tareas. Para obtener más

información, consulte IAMRoles 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 IAM rol para sus AWS Batch trabajos, consulte Creación de un IAM rol y una política 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 de configuración del entorno para vCPUs, especifique el número de vCPUs que desea reservar para el contenedor. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción a docker run. Cada v CPU equivale a 1024 accionesCPU.

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` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción a. docker run

h. En Número de GPUs, elija el número de GPUs 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` la sección Crear un contenedor del Docker Remote API y a la `--env` opción a 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 LogConfiguration: secretOptions.

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` la sección Crear un contenedor del Docker Remote API y a la `--user` opción a 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` la sección Crear un contenedor del Docker Remote API y a la `--privileged` opción a. 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:swappinessLinuxParameters.

   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, y. READWRITEMKNOD

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 Next (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 Next (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.

Tutorial: Cómo empezar con la AWS Batch orquestación de 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 Next (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. vCPUsEn Maximum, introduce el número máximo vCPUs que puede usar la instancia.
8. En la sección Configuración de red:

   a. Para el ID de Virtual Private Cloud (VPC), elige un AmazonVPC.

   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 informáticos deben comunicarse con el ECS VPC punto de conexión de Amazon a través de un VPC punto de conexión o de varias direcciones IP públicas. Para obtener más información, consulte los VPCpuntos finales de ECS la interfaz de Amazon (AWS PrivateLink). Si tu instancia no tiene un VPC punto de conexión configurado o una dirección IP pública, puedes usar la traducción de direcciones de red (NAT). Para obtener más información sobreNAT, consulte NATlas puertas de enlace yCreación de una nube virtual privada .

   c. En el caso de los grupos de EC2 seguridad, elige los grupos de seguridad de Amazon que quieres 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 Next (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 Next (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 Runtime, selecciona una LINUXo Windows.

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

   d. Para CPUla arquitectura, seleccione la CPU arquitectura 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 el rol de ejecución, elija un rol de ejecución de tareas que permita a los agentes de Amazon Elastic Container Service (AmazonECS) realizar AWS 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 (#). El parámetro se asigna a `Image` la sección Crear un contenedor del Docker Remote API y al `IMAGE` parámetro de docker run.

> **Note**
>
> Docker la arquitectura de la imagen debe coincidir con la arquitectura del procesador de los recursos informáticos en los que están programadas. Por ejemplo: Arm basado Docker las imágenes solo pueden ejecutarse en Arm recursos de cómputo basados.

- Las imágenes de los repositorios ECR públicos de Amazon utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o de `registry/repository[@digest]` nomenclatura (por ejemplo,`public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de ECR los repositorios de Amazon utilizan la convención de `registry/repository:tag` nomenclatura completa (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` la sección Crear un contenedor del Docker Remote API y el `COMMAND` parámetro a. docker run Para obtener más información sobre el `CMD` parámetro de Docker, consulta 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**
>
> Selecciona Información para revisar los ejemplos de código y de Bash. 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 IAM rol que otorgue permiso para usar el AWS APIs.

e. En la sección de configuración del entorno para vCPUs, especifique el número de vCPUs que desea reservar para el contenedor. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción a docker run. Cada v CPU equivale a 1024 accionesCPU.

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` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción a. 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` la sección Crear un contenedor del Docker Remote API y a la `--env` opción a. docker run

4. Elija Next (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 Next (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.

Tutorial: Cómo empezar AWS Batch en Amazon EKS

AWS Batch on Amazon EKS es un servicio gestionado para programar y escalar cargas de trabajo por lotes en EKS clústeres de Amazon existentes. AWS Batch no crea, administra ni realiza operaciones del ciclo de vida de tus EKS clústeres de Amazon en tu nombre. AWS Batch la orquestación amplía y reduce los nodos gestionados 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 de AWS Batch cómputo de tu clúster de AmazonEKS. AWS Batch Para funcionar de manera eficaz, su función vinculada al servicio necesita Kubernetes permisos de control de acceso basados en roles (RBAC) en tu clúster de Amazon EKS existente. Para obtener más información, consulte Uso de la RBAC autorización en la Kubernetes documentación.

AWS Batch requiere un Kubernetes espacio de nombres donde puede ubicar los pods como AWS Batch trabajos. Recomendamos un espacio de nombres dedicado para aislar los AWS Batch pods de las demás cargas de trabajo del clúster.

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

En las siguientes secciones se explica cómo configurar AWS Batch AmazonEKS.

Contenido

- Requisitos previos
- Prepara tu EKS clúster de Amazon para AWS Batch
- Cree un entorno EKS informático de Amazon
- Cree una cola de trabajos y adjunte el entorno informático
- Creación de una definición de trabajo
- Enviar un trabajo
- (Opcional) Envíe un trabajo con cambios

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 EKS los recursos de Amazon.

- AWS CLI— Una herramienta de línea de comandos para trabajar con AWS servicios, incluido AmazonEKS. 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 Kubernetes clústeres. Esta guía requiere que utilice la versión `1.23` o una posterior. Para obtener más información, consulta Instalación o actualización `kubectl` en la Guía del EKS usuario de Amazon.
- **`eksctl`**— Una herramienta de línea de comandos para trabajar con EKS clústeres de Amazon que automatiza muchas tareas individuales. Esta guía requiere que utilice la versión `0.115.0` o una posterior. Para obtener más información, consulta Instalación o actualización `eksctl` en la Guía del EKS usuario de Amazon.
- IAMPermisos necesarios: el IAM responsable de seguridad que utilices debe tener permisos para trabajar con las EKS IAM funciones de Amazon y las funciones vinculadas a servicios AWS CloudFormation, así como VPC con 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. IAM Debe completar todos los pasos de esta guía como el mismo usuario.

- Creación de un EKS clúster de Amazon: para obtener más información, consulta Cómo empezar con Amazon EKS `eksctl` en la Guía del EKS usuario de Amazon.

> **Note**
>
> AWS Batch solo admite EKS clústeres de Amazon con puntos finales de API servidor que tengan acceso público y accesibles a la Internet pública. De forma predeterminada, los puntos de enlace API del servidor de EKS clústeres de Amazon tienen acceso público. Para obtener más información, consulte el control de acceso al punto final del EKS clúster de Amazon en la Guía del EKS usuario de Amazon.

> **Note**
>
> AWS Batch no proporciona la organización de nodos gestionados para Core DNS u otros módulos de despliegue. Si necesitas CoreDNS, consulta Cómo añadir el EKS complemento Core DNS Amazon en la Guía del EKS usuario de Amazon. O bien, `eksctl create cluster create` úsalo para crear el clúster, que incluye Core de forma DNS predeterminada.

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

Prepara tu EKS clúster de Amazon para AWS Batch

Todos los pasos son obligatorios.

1. Crea 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 mediante un control de acceso basado en roles () RBAC

   Se utiliza para `kubectl` crear un Kubernetes rol del clúster para AWS Batch permitir vigilar nodos y pods y vincular el rol. Debe hacerlo una vez para cada EKS clúster.

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

```
$ 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
```

Crea un espacio de nombres con alcance Kubernetes función para gestionar los pods y AWS Batch su ciclo de vida y vincularlos. 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
```

Actualización Kubernetes `aws-authmapa` de configuración para asignar los RBAC permisos anteriores al rol vinculado al AWS Batch servicio.

```
$ 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 se `aws-service-role/batch.amazonaws.com/` ha eliminado del rol vinculado al ARN 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 sus ARN aws-authconfigmap.

Cree un entorno EKS informático de Amazon

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 instancias de los recursos informáticos (Kubernetes nodos) dentro de tu EKS clúster de Amazon. Se basa en la especificación de recursos de computación que se define al crear el entorno de computación. Puede utilizar instancias EC2 bajo demanda o instancias EC2 puntuales.

Ahora que el AWSServiceRoleForBatchel rol vinculado al servicio tiene acceso a tu EKS clúster de Amazon, puedes crear AWS Batch recursos. En primer lugar, cree un entorno informático que apunte a su EKS clúster de Amazon.

```
$ 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 y, a continuación, se AWS Batch utilizará el rol vinculado al servicio. AWS Batch on Amazon EKS solo admite la función AWS Batch vinculada al servicio.
- Solo `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, y las estrategias `SPOT_PRICE_CAPACITY_OPTIMIZED` de asignación son compatibles con los entornos EKS informáticos de Amazon.

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

- Para ello`instanceRole`, consulte Crear el IAM rol de EKS nodo de Amazon y Habilitar el acceso IAM principal a su clúster en la Guía del EKS usuario de Amazon. Si utilizas redes de pods, consulta Configuración del VPC CNI complemento Amazon para Kubernetes para usar IAM roles para las cuentas de servicio en la Guía del EKS usuario de Amazon.
- Una forma de hacer que las subredes funcionen para el `subnets` parámetro es utilizar las subredes públicas de los grupos de nodos EKS gestionados por Amazon que se crearon `eksctl` al crear un clúster de AmazonEKS. De lo contrario, utilice subredes que tengan una ruta de red que permita extraer imágenes.
- El `securityGroupIds` parámetro puede usar el mismo grupo de seguridad que el EKS clúster de Amazon. 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 EKS informático de Amazon es una responsabilidad compartida. Para obtener más información, consulte Responsabilidad compartida del Kubernetes nodos.

> **Important**
>
> Es importante confirmar que el entorno de computación está en buen estado antes de continuar. La DescribeComputeEnvironmentsAPIoperación se puede utilizar para ello.
>
> ```
> $ 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.

Cree una cola de trabajos y adjunte el entorno informático

```
$ 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 EKS clúster de Amazon asociado a tu 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
```

Creación de 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 CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS.

Enviar 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 CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS.
- Para obtener más información sobre la ejecución de tareas en EKS los recursos de Amazon, consulteEKSEmpleos en Amazon.

(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 a Kubernetes. Para examinar los detalles de un trabajo, se debe configurar el registro. Para obtener más información, consulte Usa CloudWatch Logs para monitorear AWS Batch los EKS trabajos de Amazon.
- Para mejorar la visibilidad de los detalles de las operaciones, habilite el registro del plano EKS de control de Amazon. Para obtener más información, consulta el registro del plano EKS de control de Amazon en la Guía del EKS usuario de Amazon.
- Daemonsets y kubelets La sobrecarga afecta a los recursos de memoria CPU y v disponibles, específicamente al escalamiento y a la colocación laboral. Para obtener más información, consulte CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS.

Tutorial: Cómo empezar a utilizar 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 tu EKS clúster privado de Amazon existente con el AWS Batch fin de ejecutar tus trabajos a escala. Puede utilizar `eksctl`(una interfaz de línea de comandos para AmazonEKS), la AWS consola o la AWS Command Line Interfacepara crear un EKS clúster de Amazon privado con todos los demás recursos necesarios. Por lo general, el AWS Batch soporte para EKS clústeres privados de Amazon está disponible en formato comercial, Regiones de AWS donde AWS Batch esté disponible.

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

Cada subred debe tener asociada una tabla de enrutamiento explícita porque `eksctl` no modifica la tabla de enrutamiento principal. Tu clúster debe extraer imágenes de un registro de contenedores

que esté en tu AmazonVPC. Además, puedes crear un Amazon Elastic Container Registry en tu Amazon VPC y copiar en él las imágenes del contenedor para que las extraigan tus nodos. Para obtener más información, consulte Copiar una imagen de contenedor de un repositorio a otro repositorio. Para empezar a usar los repositorios ECR privados de Amazon, consulta Repositorios ECR privados de Amazon.

Si lo desea, puede crear una regla de extracción de caché con AmazonECR. Una vez creada una regla de extracción de caché para un registro público externo, puedes extraer una imagen de ese registro público externo con el identificador de recursos uriforme () del registro ECR privado de Amazon. URI A continuación, Amazon ECR crea un repositorio y guarda la imagen en caché. Cuando se extrae una imagen en caché mediante el registro ECR privado de AmazonURI, Amazon ECR comprueba el registro remoto para ver si hay una nueva versión de la imagen y actualiza tu registro privado hasta una vez cada 24 horas.

Contenido

- Requisitos previos
- Prepare su EKS clúster para AWS Batch
- Cree un entorno EKS informático de Amazon
- Cree una cola de trabajos y adjunte el entorno informático
- Creación de una definición de trabajo
- Enviar un trabajo
- (Opcional) Envíe un trabajo con cambios
- Resolución 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 EKS los recursos de Amazon. También debe crear todos los recursos necesariosVPC, incluidas las subredes, las tablas de rutas, los puntos de conexión y el VPC clúster de Amazon. EKS Debe usar el. AWS CLI

- AWS CLI— Una herramienta de línea de comandos para trabajar con AWS servicios, incluido AmazonEKS. 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 con la que trabajar Kubernetes clústeres. Esta guía requiere que utilice la versión `1.23` o una posterior. Para obtener más información, consulta Instalación o actualización `kubectl` en la Guía del EKS usuario de Amazon.
- **`eksctl`**— Una herramienta de línea de comandos para trabajar con EKS clústeres de Amazon que automatiza muchas tareas individuales. Esta guía requiere que utilice la versión `0.115.0` o una posterior. Para obtener más información, consulta Instalación o actualización `eksctl` en la Guía del EKS usuario de Amazon.
- Permisos obligatorios AWS Identity and Access Management (IAM): el IAM responsable de seguridad que utilices debe tener permisos para trabajar con las EKS IAM funciones de Amazon y las funciones vinculadas a servicios AWS CloudFormation, así como VPC con 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. IAM Debe completar todos los pasos de esta guía como el mismo usuario.
- Creación de un EKS clúster de Amazon: para obtener más información, consulta Cómo empezar con Amazon EKS `eksctl` en la Guía del EKS usuario de Amazon.

> **Note**
>
> AWS Batch no proporciona una organización de nodos gestionados para Core DNS u otros módulos de despliegue. Si necesitas CoreDNS, consulta Cómo añadir el EKS complemento Core DNS Amazon en la Guía del EKS usuario de Amazon. O bien, úselo `eksctl create cluster create` para crear el clúster, que incluye Core de forma DNS predeterminada.

- Permisos: los usuarios que llaman a la CreateComputeEnvironmentAPIoperación para crear un entorno informático que utilice EKS los recursos de Amazon necesitan permisos para la `eks:DescribeCluster` API operación. El uso de AWS Management Console para crear un recurso de cómputo con EKS los recursos de Amazon requiere permisos `eks:DescribeCluster` tanto para como para`eks:ListClusters`.
- Cree un EKS clúster privado en la región us-east-1 con el archivo de configuración de `eksctl` 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 VPC interfaz que necesite. Para obtener más información, consulte Requisitos de los clústeres privados.

Prepare su EKS clúster 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 mediante un control de acceso basado en roles () RBAC

Se utiliza para `kubectl` crear un Kubernetes rol del clúster para AWS Batch permitir vigilar nodos y pods y vincular el rol. Debes hacerlo una vez para cada EKS clúster de Amazon.

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

```
$ 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 espacio de nombres con alcance Kubernetes función para administrar y mantener el ciclo AWS Batch de vida de los pods y vincularlos. 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
```

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

```
$ 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 se `aws-service-role/batch.amazonaws.com/` ha eliminado del rol vinculado al ARN 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 sus ARN aws-authconfigmap.

Cree un entorno EKS informático de Amazon

AWS Batch los entornos de cómputo definen los parámetros de los recursos de cómputo 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 instancias de los recursos informáticos (Kubernetes nodos) dentro de tu EKS clúster de Amazon. Se basa en la especificación de recursos de computación que se define al crear el entorno de computación. Puede utilizar instancias EC2 bajo demanda o instancias EC2 puntuales.

Ahora que el AWSServiceRoleForBatchel rol vinculado al servicio tiene acceso a tu EKS clúster de Amazon, puedes crear AWS Batch recursos. En primer lugar, cree un entorno informático que apunte a su EKS clúster de Amazon.

```
$ 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 y, a continuación, se AWS Batch utilizará el rol vinculado al servicio. AWS Batch on Amazon EKS solo admite la función AWS Batch vinculada al servicio.
- Solo `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, y las estrategias de `SPOT_PRICE_CAPACITY_OPTIMIZED` asignación son compatibles con los entornos de EKS cómputo de Amazon.

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

- Para ello`instanceRole`, consulte Crear el IAM rol de EKS nodo de Amazon y Habilitar el acceso IAM principal a su clúster en la Guía del EKS usuario de Amazon. Si utilizas redes de pods, consulta Configuración del VPC CNI complemento Amazon para Kubernetes para usar IAM roles para las cuentas de servicio en la Guía del EKS usuario de Amazon.
- Una forma de hacer que las subredes funcionen para el `subnets` parámetro es utilizar las subredes públicas de los grupos de nodos EKS gestionados por Amazon que se crearon `eksctl` al crear un clúster de AmazonEKS. De lo contrario, utilice subredes que tengan una ruta de red que permita extraer imágenes.
- El `securityGroupIds` parámetro puede usar el mismo grupo de seguridad que el EKS clúster de Amazon. 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 EKS informático de Amazon es una responsabilidad compartida. Para obtener más información, consulta Seguridad en Amazon EKS.

> **Important**
>
> Es importante confirmar que el entorno de computación está en buen estado antes de continuar. La DescribeComputeEnvironmentsAPIoperación se puede utilizar para ello.

```
$ 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.

Cree una cola de trabajos y adjunte el entorno informático

```
$ 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 han unido al EKS clúster de Amazon asociado a tu entorno de cómputo.

```
$ 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
```

Creación de 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 ECR repositorio público, proporciona el enlace a la imagen almacenada en nuestro ECR repositorio privado. Consulta 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, necesitarás acceso privado a tu clúster de Amazon. EKS Esto significa que todo el tráfico al API servidor de tu clúster debe provenir del clúster VPC o de una red conectada.

Enviar 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 CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS.
- Para obtener más información sobre la ejecución de tareas en EKS los recursos de Amazon, consulteEKSEmpleos en Amazon.

(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 a Kubernetes. Para examinar los detalles de un trabajo, se debe configurar el registro. Para obtener más información, consulte Usa CloudWatch Logs para monitorear AWS Batch los EKS trabajos de Amazon.

- Para mejorar la visibilidad de los detalles de las operaciones, habilite el registro del plano EKS de control de Amazon. Para obtener más información, consulta el registro del plano EKS de control de Amazon en la Guía del EKS usuario de Amazon.
- Daemonsets y kubelets La sobrecarga afecta a los recursos de memoria CPU y v disponibles, específicamente al escalamiento y a la colocación laboral. Para obtener más información, consulte CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS.

Resolución de problemas

Si los nodos lanzados por AWS Batch no tienen acceso al ECR repositorio de Amazon (o a ningún otro repositorio) que almacena tu imagen, tus trabajos podrían permanecer en ese STARTING estado. Esto se debe a que el pod no podrá descargar la imagen ni ejecutar tu 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, consulta ¿Cómo soluciono los problemas del estado del pod en Amazon? EKS .

Entornos informáticos para AWS Batch

Las colas de trabajos se asignan a uno o varios entornos de computación. Los entornos informáticos contienen las instancias de ECS contenedor de Amazon 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
- Entornos de computación no administrados
- Crear un entorno de computación
- Recurso informático AMIs
- Usa la plantilla de EC2 lanzamiento de Amazon con AWS Batch
- EC2configuraciones
- Estrategias de asignación de tipos de instancia para AWS Batch
- Administración de memoria de recursos informáticos
- Actualizar entornos de computación
- Entornos de cómputo Fargate
- Entornos de EKS cómputo de Amazon

Entornos de computación administrados

Puede utilizar un entorno informático gestionado para AWS Batch gestionar la capacidad y los tipos de instancias 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 elegir entre utilizar Amazon EC2 On-Demand Instances o Amazon EC2 Spot Instances. 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 procesamiento gestionado lanzan EC2 instancias de Amazon en las subredes VPC y subredes que especifique y, a continuación, las registran en un ECS clúster de Amazon. Las EC2 instancias de Amazon necesitan acceso a una red externa para comunicarse con el punto de conexión del ECS servicio de Amazon. Algunas subredes no proporcionan direcciones IP públicas a EC2 las instancias de Amazon. Si tus EC2 instancias de Amazon no tienen una dirección IP pública, deben usar la traducción de direcciones de red (NAT) para obtener este acceso. Para obtener más información, consulta NATlas pasarelas en la Guía del VPC usuario de Amazon. Para obtener más información sobre cómo crear unaVPC, consulteCreación de una nube virtual privada .

De forma predeterminada, los entornos informáticos AWS Batch gestionados utilizan una versión reciente y aprobada de Amazon ECS optimizada AMI para recursos informáticos. Sin embargo, es posible que desee crear la suya propia AMI para utilizarla en sus entornos de cómputo gestionados por varios motivos. Para obtener más información, consulte Recurso informático AMIs.

> **Note**
>
> AWS Batch no se actualiza automáticamente AMIs en un entorno informático una vez creado. Por ejemplo, no actualiza el AMIs entorno informático cuando se lanza una versión más reciente de Amazon ECS OptimizedAMI. 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 usar una nueva AMI para sus AWS Batch trabajos. El método original consiste en completar estos pasos:
>
> 1. Cree un nuevo entorno informático con el nuevoAMI.
> 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 de computación. Para utilizar la actualización mejorada de los entornos de cómputo para actualizarAMIs, siga estas reglas:

- No establezcas 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 especifiques un AMI ID en `imageId`, `imageIdOverride`(in `ec2Configuration`) o en la plantilla de lanzamiento (`launchTemplate`). En ese caso, AWS Batch selecciona la última versión ECS optimizada de Amazon compatible AWS Batch en el momento en AMI que se inicia la actualización de la infraestructura. Como alternativa, puede especificar el AMI ID en los `imageIdOverride` parámetros `imageId` o, o bien, la plantilla de lanzamiento identificada por las `LaunchTemplate` propiedades. El cambio de cualquiera de estas propiedades inicia una actualización de la infraestructura. Si el AMI ID se especifica en la plantilla de lanzamiento, no se puede reemplazar especificando un AMI ID en `imageIdOverride` los parámetros `imageId` o. 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 infraestructura provocará que se vuelva a seleccionar el AMI ID. 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) y no relacionados con varios nodos. 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 informático gestionado, si especificas un `minvCpu` valor superior a cero, se AWS Batch crea un grupo de instancias para usarlo únicamente con tareas que no sean MNP tareas. 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. Debes comprobar que lo que AMI utilizas para tus recursos informáticos cumple con la AMI especificación de la instancia de ECS contenedor de Amazon. Para obtener más información, consulte Especificación de recursos AMI informáticos y Tutorial: Crear un recurso de cómputo AMI.

> **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 DescribeComputeEnvironmentsAPIoperación para ver los detalles del entorno informático. Busca el ECS clúster de Amazon asociado al entorno y, a continuación, lanza manualmente las instancias de contenedor en ese ECS clúster de Amazon.

El siguiente AWS CLI comando también proporciona el ECS clúster de AmazonARN.

```
$ aws batch describe-compute-environments \
    --compute-environments unmanagedCE \
    --query "computeEnvironments[].ecsClusterArn"
```

Para obtener más información, consulta Cómo lanzar una instancia de ECS contenedor de Amazon en la Guía para desarrolladores de Amazon Elastic Container Service. Cuando lance sus recursos informáticos, especifique el ECS clúster de Amazon ARN que registran los recursos con los siguientes datos de EC2 usuario de Amazon. Reemplazar *ecsClusterArn* con el clúster ARN que obtuviste con el comando anterior.

```
#!/bin/bash
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

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 EC2 instancias de Amazon o los recursos de AWS Fargate dentro del entorno en función de sus especificaciones. O bien, puede crear un entorno informático no gestionado en el que gestione la configuración de la EC2 instancia de Amazon dentro del entorno.

> **Important**
>
> Las instancias de spot de Fargate no se admiten en los siguientes escenarios:
>
> - En contenedores Amazon Linux con ARM64 arquitectura.
> - 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.

Temas

- Tutorial: Cree un entorno informático gestionado con los recursos de Fargate
- Tutorial: Creación de un entorno informático gestionado con EC2 los recursos de Amazon
- Tutorial: Creación de un entorno informático no gestionado con los recursos de Amazon EC2
- Tutorial: Creación de un entorno informático gestionado con EKS los recursos de Amazon
- Recurso: plantilla de entorno informático

Tutorial: Cree un entorno informático gestionado con los recursos de Fargate

Complete los siguientes pasos para crear un entorno informático gestionado con recursos. 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, elija un rol vinculado al servicio que permita al AWS Batch servicio realizar llamadas a las AWS API operaciones requeridas en su 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, consulta Uso de Amazon EC2 Spot y Fargate_. SPOT
   b. vCPUsEn Maximum, seleccione la cantidad máxima a la vCPUs que puede ampliarse su entorno informático, independientemente de la demanda de colas de trabajos.

c. Seleccione Página siguiente.

7. Configure redes.

> **Important**
>
> Los recursos informáticos necesitan acceso para comunicarse con el punto final del ECS servicio de Amazon. Esto puede hacerse a través de un VPC punto final de interfaz o a través de los recursos de cómputo de que tienen direcciones IP públicas.
> Para obtener más información sobre los VPC puntos de enlace de la interfaz, consulte Amazon ECS Interface VPC Endpoints (AWS PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.
> Si no tiene configurado un VPC punto final de interfaz y los recursos informáticos de sus no tienen direcciones IP públicas, deberán utilizar la traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulta NATIas pasarelas en la Guía del VPC usuario de Amazon . Para obtener más información, consulte the section called “Tutorial: Crear un VPC”.

a. Para el ID de nube privada virtual (VPC), elija el VPC lugar en el que desee lanzar las instancias.

b. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, VPC están disponibles todas las subredes de la seleccionada.

> **Note**
>
> AWS Batch on Fargate no es compatible actualmente con las Zonas Locales. Para obtener más información, consulte ECSIos clústeres de Amazon en Zonas Locales, 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 elige el grupo de seguridad predeterminado para ustedVPC.

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.

Tutorial: Creación de un entorno informático gestionado con EC2 los recursos de Amazon

Complete los siguientes pasos para crear un entorno de procesamiento administrado con los recursos de Amazon Elastic Compute Cloud EC2 (Amazon).

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.

   a. Para configurar el entorno de cómputo, elija Amazon Elastic Compute Cloud (AmazonEC2).

   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. En la función de servicio, elija una función vinculada al servicio que permita al AWS Batch servicio realizar llamadas a las AWS API operaciones requeridas en su nombre. En este ejemplo, elija AWSServiceRoleForBatch. Para obtener más información, consulte Permisos de rol vinculados al servicio para AWS Batch.

   e. En el caso del rol de instancia, elija crear un nuevo perfil de instancia o usar un perfil de instancia existente que tenga adjuntos los IAM permisos necesarios. Este perfil de instancia permite a las instancias de ECS contenedor de Amazon que se crean para su entorno de cómputo realizar llamadas a las AWS API operaciones requeridas en su nombre. Para obtener más información, consulte Función de ECS instancia de Amazon. 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 EC2etiquetas, selecciona Añadir etiqueta para añadir una etiqueta a los recursos que se lanzan en el entorno informático. 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 Etiquete sus AWS Batch recursos.

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 tu precio máximo es del 20%, el precio spot debe ser inferior al 20% del precio bajo demanda actual para esa EC2 instancia. 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) Para el rol de flota puntual, elija un rol existente en Amazon EC2 Spot Fleet IAM para aplicarlo a su entorno de cómputo puntual. Si aún no tienes un IAM rol en Amazon EC2 Spot Fleet, primero debes crear uno. Para obtener más información, consulte Función de Amazon EC2 en la flota spot.

      > **Important**
      >
      > Para etiquetar sus instancias puntuales al crearlas, su IAM rol en Amazon EC2 Spot Fleet debe usar la nueva política EC2SpotFleetTaggingRole gestionada por Amazon. La política EC2SpotFleetRole gestionada por Amazon no tiene los permisos necesarios para etiquetar instancias puntuales. Para obtener más información, consulte Instancias de spot no etiquetadas en el momento de su creación y the section called “Etiquetar los recursos”.

   d. vCPUsEn Mínimo, elija el número mínimo vCPUs que mantenga su entorno informático, independientemente de la demanda de colas de trabajos.

   e. En Desired vCPUs, elija el número con el vCPUs que se inicia su entorno de cómputo. A medida que aumente la demanda de colas de trabajos, AWS Batch puede aumentar la cantidad deseada de ellas vCPUs en su entorno informático y añadir EC2 instancias, hasta el máximovCPUs. A medida que la demanda disminuya, AWS Batch puede reducir la cantidad deseada de instancias vCPUs en su entorno de cómputo y eliminar al mínimovCPUs.

   f. vCPUsEn Máximo, selecciona el número máximo al vCPUs que se puede escalar tu entorno de cómputo, independientemente de la demanda de colas de trabajos.

g. En Tipos de instancias permitidos, selecciona los tipos de EC2 instancias de Amazon 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 puedes elegir seleccionar `optimal` los tipos de instancia (de C4, M4, y R4 familias de instancias) que se adapten a la demanda de sus 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 puedes mezclar instancias x86 e ARM instancias en el mismo entorno de cómputo.

> **Note**
>
> AWS Batch escalará en GPUs función de la cantidad requerida en tus colas de trabajos. Para utilizar la GPU programación, el entorno informático debe incluir los tipos de instancias de las `g5` familias `p2` `p3p4,p5,g3,g3s,g4`, o.

> **Note**
>
> Actualmente, `optimal` utiliza tipos de instancias de C4, M4, y R4 familias de instancias. Regiones de AWS Eso no tiene tipos de instancias de esas familias de instancias, tipos de instancia de C5, M5, y R5 se utilizan familias de 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) Para el par de EC2 claves, elige un par de claves pública y privada como credenciales de seguridad cuando te conectes a la instancia. Para obtener más información sobre los pares de EC2 claves de Amazon, consulta Pares de EC2 claves de Amazon 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 cómputo EC2 bajo demandaOPTIMIZED, y SPOTCAPACITY_ _ SPOT PRICECAPACITY_ OPTIMIZED para los entornos de cómputo EC2 puntual. Para obtener más información, consulte the section called “Estrategias de asignación de tipos de instancia”.

l. (Opcional) Para la EC2configuración, elija los valores de anulación de tipo de imagen e ID de imagen AWS Batch para proporcionar información y seleccionar Amazon Machine Images (AMIs) 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 imagen ECSoptimizada recientemente para AMI Amazon. Si no se especifica ningún tipo de imagen, el valor predeterminado es una instancia de Amazon Linux 2 para instancias que no GPU sean AWS Graviton o no.

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

Amazon Linux 2

Es el valor predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo`C6g`,`M6g`,`R6g`, y`T4g`) y se puede usar para todos los tipos que no sean de instancias. GPU

Amazon Linux (2GPU)

Es el valor predeterminado para todas las familias de GPU instancias (por ejemplo, `P4` y`G4`) y se puede usar para todos los tipos de instancias que no estén AWS basados en Graviton.

Amazon Linux

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

> **Note**
>
> La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que desee utilizar para ese entorno informático. Por ejemplo, si su entorno de cómputo usa A1 tipos de instancias, el recurso de cómputo AMI que elijas debe ser compatible Arm instancias. Amazon ECS vende ambos x86 y Arm versiones de Amazon Linux 2 ECS optimizadas para AmazonAMI. Para obtener más información, consulte Amazon Linux 2 ECS optimizado para Amazon AMI en la Guía para desarrolladores de Amazon Elastic Container Service.

m. (Opcional) Para la plantilla de lanzamiento, selecciona una plantilla de EC2 lanzamiento de Amazon existente para configurar tus recursos informáticos. La versión predeterminada de la plantilla se rellena automáticamente. Para obtener más información, consulte Usa la plantilla de EC2 lanzamiento de Amazon con AWS Batch.

> **Note**
>
> En una plantilla de lanzamiento, puedes especificar una personalización AMI que hayas 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 AMI ID diferente de forma predeterminada o se selecciona la última versión de la plantilla de lanzamiento, ese AMI ID se utilizará en la actualización. Para obtener más información, consulte the section called “Actualizar el AMI ID”.

o. Seleccione Página siguiente.

7. En la sección Configuración de red:

> **Important**
>
> Los recursos informáticos necesitan acceso para comunicarse con el punto final del ECS servicio de Amazon. Esto puede hacerse a través de un VPC punto final de interfaz o a través de los recursos de cómputo de que tienen direcciones IP públicas.
> Para obtener más información sobre los VPC puntos de enlace de la interfaz, consulte Amazon ECS Interface VPC Endpoints (AWS PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.
> Si no tiene configurado un VPC punto final de interfaz y los recursos informáticos de sus no tienen direcciones IP públicas, deberán utilizar la traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulta NATlas pasarelas en la Guía del VPC usuario de Amazon . Para obtener más información, consulte the section called “Tutorial: Crear un VPC”.

a. Para el ID de Virtual Private Cloud (VPC), elige un VPC lugar donde lanzar tus instancias.

b. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, VPC están disponibles todas las subredes de la seleccionada.

> **Note**
>
> AWS Batch en Amazon EC2 admite Zonas Locales. Para obtener más información, consulte Zonas locales en la Guía del EC2 usuario de Amazon y ECSClústeres de Amazon en Local Zones, Wavelength Zones 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 elige el grupo de seguridad predeterminado para ustedVPC.

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.

Tutorial: Creación de un entorno informático no gestionado con los recursos de Amazon EC2

Complete los siguientes pasos para crear un entorno informático no gestionado con los recursos de Amazon Elastic Compute Cloud (AmazonEC2).

1. Abra la consola en AWS Batch . 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 configurar el entorno de cómputo, elija Amazon Elastic Compute Cloud (AmazonEC2).
   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. En Función de servicio, elija una función que permita al AWS Batch servicio realizar llamadas a las AWS API operaciones requeridas en su nombre.

   > **Note**
   > No se puede utilizar `AWSServiceRoleForBatch` en entornos informáticos no gestionados.

7. vCPUsEn Maximum, elige el número máximo al vCPUs que se puede escalar tu entorno de cómputo, independientemente de la demanda de 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 Etiquete sus AWS Batch recursos.
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.

Tutorial: Creación de un entorno informático gestionado con EKS los recursos de Amazon

Complete los siguientes pasos para crear un entorno informático gestionado con los recursos de Amazon Elastic Kubernetes Service (Amazon). EKS

1. Abra la consola en. AWS Batch 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 informático, 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 el caso del rol de instancia, elige un perfil de instancia existente que tenga adjuntos los permisos necesariosIAM.

   > **Note**
   >
   > Para crear un entorno informático en la AWS Batch consola, elige un perfil de instancia que tenga los `eks:DescribeCluster` permisos `eks:ListClusters` y.

8. Para el EKSclúster, elige un EKS clúster de Amazon existente.
9. Para Namespace, introduce un Kubernetes espacio de nombres para agrupar los AWS Batch procesos 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) Para usar instancias EC2 puntuales, active Habilitar el uso de instancias puntuales para usar instancias EC2 puntuales de Amazon.
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 su precio máximo es del 20%, el precio puntual debe ser inferior al 20% del precio bajo demanda actual de esa EC2 instancia. 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) Para el rol de flota de spot, elija el rol de flota IAM de Amazon EC2 Spot para el entorno de `SPOT` cómputo.

> **Important**
>
> Este rol es necesario si la estrategia de asignación está establecida en `BEST_FIT` o no se ha especificado.

15. (Opcional) Como mínimo vCPUs, elija el número mínimo vCPUs que mantenga su entorno informático, independientemente de la demanda de colas de trabajos.
16. (Opcional) En el campo Máximo vCPUs, selecciona el número máximo al vCPUs que puede ampliarse tu entorno informático, independientemente de la demanda de colas de trabajos.
17. En Tipos de instancias permitidos, selecciona los tipos de EC2 instancias de Amazon 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 puedes elegir seleccionar `optimal` los tipos de instancia (de C4, M4, y R4 familias de instancias) porque necesita que se adapten a la demanda de sus 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 puedes mezclar x86 e ARM instancias en el mismo entorno de cómputo.

> **Note**
>
> AWS Batch escala GPUs en función de la cantidad requerida en sus colas de trabajos. Para utilizar la GPU programación, el entorno informático debe incluir tipos de instancias de las `g5` familias `p2` `p3p4,p5,g3,g3s,g4`, o.

> **Note**
>
> Actualmente, `optimal` utiliza tipos de instancia de las familias de instancias C4, M4 y R4. En Regiones de AWS ese caso, no hay tipos de instancias de esas familias de instancias, sino tipos de instancias de C5, M5, y R5 se utilizan familias de 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, elige BEST_ FIT _ PROGRESSIVE.

    c. (Opcional) Para la configuración de Amazon Machine Images (AMIs), seleccione la configuración Añadir imágenes de Amazon Machine Images (AMI). A continuación, elija un tipo de imagen, introduzca una anulación del ID de imagen y Kubernetes versión.

    > **Important**
    >
    > Para utilizar una imagen personalizadaAMI, selecciona el tipo de imagen y, a continuación, introduce el AMI identificador personalizado en el cuadro de anulación del identificador de imagen.

    > **Note**
    >
    > Si no se especifica la anulación del ID de imagen para cada tipo de imagen, AWS Batch selecciona una imagen ECSoptimizada recientemente para AMI Amazon. Si no se especifica ningún tipo de imagen, el valor predeterminado es una instancia de Amazon Linux 2 para instancias que no GPU sean AWS Graviton o no.
    >
    > Amazon Linux 2
    >
    > Es el valor predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo`C6g`,`M6g`,`R6g`, y`T4g`) y se puede usar para todos los tipos que no sean de instancias. GPU

Amazon Linux (2GPU)

Es el valor predeterminado para todas las familias de GPU instancias (por ejemplo, `P4` y`G4`) y se puede usar para todos los tipos de instancias que no estén AWS basados en 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. En el caso del ID de nube privada virtual (VPC), elige un VPC lugar donde lanzar las instancias.
21. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, VPC están disponibles todas las subredes de la seleccionada.

> **Note**
>
> AWS Batch en Amazon EKS admite Zonas Locales. Para obtener más información, consulte Amazon EKS y las Zonas AWS Locales en la Guía del EKS usuario de Amazon.

22. (Opcional) En Grupos de seguridad, seleccione su grupo de seguridad para asociarlo a las instancias. De forma predeterminada, se selecciona el grupo de seguridad predeterminado para ustedVPC.
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.

Recurso: plantilla de entorno informático

En el siguiente ejemplo, se muestra una plantilla de entorno de computación vacía. Puede usar esta plantilla para crear su entorno informático que, a continuación, se puede guardar en un archivo y utilizar con la AWS CLI `--cli-input-json` opción. Para obtener más información sobre estos parámetros, consulte CreateComputeEnvironmentla AWS Batch APIReferencia.

> **Note**
>
> Puede generar una plantilla de entorno informático con el siguiente AWS CLI comando.
>
> ```
> $ aws batch create-compute-environment --generate-cli-skeleton
> ```

```
{
    "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": ""
    }
}
```

Recurso informático AMIs

De forma predeterminada, los entornos de cómputo AWS Batch gestionados utilizan una versión reciente y aprobada de Amazon ECS optimizada AMI para recursos de cómputo. Sin embargo, es posible que desee crear la suya propia AMI para utilizarla en sus entornos de cómputo gestionados y no gestionados. Si necesita alguno de los siguientes requisitos, le recomendamos que cree el suyo propioAMI:

- Aumentar el tamaño de almacenamiento de sus volúmenes AMI raíz o de datos
- Añadir volúmenes de almacenamiento de instancias para los tipos de EC2 instancias de Amazon compatibles
- Personalización del agente de ECS contenedores de Amazon
- Personalización de Docker
- Configurar una GPU carga de trabajo AMI para permitir que los contenedores accedan al GPU hardware en los tipos de EC2 instancias de Amazon compatibles

> **Note**
>
> Una vez creado un entorno informático, AWS Batch no actualiza AMIs el entorno informático. AWS Batch tampoco lo actualiza AMIs en su entorno informático cuando hay disponible una versión más reciente de Amazon ECS OptimizedAMI. 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 nueva AMI en sus AWS Batch trabajos, haga lo siguiente:

1. Cree un nuevo entorno de cómputo con el nuevoAMI.
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 de computación. Para utilizar la actualización mejorada de los entornos de cómputo para actualizarAMIs, siga estas reglas:

- No establezcas 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 especifiques un AMI ID en `imageId`, `imageIdOverride`(in `ec2Configuration`) o en la plantilla de lanzamiento (`launchTemplate`). Si no especificas un AMI ID, AWS Batch selecciona la última versión ECS optimizada de Amazon AWS Batch compatible en el momento en AMI que se inicia la actualización de la infraestructura. Como alternativa, puede especificar el AMI ID en los `imageIdOverride` parámetros `imageId` o. 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 AMI ID se especifica en la plantilla de lanzamiento, no se puede reemplazar especificando un AMI ID en `imageIdOverride` los parámetros `imageId` o. AMI El AMI ID 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 AMI ID se puede reemplazar configurando una nueva versión

predeterminada para la plantilla de lanzamiento (si`$Default`) o añadiendo una nueva versión a la plantilla de lanzamiento (si`$Latest`).

Si se siguen estas reglas, cualquier actualización que inicie una actualización de infraestructura provocará que se vuelva a seleccionar el AMI ID. 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

- Especificación de recursos AMI informáticos
- Tutorial: Crear un recurso de cómputo AMI
- Usa una GPU carga de trabajo AMI
- Obsolescencia de Amazon Linux

Especificación de recursos AMI informáticos

La AMI especificación básica del recurso AWS Batch informático consiste en lo siguiente:

Obligatorio

- Una distribución de Linux moderna que ejecuta al menos la versión 3.10 del núcleo de Linux en un tipo AMI de HVM virtualización. 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. Le recomendamos que utilice el Amazon ECS Optimized predeterminado AMI al crear su entorno de cómputo. Para ello, no especifique ninguna personalizaciónAMI. Para obtener más información, consulte Trabajos paralelos de varios nodos.

- El agente de ECS contenedores de Amazon. Le recomendamos que utilice la última versión de . Para obtener más información, consulte Instalación del Amazon ECS Container Agent en la Guía para desarrolladores de Amazon Elastic Container Service.
- El controlador de `awslogs` registro debe especificarse como un controlador de registro disponible con la variable de `ECS_AVAILABLE_LOGGING_DRIVERS` entorno cuando se inicie el agente ECS contenedor de Amazon. Para obtener más información, consulte Configuración de Amazon ECS Container Agent 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**
>
> Te recomendamos la versión de Docker que viene con la versión del ECS agente de Amazon correspondiente que estés utilizando y que se haya probado con ella. Amazon ECS proporciona un registro de cambios para la variante Linux de Amazon ECS AMI - optimized on. GitHub Para obtener más información, consulte Changelog (Registro de cambios).

Recomendado

- Un proceso de inicialización y mantenimiento para ejecutar y monitorear el agente de AmazonECS. El ECS optimizado de Amazon AMI usa el proceso `ecs-init` inicial y otros sistemas operativos podrían usarlo`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 `ecs-init`proyecto en GitHub. Como mínimo, los entornos de cómputo gestionado requieren que el ECS agente de Amazon se inicie al arrancar. Si el ECS agente de Amazon no se está ejecutando en tu recurso informático, no podrá aceptar trabajos de AWS Batch.

El Amazon ECS Optimized AMI viene preconfigurado con estos requisitos y recomendaciones. Le recomendamos que utilice Amazon ECS Optimized AMI o Amazon Linux AMI con el `ecs-init` paquete que está instalado para sus recursos informáticos. Elige otro AMI si tu aplicación requiere un sistema operativo específico o una versión de Docker que aún no esté disponible en esos AMIs

sistemas. Para obtener más información, consulte Amazon ECS -Optimized AMI en la Guía para desarrolladores de Amazon Elastic Container Service.

Tutorial: Crear un recurso de cómputo AMI

Puede crear su propio recurso informático personalizado AMI para usarlo en sus entornos informáticos gestionados y no gestionados. Para obtener instrucciones, consulte la Especificación de recursos AMI informáticos. A continuación, después de crear una personalizadaAMI, puede crear un entorno informático que lo utilice y al AMI que pueda asociar una cola de trabajos. Por último, comience a enviar trabajos a esa cola.

Para crear un recurso informático personalizado AMI

1. Elija una base desde AMI la que empezar. La base AMI debe utilizar HVM la virtualización. La base no AMI puede ser un WindowsAMI.

   > **Note**
   >
   > La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que desee utilizar para ese entorno informático. Por ejemplo, si su entorno de cómputo usa A1 tipos de instancias, el recurso de cómputo AMI que elijas debe ser compatible Arm instancias. Amazon ECS vende ambos x86 y Arm versiones de Amazon Linux 2 ECS optimizadas para AmazonAMI. Para obtener más información, consulte Amazon Linux 2 ECS optimizado para Amazon AMI en la Guía para desarrolladores de Amazon Elastic Container Service.

   Amazon Linux 2, ECS optimizado para Amazon, AMI es el predeterminado AMI para los recursos de cómputo en los entornos de cómputo gestionados. El Amazon Linux 2, ECS optimizado para Amazon, AMI está preconfigurado y probado AWS Batch por AWS ingenieros. Es un mínimo con el AMI que puede empezar y hacer que sus recursos informáticos funcionen AWS rápidamente. Para obtener más información, consulte Amazon ECS Optimized AMI 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 Instalación del agente ECS contenedor de Amazon en una EC2 instancia de Amazon Linux 2 en la Guía para desarrolladores de Amazon Elastic Container Service:

```
$ sudo amazon-linux-extras disable docker
$ sudo amazon-linux-extras install ecs-init
```

Por ejemplo, si quiere ejecutar GPU cargas de trabajo en sus recursos AWS Batch informáticos, puede empezar con Amazon Linux Deep Learning AMI. A continuación, configure las AWS Batch tareas AMI para que se ejecuten. Para obtener más información, consulte Usa una GPU carga de trabajo AMI.

> **Important**
>
> Puede elegir una base AMI que no sea compatible con el `ecs-init` paquete. Sin embargo, si lo hace, debe configurar una forma de iniciar el ECS agente de Amazon durante el arranque y mantenerlo en funcionamiento. También puede ver varios ejemplos de scripts de configuración de datos de usuario que se utilizan `systemd` para iniciar y supervisar el agente ECS contenedor de Amazon. 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. Lanza una instancia desde la base que hayas seleccionado AMI con las opciones de almacenamiento adecuadas para tiAMI. Puede configurar el tamaño y la cantidad de volúmenes de Amazon adjuntos o EBS volúmenes de almacenamiento de instancias si el tipo de instancia que seleccionó los admite. Para obtener más información, consulte Lanzamiento de una instancia y Amazon EC2 Instance Store en la Guía del EC2 usuario de Amazon.
3. Conéctese a su instancia con SSH y realice todas las tareas de configuración necesarias. Esto puede incluir una de las siguientes etapas, o todas:
   - Instalación del agente de ECS contenedores de Amazon. Para obtener más información, consulte Instalación del Amazon ECS Container Agent en la Guía para desarrolladores de Amazon Elastic Container Service.
   - Configurar un script para formatear volúmenes de almacén de instancias.
   - Añadir el volumen del almacén de instancias o los sistemas de EFS archivos de Amazon al `/etc/fstab` archivo para montarlos al arrancar.
   - Configurar opciones de Docker, como habilitar la depuración o ajustar el tamaño de la imagen de base.

- Instalar paquetes o copiar archivos.

Para obtener más información, consulte Conectarse a su instancia de Linux mediante SSH la Guía del EC2 usuario de Amazon.

4. Si has iniciado el agente de ECS contenedores de Amazon en tu instancia, debes detenerlo y eliminar cualquier archivo de punto de control de datos persistente antes de crear el tuyoAMI. De lo contrario, si no lo haces, el agente no se iniciará en las instancias que se lancen desde la tuyaAMI.

   a. Detenga al agente de ECS contenedores de Amazon.

      - Amazon Linux 2 ECS AMI optimizado para Amazon:

        ```
        sudo systemctl stop ecs
        ```

      - Amazon Linux ECS AMI optimizado para Amazon:

        ```
        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 a AMI partir de su instancia en ejecución. Para obtener más información, consulta Cómo crear un Linux EBS respaldado por Amazon AMI en la guía EC2 del usuario de Amazon.

Para usar tu nuevo AMI con AWS Batch

1. Una vez creado AMI el nuevo, cree un entorno informático con el nuevoAMI. Para ello, elija el tipo de imagen e introduzca el AMI ID personalizado en el cuadro de anulación del ID de imagen al crear el entorno AWS Batch informático. Para obtener más información, consulte the section called “Tutorial: Creación de un entorno informático gestionado con EC2 los recursos de Amazon”.

> **Note**
>
> El AMI que elijas para un entorno de cómputo debe coincidir con la arquitectura de los tipos de instancias que quieras usar para ese entorno de cómputo. Por ejemplo, si su entorno de cómputo usa A1 tipos de instancias, el recurso de cómputo AMI que elijas debe ser compatible Arm instancias. Amazon ECS vende ambos x86 y Arm versiones de Amazon Linux 2 ECS optimizadas para AmazonAMI. Para obtener más información, consulte Amazon Linux 2 ECS optimizado para Amazon AMI en la Guía para desarrolladores de Amazon Elastic Container Service.

2. Cree una cola de trabajos y asocie el entorno de computación nuevo. Para obtener más información, consulte Crear 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 combinación de tipos de arquitectura de entornos de cómputo en una sola cola de trabajos.

3. (Opcional) Envíe un trabajo de muestra a la cola de trabajos nueva. Para obtener más información, consulte Referencia: ejemplos de definiciones de trabajo, Crear una definición de trabajo de un solo nodo y Tutorial: enviar un trabajo.

Usa una GPU carga de trabajo AMI

Para ejecutar GPU cargas de trabajo en sus recursos AWS Batch informáticos, debe utilizar un dispositivo AMI con GPU soporte. Para obtener más información, consulte Working with GPUs on Amazon ECS y Amazon ECS -optimized AMIs in Amazon Elastic Container Service Developer Guide.

En los entornos de cómputo gestionado `p2` `p3` `p4` `p5` `g3``g3s`, `g4` si el entorno de cómputo especifica algún tipo de `g5` instancia o familia de instancias, AWS Batch utiliza un Amazon ECS GPU optimizadoAMI.

En entornos de cómputo no gestionados, AMI se recomienda un Amazon ECS GPU optimizado. Puede utilizar el AWS Command Line Interface almacén GetParameterde AWS Systems Manager parámetros y GetParametersByPathlas operaciones para recuperar los metadatos de la versión ECS GPU optimizada AMIs para Amazon recomendada. GetParameters

> **Note**
>
> La familia de p5 instancias solo es compatible con versiones iguales o posteriores a `20230912` la ECS GPU optimizada para Amazon AMI y son incompatibles con los tipos de g2 instancia p2 y. Si necesita usar p5 instancias, asegúrese de que su entorno de cómputo no p2 contenga g2 instancias y use el último Batch predeterminadoAMI. Al crear un nuevo entorno de cómputo se utilizará la versión más reciente, AMI pero si vas a actualizar tu entorno de cómputo para incluirlap5, puedes asegurarte de que estás usando la última versión AMI `updateToLatestImageVersion`configurándola `true` en `ComputeResource` las propiedades. Para obtener más información sobre la AMI compatibilidad con GPU las instancias, consulte Trabajar con Amazon GPUs ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

Los siguientes ejemplos muestran cómo utilizar el GetParametercomando.

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 AMI información del `Value` parámetro.

```
{
    "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'])
```

La salida solo incluye el AMI ID y el AMI nombre:

```
image_id   = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Los siguientes ejemplos muestran 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 AMI identificador y el AMI nombre, utilizando la ruta completa para 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 GetParametersByPathcomando.

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 Recuperación de ECS AMI metadatos optimizados para Amazon en la Guía para desarrolladores de Amazon Elastic Container Service.

Obsolescencia de Amazon Linux

El Amazon Linux AMI (también llamado Amazon Linux 1) llegó al final de su vida útil el 31 de diciembre de 2023. AWS Batch ha dejado de dar soporte a Amazon Linux, AMI 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 AL FAQ.

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.

AMIEs posible que sus entornos informáticos que utilizan 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 Amazon Linux AMI 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 Amazon Linux AMI a Amazon Linux 2023 o Amazon Linux 2, consulte Actualización de entornos de cómputo - AWS Batch.

Usa la plantilla de EC2 lanzamiento de Amazon con AWS Batch

AWS Batch admite el uso de plantillas de EC2 lanzamiento de Amazon con sus entornos EC2 informáticos. Con las plantillas de lanzamiento, puede modificar la configuración predeterminada de sus recursos AWS Batch informáticos sin necesidad de crear plantillas personalizadasAMIs.

> **Note**
>
> Los recursos de AWS Fargate no admiten plantillas de lanzamiento.

Es necesario crear una plantilla de lanzamiento para poder asociarla a un entorno de computación. Puedes crear una plantilla de lanzamiento en la EC2 consola de Amazon. O bien, puedes usar el AWS CLI o un AWS SDK. Por ejemplo, el siguiente JSON archivo representa una plantilla de lanzamiento que cambia el tamaño del volumen de datos de Docker para el recurso AWS Batch informático predeterminado AMI y también lo configura para que esté cifrado.

```
{
    "LaunchTemplateName": "increase-container-volume-encrypt",
    "LaunchTemplateData": {
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "Encrypted": true,
                    "VolumeSize": 100,
                    "VolumeType": "gp2"
                }
            }
        ]
    }
}
```

Puedes crear la plantilla de lanzamiento anterior guardándola JSON en un archivo llamado `lt-data.json` y ejecutando el siguiente AWS CLI comando.

```
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 EC2 usuario de Amazon.

Si utiliza una plantilla de lanzamiento para crear el entorno de computación, puede mover los siguientes parámetros del entorno de computación existente a la plantilla de lanzamiento:

> **Note**
>
> Supongamos que alguno de estos parámetros (excepto las EC2 etiquetas de Amazon) 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 EC2 etiquetas de Amazon se combinan entre la plantilla de lanzamiento y la configuración del entorno informático. Si se produce una colisión en la clave de la etiqueta, el valor de la configuración del entorno de computación tiene prioridad.

- Amazon EC2 key pair
- Amazon EC2 AMI ID
- Grupo de seguridad IDs
- EC2Etiquetas de Amazon

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 de computación)
- Rol de instancia (especifique el rol de instancia que desea utilizar al crear el entorno de computación)
- Subredes de interfaz de red (especifique las subredes que desea utilizar al crear el entorno de computación)
- Opciones del mercado de instancias (AWS Batch debe controlar la configuración de las instancias puntuales)
- Inhabilite API la terminación (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 de computación.

Datos EC2 de usuario de Amazon en plantillas de lanzamiento

Puedes proporcionar datos de EC2 usuario de Amazon en tu plantilla de lanzamiento que ejecuta cloud-init cuando se lanzan tus 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 EC2 de usuario de Amazon en las plantillas de lanzamiento deben estar en formato de archivo de MIME varias partes. Esto se debe a que sus datos de usuario se combinan con otros datos de AWS Batch usuario necesarios para configurar sus recursos informáticos. Puede combinar varios bloques de datos de usuario en un único archivo de MIME varias partes. Por ejemplo, es posible que desee combinar un boothook en la nube que configura el daemon de Docker con un script de shell de datos de usuario que escriba la información de configuración para el agente contenedor de Amazon. ECS

Si lo estás usando AWS CloudFormation, el tipo AWS:CloudFormation: :Init se puede usar con el script auxiliar cfn-init para realizar escenarios de configuración comunes.

Un archivo de MIME varias partes consta de los siguientes componentes:

- El tipo de contenido y declaración de límite de partes: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- La declaración de MIME versión: `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 de MIME varias partes: `--==BOUNDARY==--`. Debe dejar en blanco la línea anterior al límite de cierre.

> **Note**
>
> Si añades datos de usuario a una plantilla de lanzamiento en la EC2 consola de Amazon, puedes pegarlos como texto sin formato. O bien, puede cargarlos desde un archivo.
> Si utilizas AWS CLI o an AWS SDK, primero debes `base64` codificar los datos del usuario y enviar esa cadena como el valor del `UserData` parámetro cuando llames CreateLaunchTemplate, tal y como se muestra en este JSON archivo.
>
> ```
> {
>  "LaunchTemplateName": "base64-user-data",
>  "LaunchTemplateData": {
>  "UserData":
> "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWFzZS1jb250YWluZXItdm9sdW..."
>  }
> }
> ```

Temas

- Referencia: ejemplos de plantillas de EC2 lanzamiento de Amazon

Referencia: ejemplos de plantillas de EC2 lanzamiento de Amazon

A continuación, se muestran ejemplos de archivos de MIME varias partes que puede utilizar para crear sus propias plantillas.

Ejemplos

- Ejemplo: montar un sistema de EFS archivos de Amazon existente
- Ejemplo: anular la configuración predeterminada del agente ECS contenedor de Amazon
- Ejemplo: montar un sistema de archivos Amazon FSx for Lustre existente

Ejemplo: montar un sistema de EFS archivos de Amazon existente

Example

Este ejemplo de archivo de MIME varias partes configura el recurso informático para instalar el `amazon-efs-utils` paquete y montar un sistema de EFS archivos de Amazon 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: anular la configuración predeterminada del agente ECS contenedor de Amazon

Example

Este ejemplo de archivo de MIME varias partes anula la configuración predeterminada de limpieza de imágenes de Docker para 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 Amazon FSx for Lustre existente

Example

Este ejemplo de archivo de MIME varias partes configura el recurso de cómputo para instalar el `lustre2.10` paquete de la biblioteca Extras y montar un sistema de archivos existente FSx para Lustre en `/scratch` y con el nombre de montaje de. `fsx` Este ejemplo es para Amazon Linux 2. Para obtener instrucciones de instalación para otras distribuciones de Linux, consulte Instalación del cliente de Lustre en la Guía del usuario de Amazon FSx for Lustre. Para obtener más información, consulta Cómo montar automáticamente el sistema de FSx archivos de Amazon en la Guía del usuario de Amazon FSx for 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 volúmenes y mountPointslos miembros de las propiedades del contenedor, los puntos de montaje deben estar mapeados en el contenedor.

```
{
    "volumes": [
        {
```

```
            "host": {
                "sourcePath": "/scratch"
            },
            "name": "Scratch"
        }
    ],
    "mountPoints": [
        {
            "containerPath": "/scratch",
            "sourceVolume": "Scratch"
        }
    ],
}
```

EC2configuraciones

AWS Batch utiliza Amazon ECS Optimized AMIs for EC2 and EC2 Spot Compute entornos. 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 GPU de AWS Graviton o no.

> **Note**
>
> AWS Batch también es compatible con Amazon Linux 2023.

El Amazon Linux AMI (también llamado Amazon Linux 1) llegó al final de su vida útil el 31 de diciembre de 2023. AWS Batch ha dejado de dar soporte a Amazon Linux, AMI ya que no recibirá actualizaciones de seguridad ni correcciones de errores a partir del 1 de enero de 2024. Para obtener más información acerca de Amazon Linux end-of-life, consulte AL FAQ.

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.

AMIEs posible que sus entornos informáticos que utilizan Amazon Linux sigan funcionando después de la end-of-life fecha del 31 de diciembre de 2023. Sin embargo, estos entornos de cómputo ya no recibirán nuevas actualizaciones de software, parches de seguridad ni correcciones de errores AWS. Es su responsabilidad mantener estos entornos de cómputo en Amazon Linux AMI 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 Amazon Linux AMI a Amazon Linux 2023 o Amazon Linux 2, consulte Actualización de entornos de cómputo - AWS Batch

Estrategias de asignación de tipos de instancia para AWS Batch

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 está alcanzando las cuotas de EC2 servicio de Amazon, los trabajos adicionales no se ejecutarán 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 utiliza Spot Fleets con`BEST_FIT`, debe especificar el IAM rol de Spot Fleet. `BEST_FIT`no se admite cuando se actualizan los entornos de cómputo. Para obtener más información, consulte Actualizar entornos de computación.

> **Note**
>
> AWS Batch administra AWS los recursos de tu cuenta. Los entornos informáticos con la estrategia de FIT asignación BEST _ 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 BEST _ de FIT cómputo recién creados utilizarán plantillas de lanzamiento 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 instancia con un coste menor para cada unidad 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 sean 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.

Administración de memoria de recursos informáticos

Cuando el agente ECS contenedor de Amazon registra un recurso informático en un entorno informático, debe determinar la cantidad de memoria que el recurso informático tiene disponible para reservar 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 EC2 las instancias de Amazon. 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 de computación.

Suponga que especifica 8192 MiB para el trabajo y que ninguno de sus recursos de computación tiene 8192 MiB o más de memoria disponible para cumplir este requisito. Entonces, el trabajo no se puede colocar en su entorno de computación. Si utilizas un entorno informático gestionado, AWS Batch debes lanzar un tipo de instancia más grande para satisfacer la solicitud.

El recurso AWS Batch informático predeterminado AMI también reserva 32 MiB de memoria para el agente ECS contenedor de Amazon y otros procesos críticos del sistema. Esta memoria no está disponible para la asignación de trabajos. Para obtener más información, consulte Reserve memoria del sistema.

El agente ECS contenedor de Amazon usa la `ReadMemInfo()` función Docker para consultar la memoria total disponible en el 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 del resultado de una `m4.large` instancia que ejecuta Amazon Linux, ECS optimizado para Amazon. AMI

```
             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.

Temas

- Reserve memoria del sistema
- Tutorial: Vea la memoria de los recursos de cómputo
- CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS

Reserve memoria del sistema

Si ocupa toda la memoria de un recurso de computación con sus trabajos, es posible que estos compitan por la memoria con procesos críticos del sistema y provoquen un fallo del sistema. El agente ECS contenedor de Amazon 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.

El recurso AWS Batch informático predeterminado AMI reserva 32 MiB de memoria para el agente ECS contenedor de Amazon y otros procesos críticos del sistema.

Tutorial: Vea la memoria de los recursos de cómputo

Puedes ver la cantidad de memoria con la que se registra un recurso informático en la ECS consola de Amazon o en la DescribeContainerInstancesAPIoperación. Si intenta maximizar la utilización de sus recursos proporcionando a sus trabajos tanta memoria como sea posible para un tipo de instancia en particular, puede observar la memoria disponible para ese recurso de computación y luego asignar a sus trabajos esa cantidad de memoria.

Para ver la memoria de recurso de computación

1. Abra la consola en la https://console.aws.amazon.com/ecs/versión 2.
2. Elija Clústeres y, a continuación, elija el clúster que aloja los recursos de computación que desee ver.

   El nombre del clúster del entorno de computación comienza por el nombre del entorno de computación.
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 de computación.

   El valor de memoria registrada es lo que el recurso informático registró en 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.

CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS

AWS Batch En AmazonEKS, puedes especificar los recursos que están disponibles para un contenedor. Por ejemplo, puede especificar `requests` `limits` valores para v CPU y los recursos de memoria.

Las siguientes son restricciones para especificar CPU los recursos v:

- Debe especificarse al menos una v CPU `requests` o un `limits` valor.
- Una CPU unidad v equivale a un núcleo físico o virtual.
- El CPU valor v debe introducirse en números enteros o en incrementos de 0,25.
- El CPU valor v válido más pequeño es 0,25.
- Si se especifican ambos, el valor `requests` debe ser menor o igual que el valor `limits`. De esta forma, puede configurar CPU configuraciones v blandas y rígidas.
- CPULos valores v no se pueden especificar en CPU forma milimétrica. Por ejemplo, `100m` no es un valor válido.
- AWS Batch usa el `requests` valor para escalar las decisiones. 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 `requests` valor para las decisiones de escalado. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Las siguientes son restricciones para especificar GPU los recursos:

- Si se especifican ambos, el valor `requests` debe ser igual que el valor `limits`.
- AWS Batch utiliza el `requests` valor para escalar las decisiones. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Ejemplo: definiciones de puestos

La siguiente definición AWS Batch de EKS trabajo de Amazon configura soft v CPU shares. Esto permite AWS Batch a Amazon EKS utilizar toda la CPU capacidad v para el tipo de instancia. Sin embargo, si hay otros trabajos en ejecución, al trabajo se le asigna un máximo de 2vCPUs. 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 definición AWS Batch de EKS trabajo de Amazon tiene un `request` valor de 1 y asigna un máximo de 4 vCPUs 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 definición AWS Batch de EKS trabajo de Amazon establece un CPU `limits` valor v 1 y un `limits` valor de memoria 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 EKS trabajo AWS Batch de Amazon a un EKS pod de Amazon, AWS Batch copia el `limits` valor al `requests` valor. 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 nodos CPU y memoria

AWS Batch se basa en la lógica predeterminada del `bootstrap.sh` archivo para v CPU y las reservas de memoria. Para obtener más información sobre el archivo `bootstrap.sh`, consulte bootstrap.sh. Al ajustar el tamaño de v CPU y de los recursos de memoria, tenga en cuenta los ejemplos siguientes.

> **Note**
>
> Si no se está ejecutando ninguna instancia, v CPU y las reservas de memoria pueden afectar inicialmente a la lógica de AWS Batch escalado y a la toma de decisiones. Una vez ejecutadas las instancias, AWS Batch ajusta las asignaciones iniciales.

Ejemplo: reserva de nodos CPU

El valor de la CPU reserva se calcula en milicores utilizando el número total de vCPUs milicores disponibles para la instancia.

número v CPU	Porcentaje reservado
1	6%

número v CPU	Porcentaje reservado
2	1%
3-4	0,5%
4 y superior	0,25%

Si se utilizan los valores anteriores, se aplica lo siguiente:

- El valor de CPU reserva para una `c5.large` instancia con 2 vCPUs es de 70 m. Se calcula de la siguiente manera: (1*60) + (1*10) = 70 m.
- El valor de CPU reserva para una `c5.24xlarge` instancia con 96 vCPUs es 310 m. Se calcula de la siguiente manera: (1*60) + (1*10) + (2*5) + (92*2.5) = 310 m.

En este ejemplo, hay 1930 CPU unidades V de milicore (calculadas entre 2000 y 70) disponibles para ejecutar trabajos en una `c5.large` instancia. Supongamos que su trabajo requiere CPU unidades v de 2 (2*1000 m), el trabajo no cabe en una sola instancia. `c5.large` Sin embargo, un trabajo que requiere `1.75` v CPU unidades es adecuado.

Ejemplo: reserva de memoria de nodos

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 la cantidad máxima de pods que admite un tipo de instancia, consulta .txt eni-max-pods
- 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`*)) - 25 - (*`HardEvictionLimit value.`*)).

Una instancia `c5.large` admite hasta 29 pods. Para una `c5.large` instancia de 8 GB con un `HardEvictionLimit` valor 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 8.192 MiB el trabajo no cabe en este caso a pesar de que es un 8 gibibyte (GiB) instancia.

DaemonSets

Cuando usas DaemonSets, tenga en cuenta lo siguiente:

- Si no hay ninguna EKS instancia de Amazon AWS Batch en ejecución, DaemonSets puede afectar inicialmente a la lógica de AWS Batch escalamiento y a la toma de decisiones. AWS Batch inicialmente asigna 0,5 CPU unidades v y 500 MiB para las esperadas DaemonSets. Una vez ejecutadas las instancias, AWS Batch ajusta las asignaciones iniciales.
- Si una DaemonSet define v CPU o límites de memoria, AWS Batch en Amazon los EKS trabajos tienen menos recursos. Le recomendamos que mantenga el número de DaemonSets que están asignados a los AWS Batch trabajos lo más bajo posible.

Actualizar entornos de computación

Después de crear un entorno de procesamiento que utilice EC2 recursos, puede actualizar muchos de los ajustes del entorno de procesamiento directamente. Sin embargo, para cambiar algunos de los ajustes es necesario que AWS Batch reemplace las instancias del entorno de computación.

Actualización de los entornos AWS Fargate informáticos

Para los entornos de computación 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 de computación. La segunda es una actualización de infraestructura en la que se sustituyen las instancias del entorno de computación. 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 estos ajustes, se produce una actualización de escalado: deseado vCPUs vCPUs (`desiredvCpusmaxvCpus`), máximo (), mínimo vCPUs (`minvCpus`), rol de servicio (`serviceRole`) y 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 se cambia alguna de las siguientes configuraciones en un UpdateComputeEnvironmentAPIacción, AWS Batch inicia una actualización de la infraestructura. Una actualización de la infraestructura requiere que la función de servicio esté establecida en AWSServiceRoleForBatch(la opción predeterminada) y que la estrategia de asignación sea `BEST_FIT_PROGRESSIVESPOT_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 de computación 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 UpdateComputeEnvironmenty UpdatePolicyen la AWS Batch APIReferencia.

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 de computación, 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 el número deseado vCPUs está vCPUs por debajo del número máximo de al menos tanto como lo requiere el tipo de instancia más pequeño. vCPUs

Actualizaciones de la infraestructura

Es necesaria una actualización de la infraestructura para cambiar algunos ajustes de un entorno de computación. Si se cambia alguna de las siguientes configuraciones, se inicia una actualización de la infraestructura:

> **Important**
>
> El entorno informático debe utilizar la función AWSServiceRoleForBatchvinculada al servicio para realizar cambios que requieran una actualización de la infraestructura.
> Si el entorno informático usa un rol vinculado a un servicio, no se puede cambiar para usar un rol normal. IAM Del mismo modo, si el entorno de procesamiento tiene un IAM rol normal,

no se puede cambiar para usar un rol vinculado a un servicio. Por lo tanto, solo puede realizar actualizaciones de infraestructura en entornos de computación 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`)
- EC2configuración () `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`)
- VPCsubredes () `subnets`
- EC2etiquetas () `tags`
- Tipo de entorno de computación (`type`, puede ser uno de `EC2` o `SPOT`)
- Si se debe actualizar a la versión más reciente AMI compatible AWS Batch durante una actualización de infraestructura `updateToLatestImageVersion`

Actualizar el AMI ID

Durante una actualización de la infraestructura, el AMI identificador del entorno informático puede cambiar en función de si AMIs se especifica en alguna de estas tres configuraciones. AMIsse especifican en la plantilla `imageId` (`incomputeResources`), `imageIdOverride`

(in`ec2Configuration`) o de lanzamiento especificada en`launchTemplate`. Supongamos que no AMI IDs se especifica ninguna en ninguna de esas configuraciones y que la `updateToLatestImageVersion` configuración sí`true`. Luego, la última versión ECS optimizada de Amazon AMI compatible con AWS Batch se utiliza para cualquier actualización de infraestructura.

Si se especifica una AMI ID en al menos una de estas configuraciones, la actualización dependerá de la configuración proporcionada por la AMI ID utilizada antes de la actualización. Al crear un entorno informático, la prioridad a la hora de seleccionar un AMI ID es primero la plantilla de lanzamiento, después la `imageId` configuración y, por último, la `imageIdOverride` configuración. Sin embargo, si el AMI ID que se utiliza proviene de la plantilla de lanzamiento, al actualizar la `imageIdOverride` configuración `imageId` o la configuración, no se actualiza el AMI ID. La única forma de actualizar un AMI ID seleccionado de 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 AMI ID diferente de forma predeterminada o se selecciona la última versión de la plantilla de lanzamiento, ese AMI ID se utilizará en la actualización.

Si la plantilla de lanzamiento no se usó para seleccionar el AMI AMI ID, se usa el ID especificado en `imageIdOverride` los parámetros `imageId` o. Si se especifican ambos, se utiliza el AMI ID especificado en el `imageIdOverride` parámetro.

Supongamos que el entorno de cómputo usa un AMI ID especificado por los `launchTemplate` parámetros `imageIdimageIdOverride`, o y usted quiere usar la última versión ECS optimizada de Amazon AMI compatible con AWS Batch. A continuación, la actualización debe eliminar la configuración proporcionada AMIIDs. 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 AMI ID proviene de la plantilla de lanzamiento, puedes cambiarlo a la última versión ECS optimizada de Amazon AMI que sea AWS Batch 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 AMI ID.
- Si la versión actualizada de la plantilla de lanzamiento no especifica un AMI ID, el `updateToLatestImageVersion` parámetro debe configurarse en`true`.

Entornos de cómputo Fargate

Fargate es una tecnología que puedes utilizar AWS Batch para ejecutar contenedores sin tener que gestionar servidores o clústeres de instancias de AmazonEC2. Con 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.

Cuando ejecuta sus trabajos con los recursos de Fargate, empaqueta la aplicación en contenedores, especifica los CPU requisitos de memoria, define las redes y IAM las políticas y lanza la aplicación. Cada trabajo de Fargate tiene su propio límite de aislamiento y no comparte el núcleo, los CPU recursos, los recursos de memoria o la interface de red elástica subyacentes con otro trabajo.

Temas

- Cuándo usar Fargate
- Definiciones de trabajo en Fargate
- Colas de trabajo en Fargate
- Entornos de computación en Fargate

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. Normalmente, se tarda unos minutos en activar una nueva EC2 instancia de Amazon. 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.

Sin embargo, te recomendamos que utilices Amazon EC2 si tus trabajos requieren alguno de los siguientes requisitos:

- Más de 16 vCPUs
- Más de 120 gibibytes (GiB) de memoria
- Un GPU

- Una imagen de Amazon Machine personalizada (AMI)
- ¿Alguno de los linuxParametersparámetros

Si tienes un gran número de trabajos, te recomendamos que utilices la EC2 infraestructura de Amazon. Por ejemplo, si el número de trabajos que se ejecutan simultáneamente supera la limitación de regulación de Fargate. Esto se debe a que, conEC2, los trabajos se pueden enviar a un ritmo mayor a EC2 los recursos que a los recursos de Fargate. Además, se pueden ejecutar más trabajos simultáneamente cuando se usa. EC2 Para obtener más información, consulte Fargate Service Quotas en la Guía para desarrolladores de Amazon Elastic Container Service.

Definiciones de trabajo en Fargate

AWS Batch Los trabajos AWS Fargate activados no admiten todos los parámetros de definición de trabajos 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 IAMFunciones 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`

Tanto CPU los requisitos de memoria como los de v deben especificarse utilizando los valores admitidos. GPUlos recursos no se admiten para los trabajos que se ejecutan con los recursos de Fargate.

Si utiliza GuardDuty Runtime Monitoring, existe 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 CPUlos límites de 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 ECSAmazon.

```
"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 Utilice el controlador de registro awslogs.

Miembros en `networkConfiguration`

`assignPublicIp`

Si la subred privada no tiene una NAT puerta de enlace conectada para enviar tráfico a Internet, `assignPublicIp` debe ser "». `ENABLED` Para obtener más información, consulte Rol de ejecución de AWS Batch IAM.

Colas de trabajo en Fargate

AWS Batch las colas de trabajos activas prácticamente AWS Fargate no han cambiado. La única restricción es que todos los entornos de computación que aparecen en la lista en `computeEnvironmentOrder` deben ser entornos de computación de Fargate (`FARGATE` o `FARGATE_SPOT`). EC2y los entornos de cómputo de Fargate no se pueden mezclar.

Entornos de computación en Fargate

AWS Batch los entornos de cómputo activados AWS Fargate no admiten todos los parámetros del entorno de cómputo disponibles. Algunos parámetros no son compatibles. Otros tienen requisitos específicos para Fargate.

La siguiente lista describe los parámetros de entorno de computación 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 de computación de Fargate y no se pueden proporcionar.

`subnets`

Si las subredes enumeradas en este parámetro no tienen NAT pasarelas conectadas, el `assignPublicIp` parámetro de la definición del trabajo debe estar establecido en. `ENABLED`

`tags`

Esto no se aplica a los entornos de computación de Fargate y no se puede proporcionar. Para especificar etiquetas para los entornos de computación de Fargate, use el parámetro `tags` que no está en el objeto `computeResources`.

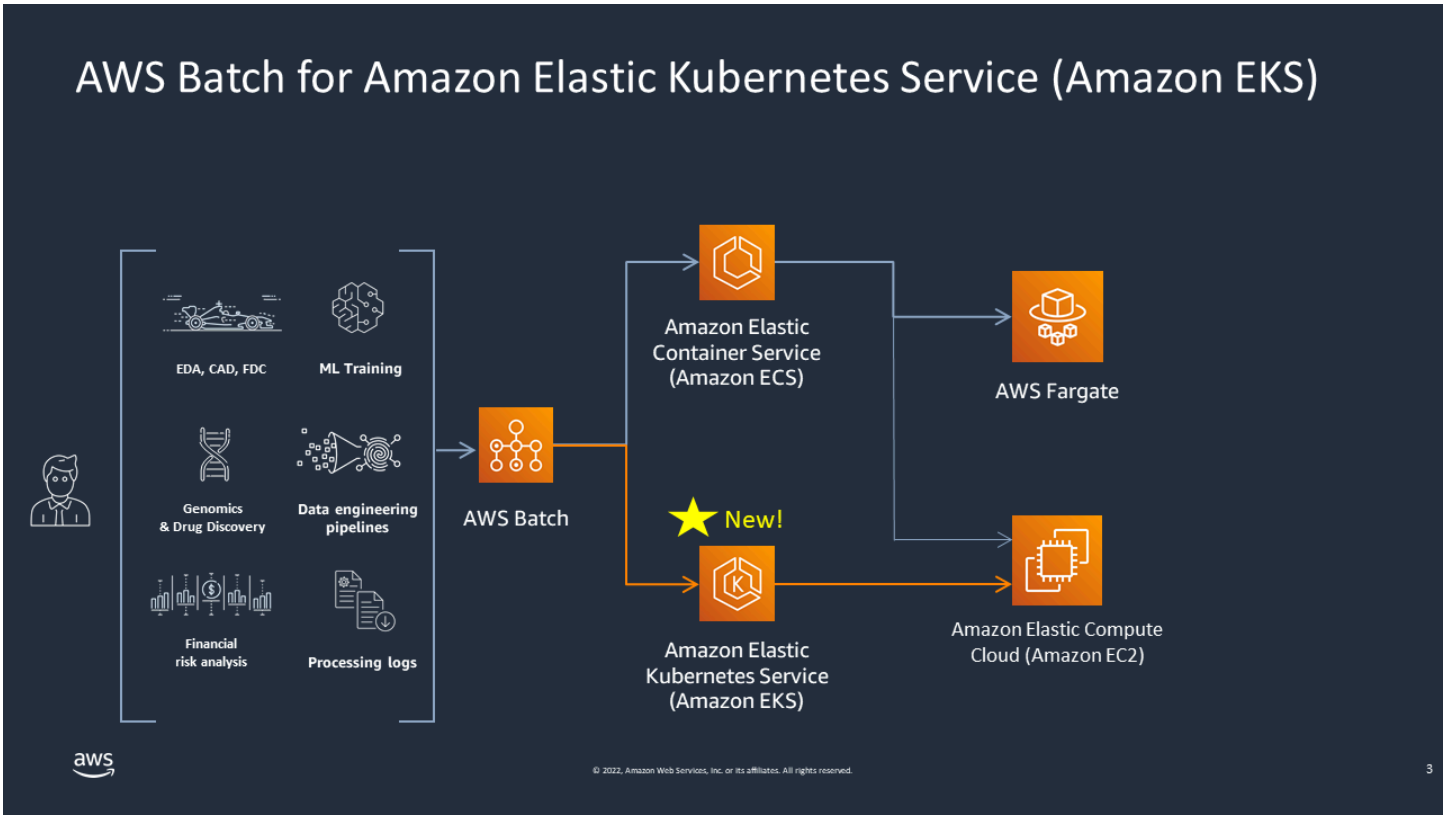
`type`

Debe ser `FARGATE` o `FARGATE_SPOT`.

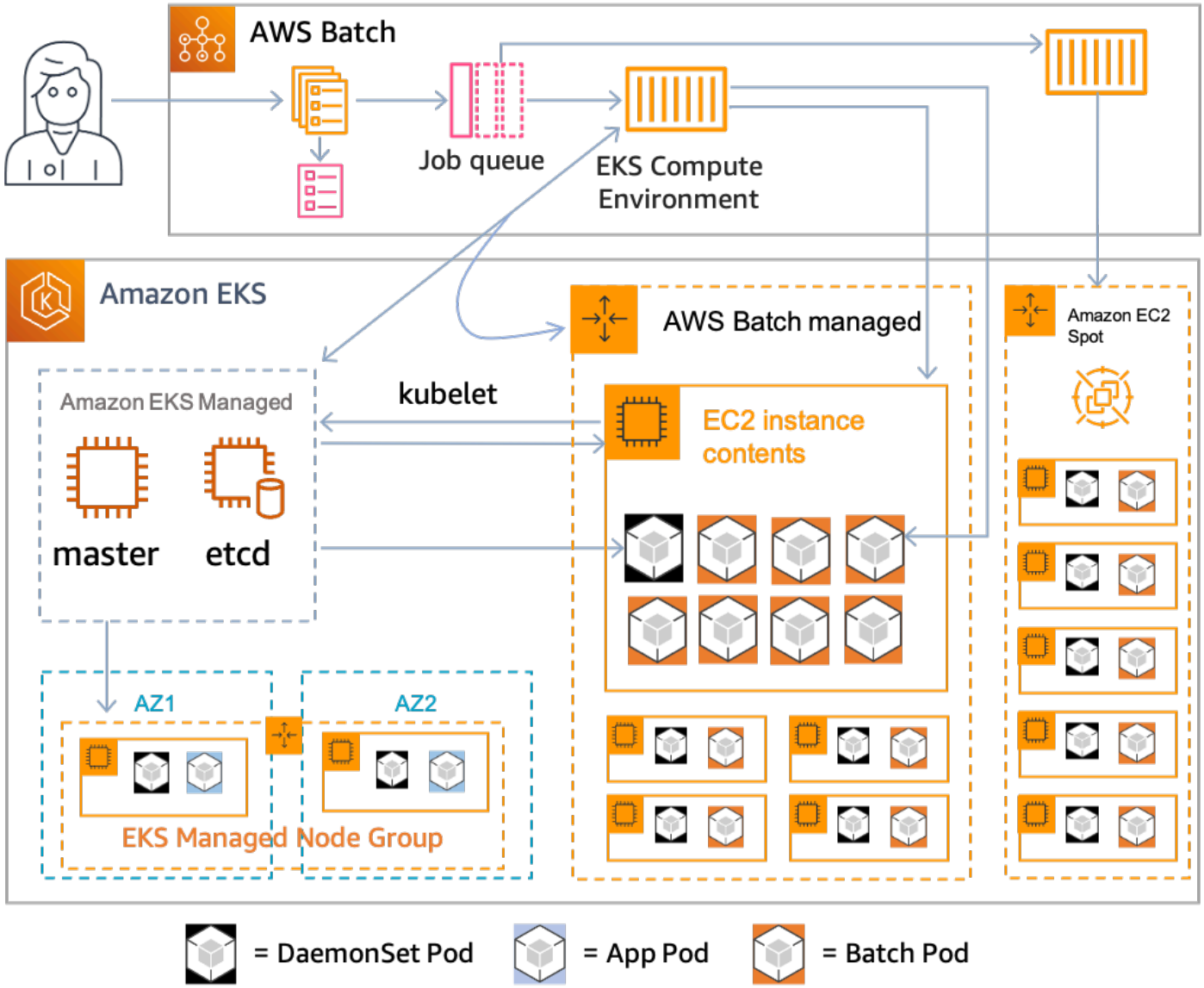
```
"type": "FARGATE_SPOT"
```

Entornos de EKS cómputo de Amazon

Tutorial: Cómo empezar AWS Batch en Amazon EKSproporciona una breve guía para crear entornos de EKS cómputo. En esta sección se proporcionan más detalles sobre los entornos de EKS cómputo de Amazon.



AWS Batch simplifica las cargas de trabajo por lotes en los EKS clústeres de Amazon 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 EC2 instancias de Amazon. AWS Batch integra varias de las mejores prácticas 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 su Kubernetes 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 es un servicio gestionado, no hay Kubernetes componentes (por ejemplo, operadores o recursos personalizados) para instalar o administrar 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 Kubernetes APIservidor. AWS Batch llamadas Kubernetes APIscrear, supervisar y eliminar Kubernetes pods y nodos.

AWS Batch tiene una lógica de escalado integrada para escalar Kubernetes nodos basados en 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, AWS Batch reduce los nodos hasta alcanzar

la capacidad mínima que hayas 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, otros Kubernetes las cargas de trabajo no se colocan en los nodos gestionados por AWS Batch. La excepción son las `DaemonSets` que pueden dirigirse a AWS Batch los nodos para proporcionar la 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 a AWS Batch, interactúas directamente con el AWS Batch API. AWS Batch traduce los trabajos `podspecs` y, a continuación, crea las solicitudes para colocar pods en los nodos gestionados por AWS Batch tu EKS clúster de Amazon. Puede utilizar herramientas como, por ejemplo, `kubectl` para ver los nodos y los pods en ejecución. Cuando un pod haya completado su ejecución, AWS Batch elimina el pod que creó para mantener una carga menor en el Kubernetes sistema.

Puedes empezar conectando un EKS clúster de Amazon válido con AWS Batch. A continuación, adjunte una cola de AWS Batch trabajos y registre una definición de EKS trabajo de Amazon con atributos `podspec` equivalentes. Por último, envíe los trabajos mediante la SubmitJobAPIoperación que hace referencia a la definición del trabajo. Para obtener más información, consulte Tutorial: Cómo empezar AWS Batch en Amazon EKS.

Amazon EKS

Temas

- AWS Batch predeterminado AMI
- Compatible Kubernetes control de versiones
- Actualice la Kubernetes versión del entorno informático
- Responsabilidad compartida del Kubernetes nodos
- Ejecuta un DaemonSet en nodos AWS Batch gestionados
- Personalice las plantillas de EKS lanzamiento de Amazon

AWS Batch predeterminado AMI

Al crear un entorno EKS informático de Amazon, no es necesario especificar una Amazon Machine Image (AMI). AWS Batch selecciona un Amazon EKS optimizado AMI en función del Kubernetes

tipos de versión e instancia que se especifican en su CreateComputeEnvironmentsolicitud. En general, le recomendamos que utilice la AMI selección predeterminada. Para obtener más información sobre Amazon EKS optimizadoAMIs, consulte Amazon Linux EKS optimizado AMIs en la Guía del EKS usuario de Amazon.

Ejecute el siguiente comando para ver qué AMI tipo AWS Batch está seleccionado para su entorno EKS informático de Amazon. El siguiente ejemplo es un tipo que no es de GPU instancia.

```
# 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 GPU instancia.

```
# 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"
      }
    ]
```

Compatible Kubernetes control de versiones

AWS Batch en Amazon EKS actualmente admite lo siguiente Kubernetes versiones:

- 1.30
- 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 `CreateComputeEnvironment` API operación o la `UpdateComputeEnvironment` API operación para crear o actualizar un entorno informático. Este problema se produce si especificas un elemento no compatible Kubernetes versión en`EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Para resolver este problema, elimine el entorno informático y, a continuación, vuelva a crearlo con un entorno compatible Kubernetes versión.

Puedes realizar una actualización de versión menor en tu EKS clúster de Amazon. 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
```

Actualice la Kubernetes versión del entorno informático

Con AWS Batch, puede actualizar el Kubernetes versión de un entorno informático que admite las actualizaciones de los EKS clústeres de Amazon. La Kubernetes la versión de un entorno de cómputo es la EKS AMI versión de Amazon para Kubernetes nodos que se AWS Batch lanzan para ejecutar tareas. Puede realizar un Kubernetes actualice la versión en sus EKS nodos de Amazon antes o después de actualizar la versión del plano de control del EKS clúster de Amazon. Le recomendamos que actualice los nodos después de actualizar el plano de control. Para obtener más información, consulta Actualización de un EKS clúster de Amazon Kubernetes versión de la Guía del EKS usuario de Amazon.

Para actualizar la Kubernetes versión de un entorno informático, utilice la UpdateComputeEnvironmentAPIoperación.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

Responsabilidad compartida del Kubernetes nodos

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 gestionados en la Guía del EKS usuario de Amazon.
- Puede segmentar un DaemonSet para ejecutarse en nodos AWS Batch gestionados. Para obtener más información, consulte Ejecuta un DaemonSet en nodos AWS Batch gestionados.

AWS Batch no actualiza automáticamente el entorno informáticoAMIs. Es su responsabilidad actualizarlas. Ejecute el siguiente comando para actualizar su AMI versión AMIs a la última.

```
$ 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 Kubernetes versión del entorno de su ordenador para *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 versión más reciente AMI o a la Kubernetes versión, puede especificar si desea finalizar los trabajos cuando se actualicen (`terminateJobsOnUpdate`) y cuánto

tiempo esperar antes de que se sustituya una instancia si los trabajos en ejecución no finalizan (`jobExecutionTimeoutMinutes`.) Para obtener más información, consulte Actualizar entornos de computación y la política de actualización de la infraestructura (UpdatePolicy) establecida en el UpdateComputeEnvironmentAPIoperación.

Ejecuta un DaemonSet en nodos AWS Batch gestionados

AWS Batch pone manchas en AWS Batch los gestionados Kubernetes nodos. Puedes apuntar a DaemonSet para ejecutarse 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"
```

Personalice las plantillas de EKS lanzamiento de Amazon

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 corre`/etc/eks/bootstrap.sh`. No lo ejecutes `/etc/eks/bootstrap.sh` en tu plantilla de lanzamiento o cloud-init user-data guiones. 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, UpdateComputeEnvironmentse debe llamar para evaluar la versión de la plantilla de lanzamiento que se va a sustituir.

Temas

- Agrega argumentos kubelet adicionales
- Configure el tiempo de ejecución del contenedor
- Monta un EFS volumen de Amazon
- IPv6soporte

Agrega argumentos **`kubelet`** adicionales

AWS Batch admite la adición de argumentos adicionales al `kubelet` comando. Para ver la lista de parámetros compatibles, consulte kubelet en la Kubernetes documentación. 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==--
```

Configure el 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 Kubernetes documentación.

> **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 Kubernetes documentación.

```
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==--
```

Monta un EFS volumen de Amazon

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, consulta los ejemplos de configuración de la nube en la cloud-init documentación.

```
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 utilizar este volumen en el trabajo, debe añadirse en el eksPropertiesparámetro 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 EFS volumen de Amazon está montado en el `/mnt/efs` directorio. En el contenedor del EKS trabajo de Amazon, el volumen se monta en el `/efs` directorio.

IPv6soporte

AWS Batch admite EKS clústeres de Amazon que tienen IPv6 direcciones. No se requieren personalizaciones para obtener AWS Batch soporte. Sin embargo, antes de empezar, te recomendamos que revises las consideraciones y condiciones que se describen en la sección Asignación de IPv6 direcciones a pods y servicios de la Guía del EKS usuario de Amazon.

Colas de trabajo

Los trabajos se envían a una cola de trabajos, donde permanecen hasta que pueden programarse para ejecutarse en un entorno de computación. Una AWS cuenta puede tener varias colas de trabajos. Por ejemplo, puede crear una cola que utilice instancias de Amazon EC2 On-Demand para trabajos de alta prioridad y otra cola que utilice Amazon EC2 Spot Instances 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

- Crear una cola de trabajos
- Ver el estado de la cola de trabajos
- Políticas de programación de cuotas justas

Crear una cola de trabajos

Antes de poder enviar trabajos AWS Batch, debe crear una cola de trabajos. Al crear una cola de trabajos, se asocian uno o varios entornos de computación a la cola y se asigna un orden de preferencia.

También establece la prioridad de la cola de trabajos que determina el orden en que el programador de AWS Batch coloca los trabajos. Esto significa que, si un entorno de computación está asociado a más de una cola de trabajos, se da preferencia a la cola de trabajos con mayor prioridad.

Temas

- Tutorial: Crear una cola de EC2 trabajos en Amazon
- Tutorial: Crear una cola de trabajos en Fargate
- Tutorial: Crear una cola de EKS trabajos en Amazon
- Referencia: plantilla de cola de trabajos

Tutorial: Crear una cola de EC2 trabajos en Amazon

Complete los siguientes pasos para crear una cola de trabajos para Amazon Elastic Compute Cloud (AmazonEC2).

Para crear una cola de EC2 trabajos en Amazon

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 Colas de trabajos.
4. Seleccione Crear.
5. Para el tipo de orquestación, elija Amazon Elastic Compute Cloud (AmazonEC2).
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 de computación. 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) Para la política de programación Amazon Resource Name (ARN), elija una política de programación existente.
9. En Entornos de computación conectados, seleccione uno o varios entornos de computación de la lista para asociarlos a la cola de trabajos. Seleccione los entornos de computación 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 de computación para determinar qué entorno de computación inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos de computación deben estar en estado `VALID`. Puede asociar hasta tres entornos de computación con una cola de trabajos. Si no tiene un entorno de computación existente, elija Crear entorno de computación

   > **Note**
   >
   > Todos los entornos informáticos asociados a una cola de trabajos deben compartir el mismo modelo de aprovisionamiento. AWS Batch no admite la combinación de modelos de aprovisionamiento en una sola cola de trabajos.

10. En Orden de entorno de computación, 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.

Tutorial: Crear una cola de trabajos en Fargate

Complete los siguientes pasos para crear una cola de trabajos para. AWS Fargate

Para crear una cola de trabajos 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 Colas de trabajos.
4. Seleccione 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 de computación. 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) Para la política de programación Amazon Resource Name (ARN), elija una política de programación existente.
9. En Entornos de computación conectados, seleccione uno o varios entornos de computación de la lista para asociarlos a la cola de trabajos. Seleccione los entornos de computación 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 de computación para determinar qué entorno de computación inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos de computación deben estar en estado `VALID`. Puede asociar hasta tres entornos de computación 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 combinación de modelos de aprovisionamiento en una sola cola de trabajos.

10. En Orden de entorno de computación, 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.

Tutorial: Crear una cola de EKS trabajos en Amazon

Complete los siguientes pasos para crear una cola de trabajos para Amazon Elastic Kubernetes Service (Amazon). EKS

Para crear una cola de EKS trabajos en Amazon

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 Colas de trabajos.
4. Seleccione 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 de computación. 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) Para la política de programación Amazon Resource Name (ARN), elija una política de programación existente.
9. En Entornos de computación conectados, seleccione uno o varios entornos de computación de la lista para asociarlos a la cola de trabajos. Seleccione los entornos de computación 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 de computación para determinar qué entorno de computación inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos de computación deben estar en estado `VALID`. Puede asociar hasta tres entornos de computación 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 combinación 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 combinación de tipos de arquitectura de entornos de cómputo en una sola cola de trabajos.

10. En Orden de entorno de computación, 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.

Referencia: 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 CreateJobQueuela AWS Batch APIReferencia.

> **Note**
>
> Puede generar una plantilla de cola de trabajos con el siguiente AWS CLI comando.
>
> ```
> $ aws batch create-job-queue --generate-cli-skeleton
> ```

```
{
   "computeEnvironmentOrder": [
      {
         "computeEnvironment": "",
         "order": 0
```

```
      }
   ],
   "jobQueueName": "",
   "jobStateTimeLimitActions": [
      {
         "state": "RUNNABLE",
         "action": "CANCEL",
   "maxTimeSeconds": 0,
         "reason": ""

      }
   ],
   "priority": 0,
   "schedulingPolicyArn": "",
   "state": "ENABLED",
   "tags": {
      "KeyName": ""
   }
}
```

Ver el estado de la cola de trabajos

Tras crear una cola de trabajos y enviarlos, es importante poder supervisar su progreso. Puede utilizar la página de detalles del trabajo para revisar, gestionar y supervisar su cola de trabajos.

Vea la información de la cola de trabajos

En la AWS Batch consola, seleccione Colas de trabajos en el panel de navegación y elija la cola de trabajos que desee para ver sus detalles. En esta página, puede revisar y administrar su cola de trabajos y ver información adicional sobre las operaciones de la cola de trabajos, como la instantánea de la cola de trabajos, los límites de estado de los trabajos, el orden del entorno, las etiquetas y el código de la cola de trabajos. JSON

Detalles de la cola de trabajos

En esta sección se proporciona una descripción general y las opciones de mantenimiento de la cola de trabajos. Es importante tener en cuenta que puede encontrar el nombre del recurso de Amazon (ARN) en esta sección.

Para encontrar esta información a través de AWS Command Line Interface, utilice la `DescribeJobQueues`operación junto con el nombre de la cola de trabajos o el nombre correspondienteARN.

Instantánea de la cola de trabajos

Esta sección proporciona una lista estática de los primeros 100 `RUNNABLE` trabajos que están en cola. Puede utilizar el campo de búsqueda para reducir la lista buscando información en cualquier columna de la sección de resultados. Los trabajos del área de resultados instantáneos se ordenan según la estrategia de ejecución de la cola de trabajos. En el first-in-first caso de las colas de trabajos de -out (FIFO), el orden de los trabajos se basa en el tiempo de envío. En el Utilice una programación equitativa para ayudar a programar los trabajos caso de las colas de trabajos, el orden de los trabajos se basa en la prioridad de los trabajos y en el uso de recursos compartidos.

Como los resultados son una instantánea de la cola de trabajos, la lista de resultados no se actualiza automáticamente. Para actualizar la lista, selecciona la opción de actualización en la parte superior de la sección. Elija el hipervínculo del nombre del trabajo para ir a Detalles del trabajo y ver el estado del trabajo y otra información relacionada.

Para buscar esta información mediante el AWS CLI, utilice la `GetJobQueueSnapshot`operación junto con el nombre de la cola de trabajos o el nombre correspondiente. ARN

Límites estatales de trabajo

Utilice esta pestaña para revisar la información de configuración sobre el tiempo que un trabajo puede permanecer en un `RUNNABLE` estado antes de que se cancele.

Para buscar esta información mediante el AWS CLI, utilice la `DescribeJobQueues`operación junto con el nombre de la cola de trabajos o el nombre correspondienteARN.

Orden del entorno

Si su cola de trabajos se ejecuta en varios entornos, esta pestaña proporciona su orden y una descripción general.

Para encontrar esta información a través de AWS CLI, utilice la `DescribeJobQueues`operación junto con el nombre de la cola de trabajos o el nombre correspondiente. ARN

Etiquetas

Utilice esta pestaña para revisar y gestionar las etiquetas asociadas a esta cola de trabajos.

JSON

Utilice esta pestaña para copiar el JSON código asociado a esta cola de trabajos. A continuación, puede volver a utilizarla JSON para AWS CloudFormation plantillas y AWS CLI scripts.

Políticas de programación de cuotas justas

El AWS Batch programador 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 utilizará de forma predeterminada la estrategia AWS Batch de primero en entrar, primero en salir (). FIFO Una FIFO estrategia 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 de computación de acuerdo con sus necesidades específicas.

> **Note**
>
> Si desea programar el orden específico en el que se ejecutan los trabajos, utilice el `dependsOn` parámetro in SubmitJobpara especificar las dependencias de cada trabajo.

Si crea una política de programación y la adjunta a una cola de trabajos, se activará la programación de reparto justo. 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 Utilice políticas de programación para asignar identificadores de participación equitativa.

Temas

- Utilice identificadores de compartición para identificar las cargas de trabajo
- Utilice políticas de programación para asignar identificadores de participación equitativa
- Utilice una programación equitativa para ayudar a programar los trabajos
- Tutorial: Crear una política de programación
- Referencia: plantilla de política de programación

Utilice identificadores de compartición para identificar las cargas de trabajo

Puede usar los identificadores de recursos compartidos para etiquetar los trabajos y diferenciar entre usuarios y cargas de trabajo. El AWS Batch planificador realiza un seguimiento del uso de cada

identificador de participación equitativa mediante la *(T * weightFactor)* fórmula, donde *T*es el CPU uso de v 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 reparto justo 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 envíen a la cola de trabajos se programarán en FIFO orden. Al enviar trabajos, no puede especificar identificadores de cuota ni prioridades de programación.

> **Note**
>
> Los recursos de computación adjuntos se asignan por igual entre todos los identificadores de recursos compartidos, a menos que se anulen de forma explícita.

Utilice políticas de programación para asignar identificadores de participación equitativa

Puede utilizar las políticas de programación para configurar cómo se asignan los recursos de computación 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 Batch asigna a cada identificador de reparto equitativo 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 de computación para los identificadores de reparto justo que no estén activos. Para obtener más información, consulte SchedulingPolicyDetail.

Utilice una programación equitativa para ayudar a programar los trabajos

La programación de reparto justo proporciona un conjunto de controles para ayudar a programar los trabajos.

> **Note**
>
> Para obtener más información sobre los parámetros de la política de programación, consulte SchedulingPolicyDetail.

- Segundos de reducción de las acciones: período de tiempo (en segundos) que el AWS Batch programador utiliza para calcular un porcentaje de acciones equitativas para cada identificador de acciones justas. 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.

Cuando la cola de trabajos esté activa y procesando trabajos, puede revisar una lista de los primeros 100 `RUNNABLE` trabajos a través de la instantánea de la cola de trabajos. Para obtener más información, consulte Ver el estado de la cola de trabajos.

Tutorial: Crear 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 justo o prefijos de identificador de reparto justo 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. Abre 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 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 de computación durante un tiempo más largo al programar los trabajos. Esto puede permitir que los trabajos que utilizan un identificador de reparto justo utilicen temporalmente más recursos de computación de los que permitiría el peso de ese identificador si ese identificador de reparto justo 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 cómputo reservará una parte vCPUs para utilizarla como identificadores de acciones equitativas que no estén activos actualmente.

La ratio reservada ActiveFairShareses (`computeReservation/100)^ActiveFairShares` el número de identificadores de acciones equitativas activos.

Por ejemplo, un `computeReservation` valor de 50 indica que AWS Batch debe reservarse el 50% del máximo disponible VCPU si solo hay un identificador de acciones equitativas, el 25% si hay dos identificadores de acciones equitativas y el 12,5% si hay tres identificadores de acciones equitativas. Un `computeReservation` valor de 25 indica que AWS Batch debería reservarse el 25% del máximo disponible VCPU si solo hay un identificador de acciones equitativas, el 6,25% si hay dos identificadores de acciones equitativas y el 1,56% si hay tres identificadores de acciones equitativas.

7. En la sección Atributos de cuota, puede especificar el identificador de reparto justo y el peso de cada identificador de reparto justo 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 justo que se utiliza para hacer coincidir los identificadores de reparto justo de los puestos de trabajo. Todos los identificadores de reparto justo y los prefijos de los identificadores de reparto justo de una política de programación deben ser únicos y no pueden superponerse. Por ejemplo, los identificadores de reparto justo no pueden tener el prefijo “UserA*” y el identificador de reparto justo “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 de computación. Si se utiliza un prefijo identificador de reparto justo, los trabajos con identificadores de reparto justo 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 justo.

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 Etiquete sus AWS Batch recursos.

9. Seleccione Enviar para terminar y crear su política de programación.

Referencia: plantilla de política de programación

A continuación, se muestra una plantilla de política de programación vacía. Puede usar esta plantilla para crear su política de programación, 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 CreateSchedulingPolicyla AWS Batch APIReferencia.

> **Note**
>
> Puede generar una plantilla de cola de trabajos con el siguiente AWS CLI comando.
>
> ```
> $ aws batch create-scheduling-policy --generate-cli-skeleton
> ```

```
{
    "name": "",
    "fairsharePolicy": {
        "shareDecaySeconds": 0,
        "computeReservation": 0,
        "shareDistribution": [
            {
                "shareIdentifier": "",
                "weightFactor": 0.0
            }
        ]
    },
    "tags": {
        "KeyName": ""
    }
}
```

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.

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

- Qué imagen de Docker usar con el contenedor de su trabajo.
- Cuánta memoria vCPUs y cuánta memoria usar con el contenedor.
- El comando que debe ejecutar el contenedor al iniciarse.
- Qué variables de entorno (si las hay) deben pasarse al contenedor cuando se inicie.
- Cualquier volumen de datos que deba usarse con el contenedor.
- Qué IAM función (si la hay) debe utilizar su trabajo para obtener AWS permisos.

Contenido

- Crear una definición de trabajo de un solo nodo
- Cree una definición de trabajo paralelo de varios nodos
- Referencia: plantilla de definición de trabajo que utiliza ContainerProperties
- Cree definiciones de trabajo mediante EcsProperties
- Utilice el controlador de registro awslogs
- Especificar datos confidenciales
- Autenticación de registro privado para trabajos
- EFSVolúmenes de Amazon
- Referencia: ejemplos de definiciones de trabajo

Crear una definición de trabajo de un solo nodo

Antes de poder ejecutar trabajos AWS Batch, debe crear una definición de trabajo. Este proceso varía ligeramente para los trabajos paralelos entre un solo nodo y de varios nodos. Este tema trata específicamente sobre cómo crear una definición de trabajo para un AWS Batch trabajo que no es un trabajo paralelo de varios nodos (también conocido como programación de bandas).

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 “Cree una definición de trabajo paralelo de varios nodos”.

Temas

- Tutorial: Crear una definición de trabajo de un solo nodo en los recursos de Amazon EC2
- Tutorial: Cree una definición de trabajo de un solo nodo en los recursos de Fargate
- Tutorial: Crear una definición de trabajo de un solo nodo en los recursos de Amazon EKS

Tutorial: Crear una definición de trabajo de un solo nodo en los recursos de Amazon EC2

Complete los siguientes pasos para crear una definición de trabajo de un solo nodo en los recursos de Amazon Elastic Compute Cloud (AmazonEC2).

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

1. Abra la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación, elija la Región de AWS que desee utilizar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione Crear.
5. Para el tipo de orquestación, elija Amazon Elastic Compute Cloud (AmazonEC2).
6. Para configurar EC2 la plataforma, 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) Activa Propagar etiquetas para propagar etiquetas del trabajo y la definición del trabajo a la tarea de AmazonECS.
14. Seleccione Página siguiente.
15. En la sección Configuración del contenedor:
    a. En Imagen, selecciona Docker imagen para utilizarla en su trabajo. De forma predeterminada, las imágenes de Docker Hub el registro está disponible. 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` la sección Crear un contenedor del Docker Remote API y al `IMAGE` parámetro de docker run.

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

ejemplo: Arm basado Docker las imágenes solo pueden ejecutarse en Arm recursos de cómputo basados.

- Las imágenes de los repositorios ECR públicos de Amazon utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o de `registry/repository[@digest]` nomenclatura (por ejemplo,`public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de ECR los repositorios de Amazon utilizan la convención de `registry/repository[:tag]` nomenclatura completa (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Las imágenes están en los repositorios oficiales de Docker Hub usa un solo nombre (por ejemplo, `ubuntu` o`mongo`).
- Imágenes de otros repositorios en Docker Hub están calificadas con el nombre de una 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 JSON el resultado es correcto y se pasa al Docker daemon. Para comandos más complicados (por ejemplo, con caracteres especiales), utilice la JSONsintaxis.

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

Este parámetro se asigna a `Cmd` la sección Crear un contenedor del Docker Remote API y el `COMMAND` parámetro a docker run. Para obtener más información acerca de Docker `CMD`parámetro, 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 Función de ejecución, especifica una IAM función que conceda permiso a los agentes de ECS contenedores de Amazon para realizar AWS API llamadas en tu nombre. Esta función utiliza los ECS IAM roles de Amazon para las tareas. Para obtener más información, consulte las IAMfunciones de ejecución de ECS tareas de Amazon en la Guía para desarrolladores de Amazon Elastic Container Service.

e. Para configurar el rol de trabajo, elija un IAM rol que tenga permisos para AWS APIs. Esta función utiliza los ECS IAM roles de Amazon para las tareas. Para obtener más información, consulte IAMRoles 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 IAM rol para sus AWS Batch trabajos, consulte Creación de un IAM rol 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. Para vCPUs, introduzca el número de vCPUs que desea reservar para el contenedor. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción a docker run. Cada v CPU equivale a 1024 accionesCPU. 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` la sección Crear un contenedor del Docker

Remote API y a la `--memory` opción a 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 memoria de recursos informáticos.

c. En Número de GPUs, elija el número de GPUs 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 LogConfiguration: secretOptions.

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` la sección Crear un contenedor del Docker Remote API y a la `--user` opción a 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` la sección Crear un contenedor del Docker Remote API y a la `--privileged` opción a. 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 Filesystem:

    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:swappinessLinuxParameters.

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, y. READWRITEMKNOD

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 la opción Habilitar 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 en 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 LogConfiguration: logDriver.

> **Note**
>
> Se utiliza el controlador de registro `awslogs` de forma predeterminada.

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 LogConfiguration: secretOptions.

22. Seleccione Página siguiente.
23. 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.

Tutorial: Cree una definición de trabajo de un solo nodo en los recursos de Fargate

Complete los siguientes pasos para crear una definición de trabajo de un solo nodo en los recursos. AWS Fargate

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

1. Abra la AWS Batch consola en. https://console.aws.amazon.com/batch/
2. En la barra de navegación superior, selecciona la Región de AWS que deseas usar.
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 Entornos de cómputo 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 de computación.
    b. En Familia de sistemas operativos, seleccione el sistema operativo para el entorno de computación.
    c. En CPUArquitectura, elija la CPU arquitectura v.
    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 un trabajo que se ejecuta en una subred privada para enviar tráfico saliente a Internet, la subred privada requiere que se conecte una NAT puerta de enlace para enrutar las solicitudes a Internet. Es posible que desee hacer esto para poder extraer imágenes de contenedores. Para obtener más información, consulte las redes de ECS tareas de Amazon 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. En Función de ejecución, especifica una IAM función que conceda permiso al ECS contenedor de Amazon y a los agentes de Fargate para realizar AWS API llamadas en tu nombre. Esta función utiliza los ECS IAM roles de Amazon para la funcionalidad de las tareas. Para obtener más información, incluidos los requisitos previos de configuración, consulte las IAMfunciones de ejecución de ECS tareas de Amazon 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` la sección Crear un contenedor del Docker Remote API y al `IMAGE` parámetro de. docker run

> **Note**
>
> Docker la arquitectura de la imagen debe coincidir con la arquitectura del procesador de los recursos informáticos en los que están programadas. Por ejemplo: Arm basado Docker las imágenes solo pueden ejecutarse en Arm recursos de cómputo basados.

- Las imágenes de los repositorios ECR públicos de Amazon utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o de `registry/repository[@digest]` nomenclatura (por ejemplo,`public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de ECR los repositorios de Amazon utilizan la convención de `registry/repository[:tag]` nomenclatura completa (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Las imágenes están en los repositorios oficiales de Docker Hub usa un solo nombre (por ejemplo, `ubuntu` o`mongo`).
- Imágenes de otros repositorios en Docker Hub están calificadas con el nombre de una 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 JSON el resultado es correcto. Se ha pasado al Docker demonio. Para comandos más complicados (por ejemplo, con caracteres especiales), utilice la JSONsintaxis.

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

Este parámetro se asigna a `Cmd` la sección Crear un contenedor del Docker Remote API y el `COMMAND` parámetro a docker run. Para obtener más información sobre el Docker `CMD`parámetro, 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 configurar el rol de Job, elija un IAM rol que tenga permisos para AWS APIs. Esta función utiliza los ECS IAM roles de Amazon para la funcionalidad de las tareas. Para obtener más información, consulte IAMFunciones 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 IAM rol para sus AWS Batch trabajos, consulte Creación de un IAM rol y una política para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

ii. Para vCPUs, introduzca el número de vCPUs que desea reservar para el contenedor. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción a docker run. Cada v CPU equivale a 1024 accionesCPU. 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` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción a docker run. Debe especificar al menos 4 MiB de memoria para un trabajo.

Si utiliza GuardDuty Runtime Monitoring, el agente de GuardDuty seguridad tiene una ligera sobrecarga de memoria. 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 CPUlos límites de 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 ECSAmazon.

> **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 memoria de recursos informáticos.

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, consulteLogConfiguration:. secretOptions

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 pasan al Docker daemon en 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 LogConfiguration: logDriver.

   > **Note**
   >
   > Se utiliza el controlador de registro `awslogs` de forma predeterminada.

   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 LogConfiguration: secretOptions.

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.

Tutorial: Crear una definición de trabajo de un solo nodo en los recursos de Amazon EKS

Complete los siguientes pasos para crear una definición de trabajo de un solo nodo en Amazon Elastic Kubernetes Service (Amazon). EKS

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

1. Abra la AWS Batch consola en https://console.aws.amazon.com/batch/.
2. En la barra de navegación superior, selecciona la Región de AWS que deseas usar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione 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 el EKS pod sección de propiedades:

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

    c. Como DNSpolítica, elija una de las siguientes opciones:

- Sin valor (nulo): el pod ignora la DNS configuración del Kubernetes entorno.
- Predeterminado: El pod hereda la configuración de resolución de nombres del nodo en el que se ejecuta.

> **Note**
>
> Si no se especifica una DNS política, Predeterminada no es la DNS política predeterminada. En su lugar, ClusterFirstse usa.

- ClusterFirst— Cualquier DNS consulta que no coincida con el sufijo del dominio del clúster configurado se reenvía al servidor de nombres ascendente que se hereda del nodo.
- ClusterFirstWithHostNet— Úselo si la red 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 Docker imagen para utilizarla en 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` la sección Crear un contenedor del Docker Remote API y al `IMAGE` parámetro de docker run

> **Note**
>
> Docker la arquitectura de la imagen debe coincidir con la arquitectura del procesador de los recursos informáticos en los que están programados. Por ejemplo: Arm basado Docker las imágenes solo pueden ejecutarse en Arm recursos de cómputo basados.

- Las imágenes de los repositorios ECR públicos de Amazon utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o de `registry/repository[@digest]` nomenclatura (por ejemplo,`public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Las imágenes de ECR los repositorios de Amazon utilizan la convención de `registry/repository[:tag]` nomenclatura completa (por ejemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Las imágenes están en los repositorios oficiales de Docker Hub usa un solo nombre (por ejemplo, `ubuntu` o`mongo`).
- Imágenes de otros repositorios en Docker Hub están calificadas con el nombre de una 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 el campo Comando, introduzca un Bash o JSON comando 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. Para vCPUs, introduzca el número de vCPUs que desea reservar para el contenedor. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción a docker run. Cada v CPU equivale a 1024 accionesCPU. 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` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción a 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 memoria de recursos informáticos.

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` la sección Crear un contenedor del Docker Remote API y a la `--privileged` opción a. 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 Ejecutar como usuario no root t para ejecutar los contenedores del pod como usuario no root.

> **Note**
>
> Si la opción Ejecutar como no root 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, elija Editar. Cuando haya terminado, seleccione Crear definición de trabajo.

Cree una definición de trabajo paralelo de varios nodos

Antes de poder ejecutar los trabajos AWS Batch, debe crear una definición de trabajo. Este proceso varía ligeramente para los trabajos paralelos entre un solo nodo y de varios nodos. Este tema trata específicamente sobre cómo crear una definición de trabajo para un trabajo paralelo de AWS Batch varios nodos (también conocido como programación de bandas). Para obtener más información, consulte Trabajos paralelos de varios nodos.

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

Contenido

- Tutorial: Crear una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2

Tutorial: Crear una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2

Crear una definición de trabajo paralelo de varios nodos en los recursos de Amazon Elastic Compute Cloud (AmazonEC2).

> **Note**
>
> Para crear una definición de trabajo de un solo nodo, consulte. Tutorial: Crear una definición de trabajo de un solo nodo en los recursos de Amazon EC2

Para crear una definición de trabajo paralelo de varios nodos en EC2 los recursos de Amazon:

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 Definiciones de trabajo.
4. Seleccione Crear.
5. Para el tipo de orquestación, elija Amazon Elastic Compute Cloud (AmazonEC2).
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 puedes activar Propagar etiquetas para propagar etiquetas del trabajo y la definición del trabajo a la tarea de AmazonECS.
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, selecciona Docker imagen para utilizarla en su trabajo. De forma predeterminada, las imágenes de Docker Hub están disponibles en el registro. 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` la sección Crear un contenedor del Docker Remote API y al `IMAGE` parámetro de docker run.

       > **Note**
       >
       > Docker la arquitectura de la imagen debe coincidir con la arquitectura del procesador de los recursos informáticos en los que están programadas. Por ejemplo: Arm basado Docker las imágenes solo pueden ejecutarse en Arm recursos de cómputo basados.

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

- Imágenes de otros repositorios en Docker Hub están calificadas con el nombre de una 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 JSON el resultado es correcto. El JSON resultado se pasa al Docker daemon. Para comandos más complicados (por ejemplo, con caracteres especiales), puede cambiar a la JSONpestaña e introducir allí la matriz de cadenas equivalente.

   Este parámetro se asigna a `Cmd` la sección Crear un contenedor del Docker Remote API y el `COMMAND` parámetro a docker run. Para obtener más información sobre Docker `CMD`parámetro, 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. Para vCPUs, especifique el número de que desea reservar vCPUs para el contenedor. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción a docker run. Cada v CPU equivale a 1024 accionesCPU. 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` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción a 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 memoria de recursos informáticos.

h. (Opcional) En Número de GPUs, especifique el número de usos de GPUs su trabajo. El trabajo se ejecuta en un contenedor y el número especificado de ellos GPUs están anclados a ese contenedor.

i. (Opcional) Para el rol Job, puede especificar un IAM rol que proporcione al contenedor de su trabajo permisos para usar el AWS APIs. Esta función utiliza los ECS IAM roles de Amazon para la funcionalidad de las tareas. Para obtener más información, incluidos los requisitos previos de configuración, consulte IAMRoles 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 IAM rol para sus AWS Batch trabajos, consulte Creación de un IAM rol y una política para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

j. (Opcional) En Función de ejecución, especifica una IAM función que conceda permiso a los agentes de ECS contenedores de Amazon para realizar AWS API llamadas en tu nombre. Esta función utiliza los ECS IAM roles de Amazon para la funcionalidad de las tareas. Para obtener más información, consulte las IAMfunciones de ejecución de ECS tareas de Amazon 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. Para la configuración del rol de trabajo, puede especificar un IAM rol que proporcione al contenedor de su trabajo permisos para usar el AWS APIs. Esta función utiliza los ECS IAM roles de Amazon para la funcionalidad de las tareas. Para obtener más información, incluidos los requisitos previos de configuración, consulte IAMRoles 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 IAM rol para sus AWS Batch trabajos, consulte Creación de un IAM rol y una política para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

c. En Función de ejecución, especifica una IAM función que conceda permiso a los agentes de ECS contenedores de Amazon para realizar AWS API llamadas en tu nombre. Esta función utiliza los ECS IAM roles de Amazon para la funcionalidad de las tareas. Para obtener más información, consulte las IAMfunciones de ejecución de ECS tareas de Amazon 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` la sección Crear un contenedor del Docker Remote API y a la `--privileged` opción a docker run.

b. (Opcional) En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor. Este parámetro se asigna a `User` la sección Crear un contenedor del Docker Remote API y a la `--user` opción a. 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 LogConfiguration: secretOptions.

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) Activa Activar init proceso para ejecutar un `init` proceso 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 habilitado, el valor predeterminado es 60. Para obtener más información, consulte:swappinessLinuxParameters.

    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, y. READWRITEMKNOD

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 Docker daemon en una instancia contenedora. 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 puedes 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 LogConfiguration: logDriver.

    > **Note**
    > Se utiliza el controlador de registro `awslogs` de forma predeterminada.

    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 LogConfiguration: secretOptions.

27. Seleccione Página siguiente.
28. 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.

Referencia: plantilla de definición de trabajo que utiliza 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 JobDefinition.

> **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
> ```

```
{
    "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
                    }
                }
            ]
        }
    }
}
```

Referencia: 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 EKS propiedades de Amazon para la definición del trabajo que son necesarias para los trabajos que se ejecutan en EKS los recursos de Amazon
- 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

- Nombre de la definición de trabajo
- Tipo
- Parámetros
- Propiedades de contenedor
- EKSPropiedades de Amazon
- 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 Cree 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` la sección Crear un contenedor del Docker Remote API y el `COMMAND` parámetro a docker run. Para obtener más información sobre el `CMD` parámetro de Docker, consulta 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` la sección Crear un contenedor del Docker Remote y a la opción a. API `--env` 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, puede especificar un IAM rol. El rol proporciona al agente de ECS contenedores de Amazon permisos para realizar en tu nombre las API acciones que se especifican en sus políticas asociadas. Para los trabajos que se ejecutan en recursos de Fargate, debe proporcionar un rol de ejecución. Para obtener más información, consulte Rol de ejecución de AWS Batch IAM.

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 EC2 recursos 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` la sección Crear un contenedor del Docker Remote API y al `IMAGE` parámetro de. docker run

> **Note**
>
> Docker la arquitectura de la imagen debe coincidir con la arquitectura del procesador de los recursos informáticos en los que están programadas. Por ejemplo: Arm basado Docker las imágenes solo pueden ejecutarse en Arm recursos de cómputo basados.

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

- Las imágenes de ECR los repositorios de Amazon utilizan la convención de `registry/repository:[tag]` nomenclatura completa. 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

Requerido: no

`jobRoleArn`

Al registrar una definición de trabajo, puede especificar un IAM rol. El rol proporciona al contenedor de trabajos permisos para ejecutar en su nombre las API acciones que se especifican en sus políticas asociadas. Para obtener más información, consulte IAMRoles 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 asigna a `Devices` la sección Crear un contenedor del Docker Remote API y a la `--device` opción 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 del Docker Remote API o superior en la instancia de contenedor. Para comprobar la API versión de Docker Remote en tu instancia de contenedor, inicia sesión en tu instancia de contenedor y ejecuta el siguiente comando: `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**
>
> Los ECS optimizados para Amazon AMIs no tienen el intercambio activado de forma predeterminada. Debe habilitar el intercambio en la instancia para utilizar esta característica. Para obtener más información, consulta los volúmenes de intercambio de almacenes de instancias en la Guía del EC2 usuario de Amazon o ¿Cómo puedo asignar memoria para que funcione como espacio de intercambio en una EC2 instancia de Amazon mediante un archivo de intercambio? .

- Los parámetros del espacio de intercambio solo se admiten para las definiciones de trabajos que utilizan EC2 recursos.
- 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 tmpfs montar.

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 absoluta del archivo en el contenedor donde está tmpfs el volumen está montado.

Tipo: cadena

Obligatorio: sí

`mountOptions`

La lista de tmpfs opciones de montaje de volumen.

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 tmpfs volumen.

Tipo: entero

Obligatorio: sí

`logConfiguration`

La especificación de configuración de registro para el trabajo.

Este parámetro se asigna a `LogConfig` la sección Crear un contenedor del Docker Remote API y a la `--log-driver` opción 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 del Docker Remote API o superior en la instancia de contenedor. Para comprobar la API versión de Docker Remote en tu instancia de contenedor, inicia sesión en tu instancia de contenedor y ejecuta el siguiente comando: `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 `awslogs` registro. Los valores válidos que se muestran para este

parámetro son controladores de registro con los que el agente de ECS contenedores de Amazon puede comunicarse de forma predeterminada.

Este parámetro se asigna a `LogConfig` la sección Crear un contenedor del Docker Remote API y a la `--log-driver` opción 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 disponibles en futuras versiones del agente de ECS contenedores de Amazon.

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 del Docker Remote API o superior en la instancia de contenedor. Para comprobar la API versión de Docker Remote en tu instancia de contenedor, inicia sesión en tu instancia de contenedor y ejecuta el siguiente comando: `sudo docker version | grep "Server API version"`

> **Note**
>
> El agente ECS contenedor de Amazon que se ejecuta en una instancia de contenedor debe registrar los controladores de registro disponibles en esa instancia con la variable de `ECS_AVAILABLE_LOGGING_DRIVERS` entorno. De lo contrario, los contenedores ubicados en esa instancia no pueden usar estas opciones de configuración de registro. Para obtener más información, consulte Configuración de Amazon ECS Container Agent 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 Utilice el controlador de registro 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 JSON de registro de archivos. Para obtener más información, incluidos el uso y las opciones, consulte JSONel controlador de registro de archivos 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 tienes un controlador personalizado que no aparece en la lista anterior y te gustaría trabajar con el agente de ECS contenedores de Amazon, puedes bifurcar el proyecto de agente de ECS contenedores de Amazon 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 del Docker Remote API o superior en tu 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 Especificar datos confidenciales.

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 del recurso de Amazon (ARN) del secreto que se va a exponer en la configuración de registro del contenedor. Los valores admitidos son la totalidad ARN del secreto de Secrets Manager o la totalidad ARN del parámetro en el almacén de SSM parámetros.

> **Note**
>
> Si el SSM parámetro del almacén de parámetros existe en la Región de AWS misma tarea que está iniciando, puede utilizar el parámetro completo ARN o su nombre. Si el parámetro existe en una región diferente, ARN debe especificarse el parámetro completo.

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` la sección Crear un contenedor del Docker Remote API y a la `--volume` opción a 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 EC2 recursos 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` la sección Crear un contenedor del Docker Remote API y a la `--privileged` opción a 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` la sección Crear un contenedor del Docker Remote API y a la `--read-only` opción a. 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 ejemplares físicos GPUs que se van a reservar para el contenedor. El número de contenedores GPUs reservados para todos los contenedores de un trabajo no puede superar el número de contenedores disponibles GPUs en el recurso informático en el que se inicia el trabajo.

tipo=»» 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 asigna a `Memory` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción Docker Run. Debe especificar al menos 4 MiB de memoria para un trabajo. Esto es obligatorio, pero se puede especificar en varios lugares para los trabajos parallel (MNP) de varios nodos. Debe especificarse para cada nodo al menos una vez. Este parámetro se asigna a `Memory` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción 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 memoria de recursos informáticos.

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 v CPU	512, 1024 y 2048 MiB
0,5 v CPU	1024-4096 MiB en incrementos de 1024 MiB
1 v CPU	2048-8192 MiB en incrementos de 1024 MiB
2 v CPU	4096-16 384 MiB en incrementos de 1024 MiB
4 v CPU	8192-30 720 MiB en incrementos de 1024 MiB
8 v CPU	16 384-61 440 MiB en incrementos de 4096 MiB
16 v CPU	32 768-122 880 MiB en incrementos de 8192 MiB

tipo=»» VCPU

El número de personas vCPUs reservadas para el trabajo. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción Docker Run. Cada v CPU equivale a 1024 CPU acciones. Para los trabajos que se ejecutan con EC2 recursos, 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` deben coincidir con uno de los valores admitidos y los `MEMORY` valores deben ser uno de los valores admitidos para ese VCPU valor. 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 Fargate On-Demand CPU frente a los recursos es 6. vCPUs 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 Especificar datos confidenciales.

```
"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 completo del recurso de Amazon (ARN) del secreto de Secrets Manager o el parámetro completo ARN del almacén de SSM parámetros.

> **Note**
>
> Si el SSM parámetro Parameter Store existe en el Región de AWS mismo lugar que el trabajo que está lanzando, puede usar el parámetro completo ARN o el nombre. Si el parámetro existe en una región diferente, ARN debe especificarse el parámetro completo.

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` la sección Crear un contenedor del Docker Remote API y a la `--ulimit` opción a 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` la sección Crear un contenedor del Docker Remote API y a la `--user` opción a. docker run

```
"user": "string"
```

Tipo: cadena

Obligatorio: no

`vcpus`

Este parámetro ha quedado obsoleto, utilice *`resourceRequirements`* en su lugar.

El número vCPUs 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 EFSVolúmenes de Amazon.

Tipo: objeto

Obligatorio: no

`authorizationConfig`

Los detalles de configuración de autorización para el sistema de EFS archivos de Amazon.

Tipo: cadena

Requerido: no

`accessPointId`

El ID del punto de EFS acceso de Amazon 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 EFS acceso. Si se utiliza un punto de acceso, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Para obtener más información, consulte Uso de puntos de EFS acceso de Amazon en la Guía del usuario de Amazon Elastic File System.

Tipo: cadena

Requerido: no

`iam`

Determina si se debe utilizar el IAM rol de AWS Batch trabajo definido en una definición de trabajo al montar el sistema de EFS archivos de Amazon. 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 Usa los puntos de EFS acceso de Amazon.

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

`fileSystemId`

El identificador del sistema de EFS archivos de Amazon que se va a utilizar.

Tipo: cadena

Requerido: no

`rootDirectory`

El directorio del sistema de EFS archivos de Amazon que se va a montar como directorio raíz dentro del host. Si se omite este parámetro, se utilizará la raíz del EFS volumen de Amazon. 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 EFS acceso en el`authorizationConfig`, el parámetro del directorio raíz debe omitirse o establecerse en. `/` Esto refuerza la ruta establecida en el punto de EFS acceso de Amazon.

Tipo: cadena

Requerido: no

`transitEncryption`

Determina si se debe habilitar el cifrado de EFS los datos de Amazon en tránsito entre el ECS host de Amazon y el EFS servidor de Amazon. El cifrado de tránsito debe estar activado si se utiliza EFS IAM la autorización de Amazon. 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 ECS host de Amazon y el EFS servidor de Amazon. Si no especificas un puerto de cifrado de tránsito, se utilizará la estrategia de selección de puertos que utiliza el asistente de EFS montaje de Amazon. Este valor debe estar entre 0 y 65 535. Para obtener más información, consulte EFSMount Helper en la Guía del usuario de Amazon Elastic File System.

Tipo: entero

Obligatorio: no

EKSPropiedades de Amazon

Un objeto con varias propiedades que son específicas de los trabajos EKS basados en Amazon. Esto no debe especificarse para las definiciones de trabajo ECS basadas en Amazon.

`podProperties`

Las propiedades del Kubernetes recursos básicos de un trabajo.

Tipo: objeto EksPodProperties

Obligatorio: no

`containers`

Las propiedades del contenedor que se usa en el EKS pod de Amazon.

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 `args` elemento que se encuentra en la parte del punto de entrada del pod Kubernetes. Las referencias a las variables de entorno se amplían utilizando 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 CMDla referencia de Dockerfile y la definición de un comando y argumentos para un pod en el Kubernetes documentación.

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 ENTRYPOINTla referencia de Dockerfile y Definir un comando y argumentos para un contenedor y un punto de entrada en el Kubernetes documentación.

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 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 el Kubernetes documentación.

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 Kubernetes documentación.

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 memoria de recursos informáticos.

cpu

El número CPUs que está reservado 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

El número GPUs que está reservado 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 memoria de recursos informáticos.

cpu

El número de los CPUs que están reservados 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

El número de los GPUs que están reservados 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 Configurar un contexto de seguridad para un pod o contenedor en Kubernetes documentación.

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 `privileged` política de las políticas de seguridad del módulo Privileged del Kubernetes documentación.

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 corresponde con las `ReadOnlyRootFilesystem` políticas de seguridad del módulo de volúmenes y sistemas de archivos del Kubernetes documentación.

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 `RunAsGroup` las `MustRunAs` políticas de seguridad del módulo Usuarios y grupos del módulo Usuarios y grupos Kubernetes documentación.

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 `RunAsUser` las `MustRunAsNonRoot` políticas de seguridad del módulo Usuarios y grupos del módulo Usuarios y grupos Kubernetes documentación.

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 `RunAsUser` las `MustRanAs` políticas de seguridad del módulo Usuarios y grupos del módulo Usuarios y grupos Kubernetes documentación.

Tipo: largo

Obligatorio: no

`volumeMounts`

El volumen se monta para un contenedor para un EKS trabajo de Amazon. Para obtener más información sobre los volúmenes y los montajes de volúmenes, en Kubernetes, consulte Volúmenes en el Kubernetes documentación.

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 DNS política del módulo. El valor predeterminado es `ClusterFirst`. Si no se especifica el `hostNetwork` parámetro, el valor predeterminado es`ClusterFirstWithHostNet`.

`ClusterFirst`indica que cualquier DNS consulta que no coincida con el sufijo del dominio del clúster configurado se reenvía al servidor de nombres ascendente heredado del nodo. Si no se especificó ningún valor `dnsPolicy` en la RegisterJobDefinitionAPIoperación, ninguna de estas operaciones devolverá ningún valor. `dnsPolicy` DescribeJobDefinitionsDescribeJobsAPI La configuración de las especificaciones del pod contendrá `ClusterFirst` o `ClusterFirstWithHostNet`, según el valor del parámetro `hostNetwork`. Para obtener más información, consulta la DNS política de Pod en la Kubernetes documentación.

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`. Si se configura de esta manera, se `false` habilita la Kubernetes modelo de red pod. 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 los espacios de nombres de host y las redes de pods en la Kubernetes documentación.

Tipo: Booleano

Obligatorio: no

`serviceAccountName`

El nombre de la cuenta de servicio que se utiliza para ejecutar el pod. Para obtener más información, consulte Kubernetes cuentas de servicio y configurar un Kubernetes cuenta de servicio para asumir una IAM función en la Guía del EKS usuario de Amazon y configurar cuentas de servicio para los pods en el Kubernetes documentación.

Tipo: cadena

Requerido: no

`volumes`

Especifica los volúmenes de una definición de trabajo que utiliza EKS los recursos de Amazon.

Tipo: matriz de objetos EksVolume

Obligatorio: no

emptyDir

Especifica la configuración de un Kubernetes `emptyDir`volumen. 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 emptyDiren la Kubernetes documentación.

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”

Usa el `tmpfs` volumen que está respaldado por el 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

Requerido: no

sizeLimit

El tamaño máximo del volumen. De forma predeterminada, no hay un tamaño máximo definido.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

hostPath

Especifica la configuración de un Kubernetes `hostPath`volumen. 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 hostPathen la Kubernetes documentación.

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 DNS subdominio. Para obtener más información, consulte los nombres de DNS subdominios en Kubernetes documentación.

Tipo: cadena

Obligatorio: sí

secreta

Especifica la configuración de un Kubernetes `secret`volumen. Para obtener más información, consulte el secreto en el Kubernetes documentación.

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 DNS subdominio. Para obtener más información, consulte los nombres de DNS subdominios en Kubernetes documentación.

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 EKS los recursos de Amazon, no debe especificarlo`platformCapabilities`.

Tipo: cadena

Valores válidos: `EC2` | `FARGATE`

Obligatorio: no

Propagar etiquetas

`propagateTags`

Especifica si se deben propagar las etiquetas del trabajo o la definición del trabajo a la ECS tarea de Amazon 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 EKS los recursos de Amazon, no debe especificarlo`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 EKS los recursos de Amazon, no debe especificarlo`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 Etiquete sus AWS Batch recursos.

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 parallel (MNP) con varios nodos, el tiempo de espera se aplica a todo el trabajo, no a los nodos individuales.

Tipo: entero

Obligatorio: no

Cree 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 (AmazonECS), Amazon Elastic Kubernetes Service (EKSAmazon) 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",
     ...
  },
...
}
```

La 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 el AWS Batch APIs

A continuación se describen con más detalle algunas de las principales diferencias a la hora de utilizar los tipos de datos `EcsProperties` y los tipos de `EcsProperties` API datos:

- Muchos de los parámetros que se utilizan dentro `ContainerProperties` aparecen dentro`TaskContainerProperties`. Algunos ejemplos incluyen`command`,`image`, `privilegedsecrets`, 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`, `nameipcMode`, 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 RegisterJobDefinition: containerProperties and:containersEcsTaskProperties.
- `taskRoleArn`es funcionalmente equivalente a. `jobRoleArn` 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 instanceTypes objetos `taskProperties` y son matrices, pero actualmente solo aceptan un elemento. Por ejemplo, EcsProperties: taskProperties y NodeRangeProperty: instanceTypes.

Definiciones de trabajos con varios contenedores para Amazon ECS

Para adaptarse a la estructura de múltiples contenedores de AmazonECS, algunos tipos de API datos son diferentes. Por ejemplo:

- `ecsProperties`es el mismo nivel que `containerProperties` en la definición de contenedor único. Para obtener más información, consulte EcsPropertiesla Guía de AWS Batch API referencia.
- `taskProperties`contiene las propiedades definidas para la ECS tarea de Amazon. Para obtener más información, consulte EcsPropertiesla Guía AWS Batch API de referencia.

- `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:contenedores ECSTaskPropertiesen la Guía de referencia AWS Batch 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`la Guía AWS Batch API de referencia.
- Con el `dependsOn`parámetro, puede definir una lista de dependencias de contenedores. Para obtener más información, consulte `dependsOn`la Guía de AWS Batch API referencia.

> **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 la RegisterJobDefinitionsolicitud de. ecsProperties

Definiciones de trabajos con varios contenedores para Amazon EKS

Para adaptarse a la estructura de múltiples contenedores de AmazonEKS, algunos tipos de API datos 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 de 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 Kubernetes documentación.

> **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 la RegisterJobDefinitionsolicitud de. eksProperties

Referencia: escenarios de AWS Batch trabajo que utilizan 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 ECS en Amazon EC2

El siguiente es un ejemplo de un 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 Fargate

El siguiente es un ejemplo de un AWS Batch trabajo para Amazon Elastic Container Service 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 EKS

El siguiente es un ejemplo de un 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"
            }
          }
```

```
        }
      ]
    }
  }
}
```

MNP AWS Batch trabajo con varios contenedores por nodo

A continuación, se muestra un ejemplo de un AWS Batch trabajo parallel (MNP) de varios nodos 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"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      ]
    }
}
```

Utilice el controlador de registro awslogs

De forma predeterminada, AWS Batch permite que el controlador de `awslogs` registro 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 AWS Batch consola, puede configurar el controlador de `awslogs` registro 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 de `awslogs` registro simplemente pasa 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 los registros del sistema de sus instancias de contenedor a CloudWatch Logs, consulteUso CloudWatch de registros con AWS Batch. Para obtener más información sobre CloudWatch los registros, consulte Supervisión de los archivos de registro y las cuotas de CloudWatch registros en la Guía del usuario de Amazon CloudWatch Logs.

opciones del controlador de registro de awslogs en el tipo de datos AWS Batch JobDefiniton

El controlador de `awslogs` registro admite las siguientes opciones en las definiciones de AWS Batch trabajos. Para obtener más información, consulte el controlador de registro de CloudWatch registros 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 optar por enviar todos los registros de los trabajos de distintas regiones a una sola región en CloudWatch los registros. 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

Con `awslogs-stream-prefix` esta opción, puedes asociar un flujo de registro al prefijo especificado y al ID de ECS tarea de Amazon del AWS Batch trabajo 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-patternla 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. Se recomienda crear el grupo de registros con antelación mediante la CreateLogGroupAPIacción CloudWatch Registros, ya que cada trabajo intenta crear el grupo de registros, lo que aumenta la probabilidad de que el trabajo falle.

> **Note**
>
> La IAM política de su función de ejecución debe incluir el `logs:CreateLogGroup` permiso antes de intentar usarlo`awslogs-create-group`.

Especifique una configuración de registro en la definición de su trabajo

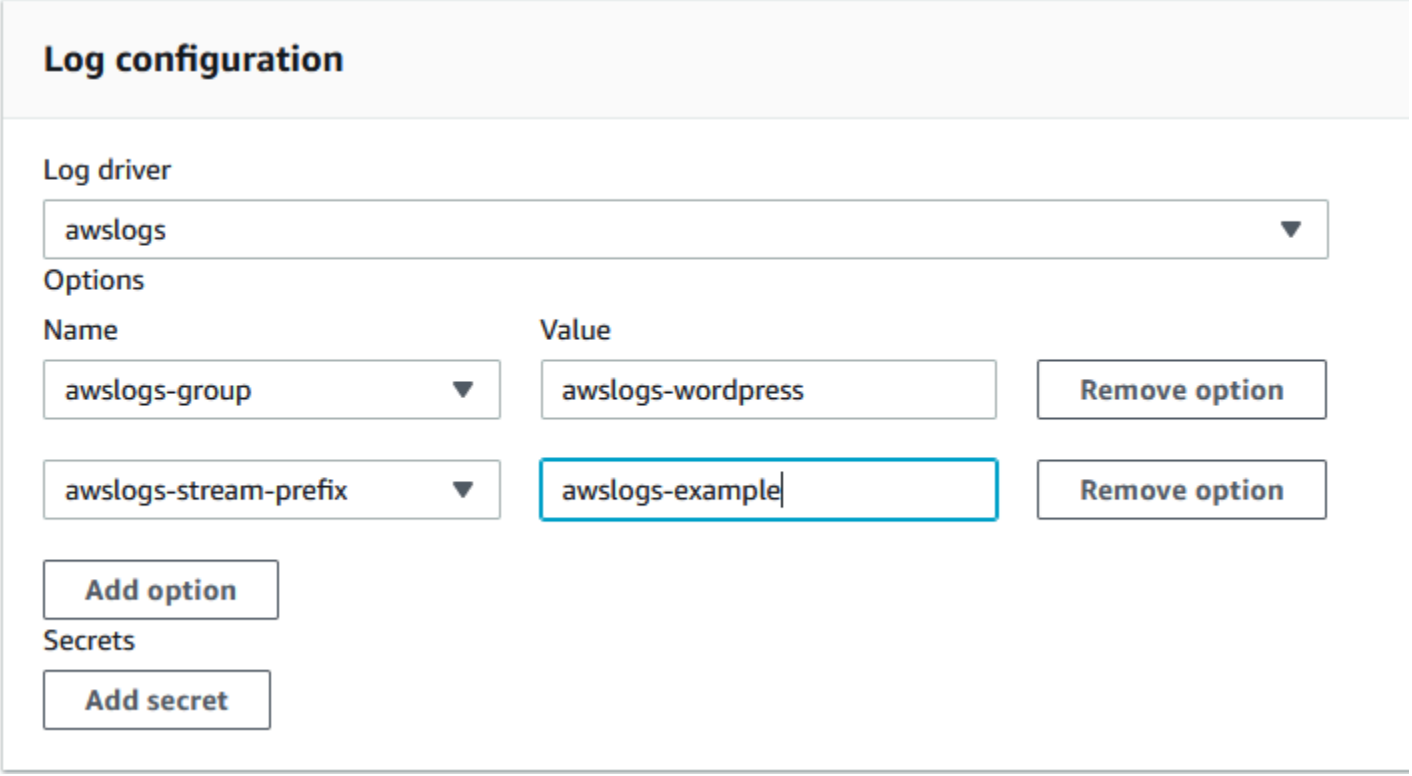
De forma predeterminada, AWS Batch habilita el controlador de `awslogs` registro. En esta sección, se describe cómo personalizar la configuración de registro `awslogs` de un trabajo. Para obtener más información, consulte Crear una definición de trabajo de un solo nodo .

Los siguientes JSON fragmentos de configuración de registro tienen un `logConfiguration` objeto especificado para cada trabajo. Uno es para un WordPress trabajo que envía registros a un grupo de registros denominado `awslogs-wordpress` y otro es para un contenedor llamado Mi SQL contenedor que envía registros a un grupo de registros 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 AWS Batch consola, se especifica la configuración de registro para la definición del `wordpress` trabajo, como se muestra en la imagen siguiente.



Tras registrar una definición de tarea con el controlador de `awslogs` registro en una configuración de registro de definición de trabajos, puede enviar un trabajo con esa definición de trabajo para empezar a enviar CloudWatch registros a Logs. Para obtener más información, consulte Tutorial: enviar un trabajo.

Especificar datos confidenciales

Con AWS Batchél, puede introducir datos confidenciales en sus trabajos almacenándolos en AWS Secrets Manager secretos o AWS Systems Manager en parámetros del almacén de parámetros y, a continuación, 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 trabajo `secrets`.
- Para hacer referencia a información sensible en la configuración del registro de un trabajo, utilice el parámetro de definición del trabajo `secretOptions`.

Temas

- Especifique los datos confidenciales con Secrets Manager
- Especifique los datos confidenciales con el almacén de parámetros de Systems Manager

Especifique los datos confidenciales con Secrets Manager

Con AWS Batchél, puede introducir datos confidenciales en sus trabajos almacenándolos en AWS Secrets Manager secreto y luego haciendo referencia a ellos en su definición de 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.

Al inyectar un secreto como variable de entorno, puede especificar la JSON clave o la versión del secreto que se va a 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 a la hora de especificar datos confidenciales 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 mediante una JSON clave específica o una versión de un secreto, la instancia contenedora de su entorno informático debe tener instalada la versión 1.37.0 o posterior del agente ECS contenedor de Amazon. No obstante, recomendamos utilizar la versión del agente de contenedor más reciente. Para obtener información sobre cómo comprobar la versión de su agente y actualizar a la última versión, consulte Actualización del agente de ECS contenedores de Amazon 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 del CreateSecretAPI. No se admiten los secretos que almacenan datos binarios, que CreateSecretAPIson secretos creados con el `SecretBinary` parámetro de.
- Cuando utilice una definición de trabajo que haga referencia a los secretos de Secrets Manager para recuperar datos confidenciales para sus trabajos, si también utiliza VPC puntos finales

de interfaz, debe crear los VPC puntos finales de interfaz para Secrets Manager. Para obtener más información, consulte Uso de Secrets Manager con VPC puntos finales en la Guía del AWS Secrets Manager usuario.

- 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.

IAMPermisos 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 Rol de ejecución de AWS Batch IAM.

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 Añadir y eliminar IAM políticas en la Guía del IAM usuario.

- `secretsmanager:GetSecretValue`: obligatorio si se referencia un secreto de Secrets Manager.
- `kms:Decrypt`—Necesario solo si el secreto utiliza una KMS clave personalizada y no la clave predeterminada. La clave ARN para tu personalizada debe añadirse como 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

Puede especificar lo siguiente en la definición de su puesto:

- El objeto `secrets` que contiene el nombre de la variable de entorno que se va a establecer en el trabajo
- El 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 JSON formato. Si no especificas una JSON clave, se utilizará todo el contenido 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`.

Las versiones se IDs utilizan para realizar un seguimiento de las diferentes versiones de un secreto cuando se actualizan o se 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 del resultado anterior en una definición de contenedor especificando el nombre de la clave al final de. 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 provisional de una versión específica de la salida anterior de una definición de contenedor especificando el nombre de la clave al final de. ARN

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::AWSPREVIOUS:"
    }]
  }]
}
```

Haga referencia a un identificador de versión específico del resultado anterior de una definición de contenedor especificando el nombre de la clave al final delARN.

```
{
  "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"
    }]
  }]
}
```

Inserte datos confidenciales en una configuración de registro

Al especificar una opción `logConfiguration` dentro de la definición de su trabajo, puede especificarla `secretOptions` con el nombre de la opción de controlador de registro que se va a configurar en el contenedor y con el secreto completo ARN de Secrets Manager que contiene los datos confidenciales que se van 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"
      }]
    }]
  }]
}
```

Cree un AWS Secrets Manager secreto

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 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. 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, una dirección de servidor o un TCP puerto. 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 puedes elegir Añadir nueva clave para crear una KMS clave personalizada específica para este secreto. Para crear tu propia KMS clave, debes tener permisos para crear KMS claves en tu cuenta.
6. Elija Next (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 secreto debe estar formado por ASCII letras, 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 Cómo cambiar tus 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.

Especifique los datos confidenciales con el almacén de parámetros de Systems Manager

Con AWS Batchél, puede inyectar datos confidenciales en sus contenedores almacenándolos en AWS Systems Manager los parámetros del almacén de parámetros y, a continuación, haciendo referencia a ellos en la definición de su contenedor.

Temas

- Consideraciones a la hora de especificar datos confidenciales mediante el almacén de parámetros de Systems Manager
- IAMPermisos necesarios para AWS Batch los secretos
- Inyecta datos confidenciales como una variable de entorno
- Inserte datos confidenciales en una configuración de registro
- Cree un AWS Systems Manager parámetro de almacén de parámetros

Consideraciones a la hora de especificar datos confidenciales mediante el almacén de parámetros 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 de su agente y actualizar a la última versión, consulte Actualización del agente de ECS contenedores de Amazon 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.

IAMPermisos necesarios para AWS Batch los secretos

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 ECS contenedores de Amazon obtener los AWS Systems Manager recursos necesarios. Para obtener más información, consulte Rol de ejecución de AWS Batch IAM.

Para proporcionar acceso a los AWS Systems Manager parámetros del almacén de parámetros que cree, añada manualmente los siguientes permisos como una política integrada a la función de ejecución. Para obtener más información, consulte Añadir y eliminar IAM políticas en la Guía del IAM usuario.

- `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`—Necesario solo si el secreto utiliza una KMS clave personalizada y no la clave predeterminada. La clave ARN para tu personalizada debe añadirse como 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>"
      ]
    }
  ]
}
```

Inyecta datos confidenciales como una variable de entorno

En la definición de contenedor, especifique `secrets` el nombre de la variable de entorno que se va a configurar en el contenedor y el parámetro completo ARN del almacén de parámetros de Systems Manager que contiene los datos confidenciales que se van 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 Almacén de parámetros de Systems Manager existe en la misma región que la tarea que está iniciando, puede usar el parámetro completo ARN o el nombre. Si el parámetro existe en una región diferente, ARN debe especificarse el parámetro completo.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Inserte datos confidenciales en una configuración de registro

Dentro de la definición de contenedor, al especificar un, `logConfiguration` puede especificarlo `secretOptions` con el nombre de la opción del controlador de registro que se va a configurar en el contenedor y con el parámetro completo ARN del almacén de parámetros de Systems Manager que contiene los datos confidenciales que se van a presentar al contenedor.

> **Important**
>
> Si el parámetro Almacén de parámetros de Systems Manager existe en la misma región que la tarea que está iniciando, puede usar el parámetro completo ARN o el nombre. Si el parámetro existe en una región diferente, ARN debe especificarse el parámetro completo.

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"
      }]
    }]
  }]
}
```

Cree 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 StringList, o SecureString.

> **Note**
>
> - Si lo elige SecureString, aparece el campo ID de KMS clave. Si no proporciona un identificador de KMS clave, una KMS claveARN, un nombre de alias o un aliasARN, el sistema utilizará`alias/aws/ssm`. Esta es la KMS clave predeterminada de 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 .

- Al crear un parámetro de cadena segura en la consola utilizando el `key-id` parámetro con un nombre de alias de KMS clave personalizado o un aliasARN, debe especificar el prefijo `alias/` antes del alias. A continuación se muestra un ARN ejemplo:

```
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 desea 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 EC2 instancias de Amazon y Fargate.

> **Important**
>
> Si la definición de tu puesto hace referencia a una imagen almacenada en AmazonECR, este tema no se aplica. Para obtener más información, consulte Uso de Amazon ECR Images con Amazon ECS en la Guía del usuario de Amazon Elastic Container Registry.

Para los trabajos alojados en EC2 instancias de Amazon, 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 de su agente y

actualizarla a la última versión, consulte Actualización del agente ECS contenedor de Amazon 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**
>
> Cuando utilices la AWS Batch API AWS CLI, o AWS SDK, si el secreto existe en la Región de AWS misma tarea que estás lanzando, puedes usar el nombre completo ARN o el nombre del secreto. Si el secreto existe en una cuenta diferente, debes especificar el secreto completoARN. Al usar el AWS Management Console, debe especificarse siempre el número completo ARN del secreto.

El siguiente es 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"
    }
  }
]
```

IAMPermisos necesarios para la autenticación del registro privado

La función de ejecución es necesaria para utilizar esta función. Esto permite que el agente de contenedor extraiga la imagen del contenedor. Para obtener más información, consulte Rol de ejecución de AWS Batch IAM.

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 Añadir y eliminar IAM políticas.

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`—Necesario solo si la clave utiliza una KMS clave personalizada y no la predeterminada. El nombre del recurso de Amazon (ARN) de la clave personalizada debe añadirse como 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"
      ]
    }
  ]
}
```

Tutorial: Cree un secreto para la autenticación del registro privado

Complete los siguientes pasos para crear un secreto para sus credenciales de registro privado con AWS Secrets Manager.

Crea un secreto básico

1. Abre 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. Elija Next (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 secreto debe estar ASCII formado por letras, dígitos o cualquiera 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 secretos nuevos o existentes, consulte Rotación de sus 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 Secrets Manager ARN o nombre, introduce el nombre del recurso de Amazon (ARN) del secreto. Para obtener más información, consulte IAMPermisos necesarios para la autenticación del registro privado.

EFSVolúmenes de Amazon

Amazon Elastic File System (AmazonEFS) proporciona un almacenamiento de archivos sencillo y escalable para usarlo en sus AWS Batch trabajos. Con AmazonEFS, 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 los sistemas de EFS archivos de Amazon AWS Batch para exportar los datos del sistema de archivos de toda su flota de instancias de contenedores. De esta forma, sus trabajos tienen acceso al mismo almacenamiento persistente. Sin embargo, debe configurar su instancia de contenedor AMI para montar el sistema de EFS archivos de Amazon 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 siguientes secciones te ayudan a empezar a utilizar Amazon EFS con AWS Batch.

Consideraciones sobre EFS el volumen de Amazon

Al utilizar EFS volúmenes de Amazon, se debe tener en cuenta lo siguiente:

- Para los trabajos que utilizan EC2 recursos, se agregó la compatibilidad con el sistema de EFS archivos de Amazon como una vista previa pública con la AMI versión ECS optimizada de Amazon `20191212` con agente contenedor versión 1.35.0. Sin embargo, el soporte del sistema de EFS archivos de Amazon entró en disponibilidad general con la AMI versión ECS optimizada de Amazon `20200319` con el agente contenedor versión 1.38.0, que contenía las funciones de IAM autorización y punto de EFS acceso de Amazon. Te recomendamos que utilices la AMI versión ECS optimizada de Amazon `20200319` o posterior para aprovechar estas funciones. Para obtener más información, consulte AMIlas versiones ECS optimizadas de Amazon en la Guía para desarrolladores de Amazon Elastic Container Service.

> **Note**
>
> Si creas el tuyo propioAMI, debes usar el agente contenedor 1.38.0 o posterior, la `ecs-init` versión 1.38.0-1 o posterior, y ejecutar los siguientes comandos en tu instancia de Amazon. EC2 Todo esto es para habilitar el complemento de ECS volumen de Amazon. 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 los recursos de Fargate, se agregó la compatibilidad con EFS el sistema de archivos de Amazon cuando se utilizó la versión 1.4.0 o posterior de la plataforma. Para obtener más información, consulte Versiones de la plataforma de AWS Fargate en la Guía para desarrolladores de Amazon Elastic Container Service.
- Al especificar EFS los volúmenes de Amazon en los trabajos que utilizan los recursos de Fargate, Fargate crea un contenedor supervisor que se encarga de gestionar 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.

Usa los puntos de EFS acceso de Amazon

Los puntos de EFS acceso de Amazon son puntos de entrada específicos de la aplicación a un sistema de EFS archivos que le ayudan a gestionar el acceso de las aplicaciones a los conjuntos de datos compartidos. Para obtener más información sobre los puntos de EFS acceso de Amazon y cómo controlar el acceso a ellos, consulte Trabajar con puntos de EFS acceso de Amazon en la Guía del usuario de Amazon Elastic File System.

Los puntos de acceso pueden imponer la identidad de un usuario, incluidos los POSIX grupos de usuarios, para todas las solicitudes del sistema de archivos que se realicen 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**
>
> Al crear un punto de EFS acceso, se especifica una ruta en el sistema de archivos que sirva de directorio raíz. Al hacer referencia al sistema de EFS archivos con un identificador de punto de acceso en la definición de su AWS Batch trabajo, el directorio raíz debe omitirse o configurarse en. De `/` este modo, se aplica la ruta establecida en el punto de EFS acceso.

Puede utilizar un IAM rol de AWS Batch trabajo para imponer que aplicaciones específicas usen un punto de acceso específico. Al combinar IAM políticas con puntos de acceso, puede proporcionar fácilmente un acceso seguro a conjuntos de datos específicos para sus aplicaciones. Esta función utiliza los ECS IAM roles de Amazon para la funcionalidad de las tareas. Para obtener más

información, consulte IAMFunciones para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

Especifica un sistema de EFS archivos de Amazon en tu definición de trabajo

Para usar los volúmenes del sistema de EFS archivos de Amazon para sus contenedores, debe especificar las configuraciones de volumen y punto de montaje en su definición de trabajo. El siguiente JSON fragmento de definición de trabajo muestra la sintaxis de los `mountPoints` objetos `volumes` y de 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 utilizan EFS volúmenes de Amazon.

`fileSystemId`

Tipo: cadena

Obligatorio: sí

El identificador del sistema de EFS archivos de Amazon que se va a utilizar.

`rootDirectory`

Tipo: cadena

Requerido: no

El directorio del sistema de EFS archivos de Amazon que se va a montar como directorio raíz dentro del host. Si se omite este parámetro, se utilizará la raíz del EFS volumen de Amazon. 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 EFS acceso en el`authorizationConfig`, el parámetro del directorio raíz debe omitirse o establecerse en. `/` Esto impone la ruta establecida en el punto de EFS acceso.

`transitEncryption`

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

Determina si se debe habilitar el cifrado de EFS los datos de Amazon que están en tránsito entre el AWS Batch host y el EFS servidor de Amazon. El cifrado de tránsito debe estar activado si se utiliza EFS IAM la autorización de Amazon. 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 AWS Batch host y el EFS servidor de Amazon. Si no especificas un puerto de cifrado de tránsito, se utilizará la estrategia de selección de puertos que utiliza el asistente de EFS montaje de Amazon. Este valor debe estar entre 0 y 65 535. Para obtener más información, consulte EFSMount Helper en la Guía del usuario de Amazon Elastic File System.

`authorizationConfig`

Tipo: objeto

Obligatorio: no

Los detalles de configuración de autorización para el sistema de EFS archivos de Amazon.

`accessPointId`

Tipo: cadena

Requerido: 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 EFS acceso. Si se utiliza un punto de acceso, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Para obtener más información, consulte Uso de puntos de EFS acceso de Amazon 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 IAM rol de AWS Batch trabajo que se define en una definición de trabajo al montar el sistema de EFS archivos de Amazon. 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 el rol de IAM ejecución, consulteRol de ejecución de AWS Batch IAM.

Referencia: ejemplos de definiciones de trabajo

Los ejemplos de definiciones de puestos de este tema ilustran cómo utilizar patrones comunes, como las variables de entorno, la sustitución de parámetros y los montajes de volumen.

Contenido

- Referencia: variables de entorno
- Referencia: sustitución de parámetros
- Referencia: Pruebe la GPU funcionalidad
- Referencia: trabajo paralelo multinodo

Referencia: variables de entorno

El siguiente ejemplo de definición de trabajo utiliza variables de entorno para especificar un tipo de archivo y Amazon S3URL. Este ejemplo en particular proviene de la entrada del blog Creating a Simple «Fetch & Run» AWS Batch Job Compute. 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://amzn-s3-demo-source-bucket/myjob.sh"
            },
            {
                "name": "BATCH_FILE_TYPE",
                "value": "script"
            }
        ],
        "user": "nobody"
    }
}
```

Referencia: 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"
    }
}
```

Referencia: Pruebe la GPU funcionalidad

El siguiente ejemplo de definición de trabajo comprueba si la GPU carga de trabajo AMI descrita en Usa una GPU carga de trabajo AMI está configurada correctamente. En este ejemplo de definición de trabajo se ejecuta el ejemplo del MNIST clasificador TensorFlow profundo utilizado. 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"
}
```

Puede crear un archivo con el JSON texto anterior denominado `tensorflow_mnist_deep.json` y, a continuación, registrar una definición de AWS Batch trabajo con el siguiente comando:

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Referencia: trabajo paralelo multinodo

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"
        }
      }
    ]
  }
}
```

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 ECS contenedores de Amazon en un ECS clúster.

Los trabajos en contenedores pueden hacer referencia a una imagen, comando o parámetros de contenedor. Para obtener más información, consulte. JobDefinition

Puede enviar una gran cantidad de trabajos sencillos e independientes.

Temas

- Tutorial: enviar un trabajo
- Estados de trabajo
- AWS Batch variables de entorno laboral
- Reintentos automáticos de trabajo
- Dependencias de trabajos
- Tiempos de espera de trabajo
- EKSEmpleos en Amazon
- Trabajos paralelos de varios nodos
- Trabajos paralelos de varios nodos en Amazon EKS
- Trabajos de matrices
- Ejecute GPU trabajos
- Busque AWS Batch trabajos en una cola de trabajos
- Tutorial: envíe los registros de AWS Batch trabajos a CloudWatch Logs
- Tutorial: Revise la información del AWS Batch trabajo

Tutorial: enviar un trabajo

Después de registrar una definición de trabajo, puede enviarla como trabajo a una lista 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 Crear una definición de trabajo de un solo nodo .
7. En Cola de trabajos, elija una cola de trabajos existente. Para obtener más información, consulte Crear 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) Activa Propagar etiquetas para propagar etiquetas del trabajo y la definición del trabajo a la tarea de AmazonECS.

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 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 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 JSONsintaxis.

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

b. Para vCPUs, introduzca el número de vCPUs que desea reservar para el contenedor. Este parámetro se asigna a `CpuShares` la sección Crear un contenedor del Docker Remote API y a la `--cpu-shares` opción a docker run. Cada v CPU equivale a 1024 accionesCPU. 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` la sección Crear un contenedor del Docker Remote API y a la `--memory` opción a 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 memoria de recursos informáticos.

d. (Opcional) En Número de GPUs, elija el número de GPUs 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 `SUBMITTED` estado. 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`.

La duración de la extracción de imágenes, EKS initContainer la duración de finalización de Amazon y la duración de la ECS containerDependency resolución de Amazon se producen en el STARTING estado. La cantidad de tiempo que se tarda en obtener una imagen para su trabajo equivale a la cantidad de tiempo que su trabajo estará en el STARTING estado.

Por ejemplo, si se tarda tres minutos en obtener la imagen de su trabajo, este estará en el STARTING estado durante tres minutos. Si initContainers tarda un total de diez minutos en completarse, tu EKS trabajo en Amazon estará disponible STARTING durante diez minutos.

Si tienes Amazon ECS containerDependencies sets en tu trabajo de Amazon, el ECS trabajo permanecerá activo STARTING hasta que se resuelvan todas las dependencias de los contenedores (su tiempo de ejecución). STARTINGno se incluye en los tiempos de espera; la duración comienza a las. RUNNING Para obtener más información, consulte Estados de trabajo.

`RUNNING`

El trabajo se ejecuta como un trabajo de contenedor en una instancia de ECS contenedor de Amazon dentro de un entorno informático. 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 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`*. Sin embargo, esto puede cambiar en el futuro.
> Cuando un trabajo alcanza el `RUNNING` estado, puede recuperar mediante programación el nombre del flujo de registro con la DescribeJobsAPIoperación. 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.
Cuando un trabajo alcanza el RUNNING estado, puede recuperar mediante programación el nombre del flujo de registro con la DescribeJobsAPIoperación. 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, puede recuperar mediante programación su flujo de registro con la DescribeJobsAPIoperación. 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: Utilice el índice de trabajos de la matriz para controlar la diferenciación de los 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 Kubernetes UIDdel 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 EKS los recursos de Amazon. Para obtener más información, consulte UIDsla 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 está configurada en la IPv4 dirección 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
- Fallo o cancelación de una EC2 instancia de Amazon
- Error o interrupción del AWS servicio interno

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, Crear una definición de trabajo de un solo nodo , Tutorial: enviar un trabajo y Códigos de error de tareas detenidas.

Dependencias de trabajos

Al enviar un AWS Batch trabajo, puede especificar el trabajo del IDs que depende. 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, consulteTutorial: enviar un trabajo.

Tiempos de espera de trabajo

Puede configurar una duración de tiempo de espera para sus trabajos de modo que si un trabajo se ejecuta durante más tiempo, AWS Batch lo finaliza. 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 cómo enviar un AWS Batch trabajo con una configuración de tiempo de espera, consulte. Tutorial: enviar un trabajo

EKSEmpleos en Amazon

Un trabajo es la unidad de trabajo más pequeña de AWS Batch. Un AWS Batch trabajo en Amazon EKS tiene un one-to-one mapeo a un Kubernetes cápsula. 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 AmazonEKS, el eksPropertiesparámetro define el conjunto de parámetros que admite un EKS trabajo AWS Batch en Amazon. 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 tu EKS clúster de Amazon, 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 EKS trabajo de Amazon, 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 tareas poco después de que un módulo 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 Usa CloudWatch Logs para monitorear AWS Batch los EKS trabajos de Amazon.

Temas

- Tutorial: Asigne un trabajo en ejecución a un pod y un nodo
- Tutorial: asigne un pod en ejecución a su función

Tutorial: 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 DescribeJobsAPIoperación 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 activados, el `podName` y `nodeName` de cada intento completado aparece en el parámetro de `eksAttempts` lista de la DescribeJobsAPIoperación. El `podName` y el `nodeName` del intento en ejecución actual se encuentra en el objeto `podProperties`.

Tutorial: asigne un pod en ejecución a su función

Un pod tiene etiquetas que indican el `jobId` extremo `uuid` del entorno 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

...
```

Funciones compatibles con AWS Batch Amazon EKS Jobs

Estas son las características AWS Batch específicas que también son comunes Kubernetes trabajos que se ejecutan en AmazonEKS:

- Dependencias de trabajos
- Trabajos de matrices
- Tiempos de espera de trabajo
- Reintentos automáticos de trabajo
- Utilice una programación equitativa para ayudar a programar los trabajos

Kubernetes**`Secrets`** y **`ServiceAccounts`**

AWS Batch admite referencias Kubernetes `Secrets` y `ServiceAccounts`. Puedes configurar los pods para que usen los EKS IAM roles de Amazon en las cuentas de servicio. Para obtener más información, consulta Cómo configurar los pods para usar un Kubernetes cuenta de servicio en la Guía del EKS usuario de Amazon.

Documentos relacionados

- CPUConsideraciones sobre memoria y v para AWS Batch Amazon EKS
- Ejecute GPU trabajos
- Trabajos bloqueados en estado `RUNNABLE`

Trabajos paralelos de varios nodos

Puede utilizar trabajos paralelos de varios nodos para ejecutar trabajos individuales que abarquen varias EC2 instancias de Amazon. Con los trabajos paralelos de AWS Batch varios nodos (también conocidos como programación de bandas), puede ejecutar aplicaciones informáticas de alto rendimiento a gran escala y entrenamiento de GPU modelos distribuidos sin necesidad de lanzar, configurar y administrar EC2 los recursos de Amazon 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 EC2 instancia de Amazon.

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 usando la AWS Batch DescribeJobsAPIoperación. 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.

Temas

- Variables de entorno
- Grupos de nodos
- Ciclo de vida laboral para los MNP trabajos
- Consideraciones sobre el entorno de cómputo para MNP con AWS Batch

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 IPv4 dirección 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 para las instancias de un ECS clúster de Amazon. 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 laboral para los MNP trabajos

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 garantizar que todos los nodos de los trabajos estén en el mismo `RUNNING` estado después de que el contenedor del nodo esté en ejecución, el código de la aplicación puede consultarlo AWS Batch API para obtener información sobre el nodo principal y el nodo secundario. 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 de la tarea. 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 sobre el entorno de cómputo para MNP con AWS Batch

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, consulta Grupos de ubicación en la Guía del EC2 usuario de Amazon.

- 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 ECS `awsvpc` red de Amazon, que proporciona a los contenedores de trabajos paralelos de varios nodos las mismas propiedades de red que las instancias de AmazonEC2. Cada contenedor de trabajos paralelos de varios nodos tiene su propia interfaz de red elástica, una dirección IP privada principal y un nombre de DNS host interno. La interfaz de red se crea en la misma VPC subred que su recurso informático anfitrión. 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 acceder a Internet, los recursos informáticos deben lanzarse en una subred privada configurada para usar una NAT puerta de enlace. Para obtener más información, consulta NATGateways en la Guía del VPC usuario de Amazon. La comunicación entre nodos debe utilizar la dirección IP privada o el DNS nombre de host del 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 NAT puerta de enlace, consulte. Creación de una nube virtual privada
- 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.
- Su entorno de cómputo debe tener un máximo suficiente vCPUs para soportar su trabajo paralelo de varios nodos.
- Tu cuota de EC2 instancias de Amazon incluye la cantidad de instancias necesarias para ejecutar tu 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 paralelos de varios nodos en Amazon EKS

Puede utilizarlos AWS Batch en Amazon Elastic Kubernetes Service para ejecutar trabajos parallel () de varios nodos MNP (también conocidos como programación de bandas) en su servidor gestionado Kubernetes clústeres. Esta opción se suele utilizar para trabajos grandes, estrechamente acoplados y de alto rendimiento que no se pueden ejecutar en una sola instancia de Amazon Elastic Compute Cloud. Para obtener más información, consulte Trabajos paralelos de varios nodos.

Puedes usar esta función para ejecutar Amazon EKS Managed Kubernetes-aplicaciones informáticas específicas de alto rendimiento, formación en modelos de lenguaje de gran tamaño y otros trabajos de inteligencia artificial (IA) y aprendizaje automático (ML).

Temas

- ¿Ejecutando trabajos? MNP
- Crear una definición de EKS MNP trabajo en Amazon
- Envía un EKS MNP trabajo en Amazon
- Anular una definición de EKS MNP trabajo de Amazon

¿Ejecutando trabajos? MNP

AWS Batch admite MNP trabajos en Amazon Elastic Container Service y Amazon a EKS través de AmazonEC2. A continuación, se proporcionan más detalles sobre los parámetros de instancia y contenedor de la función.

Cuotas de instancias para MNP Amazon EKS

- Se pueden usar hasta 1000 instancias para un solo MNP trabajo.
- Se pueden unir hasta 5000 instancias a un único EKS clúster de Amazon.
- Se pueden agrupar en clústeres y adjuntar hasta 5 entornos de cómputo a una cola de trabajos.

Por ejemplo, puede escalar hasta 5 entornos informáticos agrupados en una cola de trabajos y 1000 instancias en cada entorno informático.

Además de los parámetros de la instancia, es importante tener en cuenta que no puede usar Fargate para MNP trabajos a través de ninguno de los dos servicios.

Solo puedes usar un tipo de instancia en cada MNP trabajo. Puede cambiar el tipo de instancia actualizando el entorno de cómputo o al definir un nuevo entorno de cómputo. También puedes especificar el tipo de instancia y proporcionar v CPU y los requisitos de memoria al crear la definición del trabajo.

Cuotas de contenedores para MNP Amazon EKS

- Un trabajo paralelo de varios nodos admite un pod por nodo.
- Hasta 10 contenedores (o 10 contenedores de inicio). Para obtener más información, consulte Contenedores de inicio (en la documentación de Kubernetes) en cada pod.
- Hasta 5 rangos de nodos en cada trabajo. MNP
- Hasta 10 imágenes de contenedores distintas en cada rango de nodos.

Por ejemplo, puede ejecutar hasta un máximo de 10 000 contenedores en una sola MNP tarea que contenga 5 rangos de nodos y un total de 50 imágenes únicas.

Ejecutar MNP tareas en un Amazon privado VPC y en un EKS clúster de Amazon

MNPLos trabajos pueden ejecutarse en cualquier EKS clúster de Amazon, ya sea que tenga Internet público o no. Cuando utilices un EKS clúster de Amazon con acceso exclusivo a la red privada, asegúrate de que AWS Batch puede acceder al plano de EKS control de Amazon y al gestionado. Kubernetes APIservidor. Puede conceder el acceso necesario a través de los puntos de conexión de Amazon Virtual Private Cloud. Para obtener más información, consulte Configurar un servicio de punto final.

Los pods de EKS clúster de Amazon no pueden descargar una imagen de una fuente pública porque la privada VPC no tiene acceso a Internet. Tu EKS clúster de Amazon debe extraer imágenes de un registro de contenedores que esté en tu AmazonVPC. Puedes crear un (AmazonECR) en tu Amazon VPC y copiarle las imágenes del contenedor para acceder a tus nodos.

También puedes crear una regla de extracción de caché con AmazonECR. Una vez creada una regla de extracción de caché para un registro público externo, simplemente puedes extraer una imagen de ese registro público externo utilizando tu registro ECR privado de AmazonURI. A continuación, Amazon ECR crea un repositorio y guarda la imagen en caché. Cuando se extrae una imagen en caché mediante el registro ECR privado de AmazonURI, Amazon ECR comprueba el registro remoto para ver si hay una nueva versión de la imagen y actualizará tu registro privado hasta una vez cada 24 horas. Para obtener más información, consulta Cómo crear una regla de extracción de caché en Amazon ECR.

Para obtener más información sobre este tema, consulteTutorial: Cómo empezar a utilizar AWS Batch Amazon EKS Private Clusters.

Notificación de error

Si tus MNP trabajos están bloqueados, puedes recibir notificaciones a través de Amazon AWS Management Console y Amazon EventBridge. Por ejemplo, si un MNP trabajo está atascado al principio de la lista, puedes recibir una notificación sobre el problema junto con información sobre su causa para que puedas tomar medidas inmediatas para desbloquear tu lista de trabajos. Opcionalmente, puede finalizar automáticamente el MNP trabajo si no se realiza ninguna acción en un período de tiempo determinado, que se puede definir en la plantilla de cola de trabajos. Para obtener más información, consulte Recurso: Eventos bloqueados por la cola de trabajos

Crear una definición de EKS MNP trabajo en Amazon

Para definir y ejecutar MNP trabajos en AmazonEKS, hay nuevos parámetros en las `SubmitJob`APIoperaciones `RegisterJobDefinition`y.

- Utilízalo en la `eksPropertiesnodeProperties`sección para definir tu definición de MNP trabajo.
- Utilícelo en la `eksPropertiesOverridenodePropertyOverrides`sección para anular los parámetros definidos en la definición del trabajo al enviar un MNP trabajo.

Estas acciones se pueden definir mediante API las operaciones y el AWS Management Console.

Referencia: Registra la carga útil de la solicitud de definición de EKS MNP trabajo de Amazon

El siguiente ejemplo ilustra cómo se puede registrar una definición de EKS MNP trabajo de Amazon con dos nodos.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes" : "0:",
```

```
        "eksProperties": {
          "podProperties": {
            "containers": [
              {
                "name": "test-eks-container-1",
                "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                "command": [
                  "sleep",
                  "60"
                ],
                "resources": {
                  "limits": {
                    "cpu": "1",
                    "memory": "1024Mi"
                  }
                },
                "securityContext":{
                  "runAsUser":1000,
                  "runAsGroup":3000,
                  "privileged":true,
                  "readOnlyRootFilesystem":true,
                  "runAsNonRoot":true
               }
              }
            ],
            "initContainers": [
               {
                  "name":"init-ekscontainer",
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                  "command": [
                     "echo",
                     "helloWorld"
                   ],
                   "resources": {
                     "limits": {
                       "cpu": "1",
                       "memory": "1024Mi"
                     }
                  }
               }
            ],
            "metadata": {
               "labels": {
                  "environment" : "test"
```

```
                }
            }
          }
        }
      }
    ]
  }
}
```

Para registrar la definición de trabajo mediante el AWS CLI, copie la definición en un archivo local denominado MyEksMnpJobDefinition.json y ejecute el siguiente comando.

```
aws batch register-job-definition --cli-input-json file://MyEksMnpJobDefinition.json
```

Recibirás la siguiente JSON respuesta.

```
{
    "jobDefinitionName": "MyEksMnpJobDefinition",
    "jobDefinitionArn": "arn:aws:batch:us-east-1:0123456789:job-definition/
MyEksMnpJobDefinition:1",
    "revision": 1
}
```

Envía un EKS MNP trabajo en Amazon

Para enviar un trabajo utilizando la definición de trabajo registrada, introduce el siguiente comando. Sustituya el valor de < EKS JOB _ QUEUE _ _ NAME > por el nombre o una cola ARN de trabajos preexistente asociada a un entorno EKS informático de Amazon.

```
aws batch submit-job --job-queue <EKS_JOB_QUEUE_NAME> \
    --job-definition MyEksMnpJobDefinition \
    --job-name myFirstEksMnpJob
```

Recibirá la siguiente JSON respuesta.

```
{
    "jobArn": "arn:aws:batch:region:account:job/9b979cce-9da0-446d-90e2-ffa16d52af68",
    "jobName": "myFirstEksMnpJob",
    "jobId": "<JOB_ID>"
}
```

Puede comprobar el estado del trabajo utilizando lo devuelto jobId con el siguiente comando.

```
aws batch describe-jobs --jobs <JOB_ID>
```

Anular una definición de EKS MNP trabajo de Amazon

Si lo desea, puede anular los detalles de la definición del trabajo (como cambiar el tamaño del MNP trabajo o los detalles del trabajo secundario). A continuación, se proporciona un ejemplo de carga útil de JSON solicitud para enviar un MNP trabajo de cinco nodos y cambios en el comando del `test-eks-container-1` contenedor.

```
{
  "numNodes": 5,
  "nodePropertyOverrides": [
    {
      "targetNodes": "0:",
      "eksPropertiesOverride": {
        "podProperties": {
          "containers": [
            {
              "name": "test-eks-container-1",
              "command": [
                "sleep",
                "150"
              ]
            }
          ]
        }
      }
    }
  ]
}
```

Para enviar un trabajo con estas anulaciones, guarda el ejemplo en un archivo local, eks-mnp-job-nodeoverride.json, y úsalo AWS CLI para enviar el trabajo con las anulaciones.

Trabajos de matrices

Un trabajo de matriz es un trabajo que comparte parámetros comunes, como la definición del trabajo y la memoriavCPUs. 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.

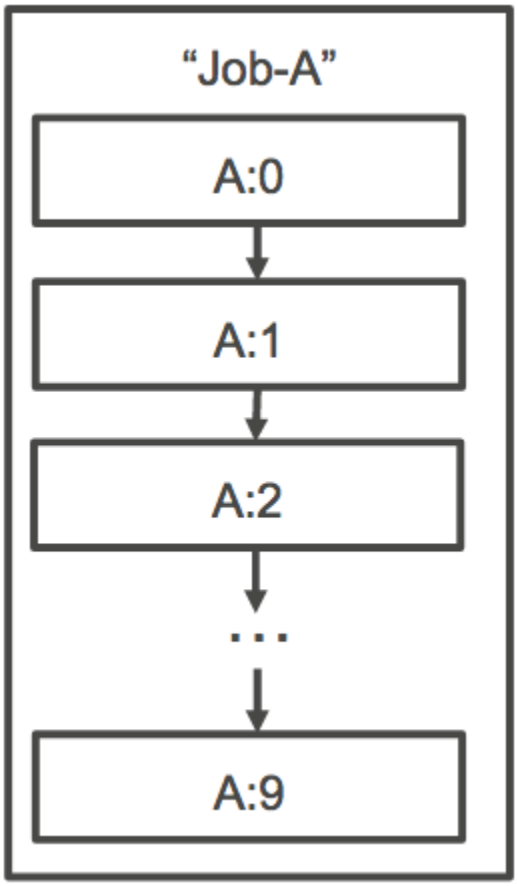
Al enviar un trabajo de matriz, el trabajo de matriz principal recibe un ID de AWS Batch trabajo normal. Cada trabajo secundario tiene el mismo ID base. Cada trabajo secundario tiene el mismo ID de base, pero su índice de matriz se añade al final del ID principal, por ejemplo, `example_job_ID:0` para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

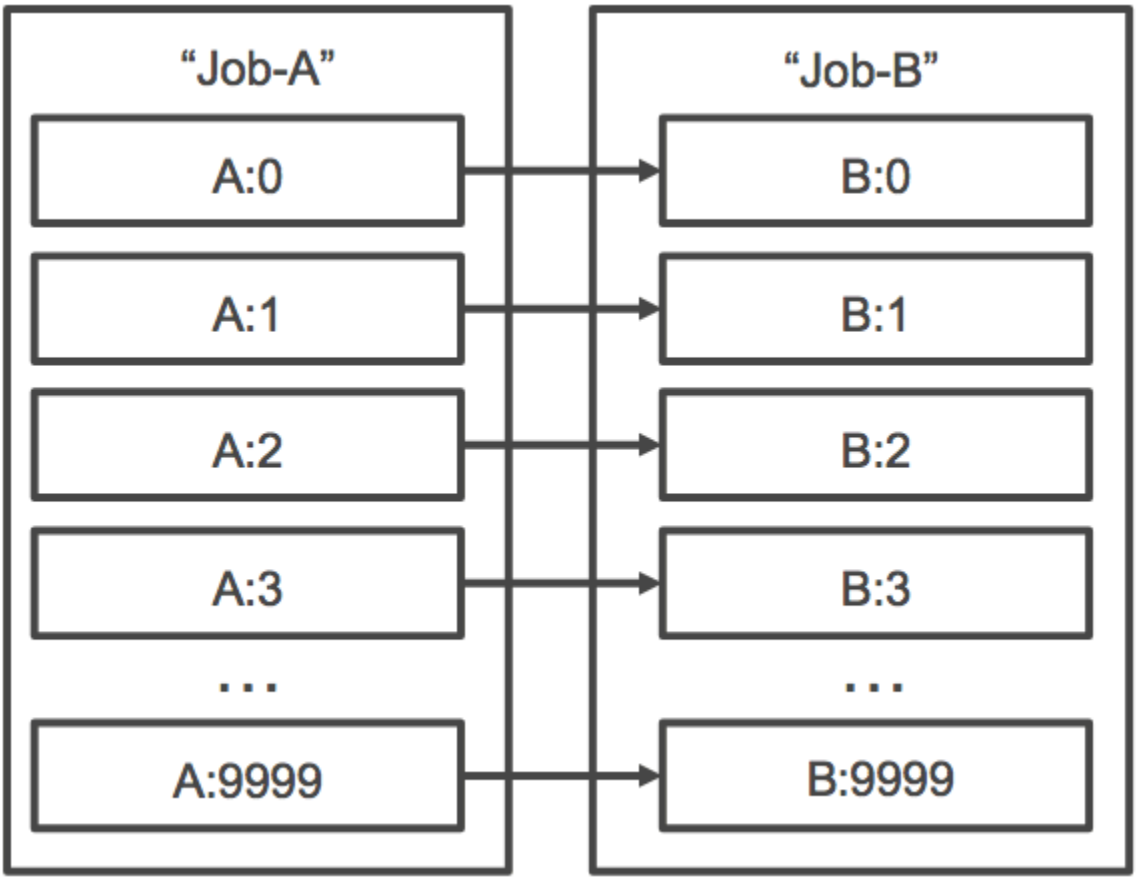
Durante el tiempo de ejecución, la variable de entorno `AWS_BATCH_JOB_ARRAY_INDEX` se establece en el número de índice del trabajo de matriz correspondiente del contenedor. Al primer índice de trabajo de matriz se le asigna el número `0`, y los trabajos secundarios posteriores se numeran en orden ascendente (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: Utilice el índice de trabajos de la matriz para controlar la diferenciación de los 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 finalizar trabajos secundarios individuales (lo que los traslada a un `FAILED` estado) sin que ello afecte a los demás trabajos secundarios. Sin embargo, si un trabajo de matriz secundaria falla (por sí solo o al cancelar o terminar manualmente el trabajo), el trabajo principal también fallará. En este escenario, el trabajo principal pasa a `FAILED` cuando se completan todos los trabajos secundarios.

Temas

- Referencia: ejemplo de un flujo de trabajo de tareas matriciales
- Tutorial: Utilice el índice de trabajos de la matriz para controlar la diferenciación de los trabajos

Referencia: ejemplo de un flujo de trabajo de tareas matriciales

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 SubmitJobJSONsintaxis es la siguiente.

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

- `JobB`: Un trabajo de matriz con 10 000 copias del que depende `JobA` que ejecute comandos CPU intensivos en cada objeto `BucketA` y en el que cargue los resultados. `BucketB` La SubmitJobJSONsintaxis 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 SubmitJobJSONsintaxis 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 SubmitJobJSONsintaxis 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 último trabajo, que no es de matriz, que realiza algunas operaciones de limpieza sencillas y envía una SNS notificación de Amazon con un mensaje de que la canalización se ha completado y un enlace al resultado. URL La SubmitJobJSONsintaxis es la siguiente.

```
{
    "jobName": "JobE",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobE-Cleanup-and-Notification:1",
    "parameters": {
        "SourceBucket": "s3://amzn-s3-demo-source-bucket",
        "Recipient": "pipeline-notifications@mycompany.com"
    },
    "dependsOn": [
        {
            "jobId": "JobD_job_ID"
        }
```

```
    ]
}
```

Tutorial: Utilice el índice de trabajos de la matriz para controlar la diferenciación de los 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 y los mismos argumentos de comando de Docker. 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 crean a partir de una imagen de Docker que se envía a Amazon. ECR A continuación, registras una definición de trabajo que utiliza tu nueva imagen de contenedor, envías un trabajo de AWS Batch matriz con esa definición de trabajo y ves los resultados.

Temas

- Requisitos previos
- Tutorial: Construye una imagen de contenedor
- Tutorial: Envía tu imagen a Amazon ECR
- Tutorial: Crear y registrar una definición de trabajo
- Tutorial: Envíe un trabajo AWS Batch de matriz
- Tutorial: consulte los registros de trabajos de su matriz

Requisitos previos

El flujo de trabajo de este tutorial tiene los siguientes requisitos previos:

- Un entorno AWS Batch informático. Para obtener más información, consulte Crear un entorno de computación.
- Una cola de AWS Batch trabajos y un entorno informático asociado. Para obtener más información, consulte Crear una cola de trabajos.
- Están AWS CLI instalados en su sistema local. Para obtener más información, consulte Installing the 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.

Tutorial: Construye 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.
```

Tutorial: Envía tu imagen a Amazon ECR

Ahora que ha compilado y probado el contenedor de Docker, debe enviarlo a un repositorio de imágenes. En este ejemplo se usa AmazonECR, pero puedes usar otro registro, como DockerHub.

1. Crea un repositorio de ECR imágenes de Amazon para almacenar la imagen de tu 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. Etiqueta tu `print-color` imagen con el ECR repositorio de Amazon URI que devolviste en el paso anterior.

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

3. Inicia sesión en tu ECR registro de Amazon. 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. Envía tu imagen a AmazonECR.

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

Tutorial: 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 Crear 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. Sustituya el repositorio URI de imágenes por el suyo propio. URI

```
{
  "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 de trabajo con AWS Batch.

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

Tutorial: 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`.

Tutorial: consulte los registros de trabajos de su matriz

Una vez que 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.

Ejecute GPU trabajos

GPUlos trabajos le ayudan a ejecutar trabajos que utilizan una instanciaGPUs.

Se admiten los siguientes tipos de instancias EC2 GPU basadas en Amazon. Para obtener más información, consulte Instancias Amazon EC2 G3, InstanciasAmazon EC2 G4, InstanciasAmazon EC2 G5, Instancias Amazon EC2 P2, Instancias Amazon EC2 P3,Instancias AmazonP4d e Instancias EC2 Amazon P5. EC2

Tipo de instancia	GPUs	GPUmemoria	vCPUs	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
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

Tipo de instancia	GPUs	GPUmemoria	vCPUs	Memoria	Ancho de banda de red
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
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 se admiten los tipos de instancias que admiten NVIDIA GPU y utilizan una arquitectura x86_64 para GPU los trabajos en. AWS Batch Por ejemplo, las familias de instancias de G4ad y G5g no son compatibles.

El resourceRequirementsparámetro de la definición del trabajo especifica el número de tareas que se van GPUs a fijar al contenedor. Este número de no GPUs está disponible para ningún otro trabajo que se ejecute en esa instancia durante ese trabajo. Todos los tipos de instancias de un entorno de cómputo que ejecutan GPU trabajos deben pertenecer a las familias de instancias `p2` `p3``p4`,`p5`,,`g3`, `g3s``g4`, o de las familias de `g5` instancias. Si esto no se hace, es posible que un GPU trabajo quede atrapado en el `RUNNABLE` estado.

Los trabajos que no utilizan el se GPUs pueden ejecutar en GPU instancias. Sin embargo, es posible que su ejecución en las GPU instancias cueste más que en otras GPU instancias similares. En función de la v específicaCPU, la memoria y el tiempo necesarios, es posible que estas GPU tareas no ejecutadas impidan la ejecución de las GPU tareas.

Temas

- Tutorial: Crear una GPU basada Kubernetes clúster en Amazon EKS
- Tutorial: Crear una definición de EKS GPU trabajo en Amazon
- Tutorial: Ejecuta un GPU trabajo en tu EKS clúster de Amazon

Tutorial: Crear una GPU basada Kubernetes clúster en Amazon EKS

Antes de crear una GPU basada Kubernetes clúster en AmazonEKS, debes haber completado los pasos indicadosTutorial: Cómo empezar AWS Batch en Amazon EKS. Además, tenga en cuenta lo siguiente:

- AWS Batch admite tipos de instancias con NVIDIAGPUs.
- De forma predeterminada, AWS Batch selecciona Amazon EKS accelerated AMI con el Kubernetes versión que coincida con tu versión del plano de control del EKS clúster de Amazon.

```
$ 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 gestiona el NVIDIA GPUcomplemento de dispositivo en tu nombre. Debes instalar este complemento en tu EKS clúster de Amazon y permitir que se dirija a los AWS Batch nodos. Para obtener más información, consulte Habilitar GPU Support en Kubernetesactivado GitHub.

Para configurar el NVIDIA device plugin (`DaemonSet`) para dirigirse 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 (CPUy de memoria) con cargas de trabajo GPU basadas en la misma combinación de entorno informático y cola de tareas. Esto se debe a que las tareas informáticas pueden consumir capacidad. 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
```

Tutorial: Crear una definición de EKS GPU trabajo en Amazon

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

Para registrar una definición de GPU trabajo para AmazonEKS, ejecuta 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
```

Tutorial: Ejecuta un GPU trabajo en tu EKS clúster de Amazon

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

Para enviar un GPU trabajo, 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"
}
```

Busque AWS Batch trabajos en una cola de trabajos

Puede buscar y filtrar sus trabajos AWS Batch mediante la búsqueda de empleo. Esta función ofrece una opción para buscar en una cola de trabajos preexistente y filtrar sus trabajos por una de las

cinco opciones (Creado después, Creado antes, Nombre del trabajo, Definición del trabajo y Estado). Para que el proceso utilice esta función, complete el Tutorial: Busque y filtre AWS Batch trabajos procedimiento.

Para buscar con varios criterios simultáneamente, utilice la función de búsqueda de empleo Beta. Por ejemplo, puede incluir uno o todos los filtros siguientes: Estado del trabajo, Intervalo de fechas y Criterios adicionales (como el nombre del trabajo, la definición del trabajo o el identificador del trabajo). Para conocer el proceso de uso de esta función, consulte el Tutorial: (Beta) Busque y filtre AWS Batch trabajos procedimiento.

Temas

- Tutorial: Busque y filtre AWS Batch trabajos
- Tutorial: (Beta) Busque y filtre AWS Batch trabajos

Tutorial: Busque y filtre AWS Batch trabajos

Utilice este procedimiento para enumerar todos los trabajos de una cola de trabajos en la AWS Batch consola. Si lo desea, utilice la lista desplegable de tipos de filtro para restringir los resultados en función de los criterios que especifique.

1. Vaya a la consola de AWS Batch.
2. En el panel de navegación, elija Trabajos.
3. Amplíe la lista desplegable de colas de trabajos y elija la cola de trabajos en la que desee buscar.

   > **Note**
   >
   > Puede buscar trabajos en una sola cola de trabajos a la vez.

4. (Opcional) Elija el tipo de filtro que desee aplicar.

   - Creado después: devuelve los trabajos creados después de una fecha y hora específicas (en pulgadasUTC). Si selecciona esta opción, aparecen dos campos de valores de filtro en los que puede especificar la fecha y hora más tempranas en las que se devolverá la búsqueda.
   - Creado antes: devuelve los trabajos creados antes de una fecha y hora específicas (en pulgadasUTC). Si selecciona esta opción, aparecen dos campos de valores de filtro en los que puede especificar la última fecha y hora en que se mostrará la búsqueda.

- Definición de trabajo: devuelve los trabajos creados con un nombre de definición de trabajo específico. Si selecciona esta opción, aparece un campo en el que puede especificar la definición completa o parcial de un trabajo (si añade un asterisco (*) al final, el filtro coincidirá con cualquier definición de trabajo que comience por la cadena anterior al asterisco).
- Nombre del trabajo: devuelve los trabajos que tienen un nombre de trabajo específico. Si selecciona esta opción, aparece un campo en el que puede especificar el nombre completo o parcial de un trabajo (si añade un asterisco (*) al final, el filtro coincidirá con cualquier nombre de trabajo que comience por la cadena situada antes del asterisco).
- Estado: devuelve los trabajos que tienen el estado del trabajo seleccionado. Si selecciona esta opción, aparecerá una lista desplegable para que la amplíe y seleccione un estado específico. Para obtener más información, consulte Estados de trabajo.

5. (Opcional) Seleccione Cargar todos los trabajos para ver todos los trabajos de una cola de trabajos de los últimos siete días y sin los filtros anteriores. Si elige esta opción, aparecerá el siguiente mensaje:

> **Note**
>
> AWS Batch ahora almacena siete (7) días del historial de trabajos. Si decide cargar todos los trabajos, aumentará el tiempo necesario para acceder a sus trabajos y empezar a buscarlos y filtrarlos. En su lugar, el uso de un tipo de filtro proporcionará resultados más inmediatos. ¿Está seguro de que desea cargar todos los trabajos?

R Seleccione Confirmar para cerrar el mensaje.

- Si elige esta opción, aparece una lista desplegable en la que puede elegir opciones de filtro similares: Estado, Nombre del trabajo e ID del trabajo. Si elige una de estas opciones, puede filtrar aún más los criterios de búsqueda cuando el filtro especificado sea igual, no sea igual, contenga o no contenga los criterios que introduzca.

6. Cuando las opciones estén completas, elija Buscar.

Tutorial: (Beta) Busque y filtre AWS Batch trabajos

Utilice este procedimiento para enumerar todos los trabajos de una cola de trabajos en la AWS Batch consola y utilizar las funciones de búsqueda beta. Si lo desea, utilice la sección de opciones de búsqueda para restringir los resultados en función de los criterios que especifique. Esta función

de búsqueda beta le permite combinar los criterios de búsqueda para limitar los resultados. Sin embargo, solo puede definir un criterio por filtro.

1. Vaya a la consola de AWS Batch.
2. En el panel de navegación, elija Trabajos.
3. Asegúrese de que la opción Buscar trabajos: versión beta esté activa.
4. Amplíe la lista desplegable de colas de trabajos y elija la cola de trabajos en la que desee buscar.

> **Note**
>
> Puede buscar trabajos en una sola cola de trabajos a la vez.

5. (Opcional) Amplíe la lista desplegable Estado del trabajo y seleccione uno o más estados del trabajo. Para obtener más información, consulte Estados de trabajo.
6. (Opcional) Elija un rango de fechas para filtrar los resultados en función de un rango de fechas y horas.
   - Seleccione el modo relativo para buscar trabajos que se hayan ejecutado dentro de un intervalo de tiempo que cuente hacia atrás desde la fecha y la hora actuales.
   - Elija el modo Absoluto para buscar los trabajos que se ejecutaron dentro del intervalo de fechas y horas que especifique.
7. (Opcional) En el campo Criterios adicionales, introduzca las palabras clave que desee incluir en los resultados de la búsqueda. Por ejemplo, puede usar este campo para buscar por nombre de trabajo, definición de trabajo o ID de trabajo. Según la propiedad, es posible que deba definir operadores adicionales, como iguales a (=) o que comiencen por (^).
8. Cuando las opciones estén completas, elija Buscar.

> **Tip**
>
> Puede elegir el icono de configuración para personalizar las columnas que aparecen y su orden en la tabla de resultados de la búsqueda.

Tutorial: envíe los registros de AWS Batch trabajos a CloudWatch Logs

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

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.

Tutorial: Revise la información del AWS Batch 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 Batch.
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 Buscar y filtrar para encontrar un trabajo más rápido. Para obtener más información, consulte Busque AWS Batch trabajos en una cola de trabajos.

4. Elija el trabajo que desee.

> **Note**
>
> También puedes 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.

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 Batch IAMpolíticas, funciones y permisos
- Rol de ejecución de AWS Batch IAM
- Creación de una nube virtual privada
- Utilice un punto final de interfaz para acceder AWS Batch
- Validación de conformidad para AWS Batch
- Seguridad de la infraestructura en AWS Batch
- Prevención de la sustitución confusa entre servicios
- Registrar AWS Batch API llamadas con AWS CloudTrail

- Solucionar problemas AWS Batch IAM

Identity and Access Management para AWS Batch

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. IAMlos administradores controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar AWS Batch los recursos. IAMes una Servicio de AWS que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración de acceso mediante políticas
- ¿Cómo AWS Batch funciona con IAM

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 Solucionar problemas AWS Batch IAM.

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. A continuación, debe enviar solicitudes a su IAM administrador para cambiar los permisos de los usuarios del servicio. Revise la información de esta página para comprender los conceptos básicos deIAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con AWS Batch, consulte¿Cómo AWS Batch funciona con IAM.

IAMadministrador: si es IAM administrador, puede que desee obtener más información sobre cómo puede redactar políticas para administrar el acceso a ellas AWS Batch. Para ver ejemplos de

políticas AWS Batch basadas en la identidad que puede utilizarIAM, 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 IAM usuario o asumiendo un IAM rol. 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 (IAMIdentity 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, el administrador configuró previamente la federación de identidades mediante roles. IAM Cuando accede AWS mediante la federación, asume indirectamente un rol.

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 incluye 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 AWS API las solicitudes en la Guía del IAM usuario.

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 multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactorial en la Guía del AWS IAM Identity Center usuario y Uso de la autenticación multifactorial (MFA) AWS en la Guía del IAM usuario.

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 tareas que requieren

que inicie sesión como usuario root, consulte Tareas que requieren credenciales de usuario root en la Guía del IAM usuario.

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 IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones Cuentas de AWS . Para obtener información sobre IAM Identity Center, consulte ¿Qué es IAM Identity Center? en la Guía AWS IAM Identity Center del usuario.

Usuarios y grupos de IAM

Un IAMusuario es una identidad dentro de ti Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte Rotar las claves de acceso con regularidad para los casos de uso que requieran credenciales de larga duración en la Guía del IAM usuario.

Un IAMgrupo es una identidad que especifica un conjunto de IAM usuarios. 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 para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar IAM los recursos.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales

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

IAMroles

Un IAMrol es una identidad dentro de ti Cuenta de AWS que tiene permisos específicos. Es similar a un IAM usuario, pero no está asociado a una persona específica. Puede asumir temporalmente un IAM rol en el AWS Management Console cambiando de rol. Puede asumir un rol llamando a una AWS API operación AWS CLI o o utilizando una operación personalizadaURL. Para obtener más información sobre los métodos de uso de roles, consulte Métodos para asumir un rol en la Guía del IAM usuario.

IAMlos roles 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 sobre los roles para la federación, consulte Creación de un rol para un proveedor de identidad externo en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en. IAM Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos IAM de usuario temporales: un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- Acceso multicuenta: puedes usar un IAM rol para permitir que alguien (un responsable de confianza) de una cuenta diferente acceda a los recursos de tu cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los recursos entre cuentas IAM en la Guía del IAM usuario.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros. Servicios de AWS Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un IAM usuario o un rol para realizar acciones en AWS ellas, se te considera director. Cuando utiliza algunos servicios, es posible que realice

una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del principal que llama a an Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes 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 detalles sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar sesiones de acceso.

- Función de servicio: una función de servicio es una IAMfunción que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos Servicio de AWS en un rol en el IAMManual del usuario.
- Función vinculada a un 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 IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y que realizan AWS CLI o AWS API solicitan. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Uso de un IAM rol para conceder permisos a aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del IAM usuario.

Para saber si se deben usar IAM roles o IAM usuarios, consulte Cuándo crear un IAM rol (en lugar de un usuario) en la Guía del IAM usuario.

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 JSON documentos. Para obtener más información sobre

la estructura y el contenido de los documentos de JSON políticas, consulte Descripción general de JSON las políticas en la Guía del IAM usuario.

Los administradores pueden usar AWS JSON las políticas 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 a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

IAMlas políticas definen los 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 AWS Management Console AWS CLI, el o el AWS API.

Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, 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 información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario. 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 funciones de su empresa. Cuenta de AWS Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para saber cómo elegir entre una política gestionada o una política integrada, consulte Elegir entre políticas gestionadas y políticas integradas en la Guía del IAM usuario.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles 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 puede usar políticas AWS administradas desde una política IAM basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3 AWS WAF y Amazon VPC son ejemplos de servicios compatiblesACLs. Para obtener más informaciónACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAMusuario o rol). 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 los límites de los permisos para IAM las entidades en la Guía del IAMusuario.
- Políticas de control de servicios (SCPs): SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU) en AWS Organizations. AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. SCPLimita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre OrganizationsSCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.

- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte las políticas de sesión en la Guía del IAM usuario.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del IAM usuario.

¿Cómo AWS Batch funciona con IAM

¿Cómo AWS Batch funciona con IAM

Antes de administrar el IAM acceso a AWS Batch, infórmese sobre IAM las funciones disponibles para su uso AWS Batch.

IAMfunciones que puedes usar con AWS Batch

IAMcaracterística	AWS Batch apoyo
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACLs	No
ABAC(etiquetas en las políticas)	Sí
Credenciales temporales	Sí

IAMcaracterística	AWS Batch apoyo
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 IAM funciones, consulte AWS los servicios con los que funcionan IAM en la Guía del IAM usuario.

Políticas basadas en la identidad para AWS Batch

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, 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 información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario. IAM

Con las políticas IAM basadas en la identidad, 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 obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la referencia sobre los elementos de la IAM JSON política en la Guía del IAMusuario.

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

Compatibilidad con las acciones de política: sí

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

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de política suelen tener el mismo nombre que la AWS API operación asociada. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API 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

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar AWS JSON políticas 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` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica

recomendada, especifique un recurso mediante su 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 respectivos tiposARNs, consulte Recursos definidos por AWS Batch en la Referencia de autorización del servicio. Para saber con qué acciones puede especificar cada recurso, consulte Acciones definidas por AWS Batch. ARN

Claves de condición de políticas para AWS Batch

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar AWS JSON políticas 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 a un IAM usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de IAM usuario. Para obtener más información, consulte los elementos IAM de la política: variables y etiquetas en la Guía del IAM usuario.

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 IAMusuario.

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.

Control de acceso basado en atributos () ABAC con AWS Batch

Soportes ABAC (etiquetas en las políticas): Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchos AWS recursos. Etiquetar entidades y recursos es el primer paso deABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABACes útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve 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 al respectoABAC, consulte ¿Qué es? ABAC en la Guía IAM del usuario. Para ver un tutorial con los pasos de configuraciónABAC, consulte Usar el control de acceso basado en atributos (ABAC) en la Guía del IAMusuario.

Utilice credenciales temporales con AWS Batch

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta la sección Servicios de AWS Cómo trabajar con credenciales temporales IAM en la Guía del IAM usuario.

Está utilizando credenciales temporales si inicia sesión AWS Management Console con cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea

automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de rol, consulte Cambiar a un rol (consola) en la Guía del IAMusuario.

Puede crear credenciales temporales manualmente con la tecla AWS CLI o AWS API. A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios de AWS Batch

Admite sesiones de acceso directo (FAS): Sí

Cuando utilizas un IAM usuario o un rol 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. FASutiliza los permisos del principal que llama a una Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes 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 detalles sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar sesiones de acceso.

Funciones de servicio para AWS Batch

Compatibilidad con roles de servicio: sí

Una función de servicio es una IAMfunción que asume un servicio para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos Servicio de AWS en un rol en el IAMManual del usuario.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría afectar a AWS Batch la funcionalidad. Edite los roles de servicio sólo cuando AWS Batch proporcione orientación para hacerlo.

Funciones vinculadas al servicio para AWS Batch

Admite 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 IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.

Para obtener más información sobre la creación o la administración de funciones vinculadas a un servicio, consulte los AWS servicios 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.

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 con AWS Management Console, AWS Command Line Interface (AWS CLI) o. AWS API Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte Creación de IAM políticas en la Guía del IAMusuario.

Para obtener más información sobre las acciones y los tipos de recursos definidos AWS Batch, incluido el formato de cada uno de los tipos de recursos, consulte las claves de condición, recursos y acciones de la Referencia AWS Batch de autorización de servicios. ARNs

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la AWS Batch consola
- 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. Para obtener más información, consulte las políticas AWS gestionadas o las políticas AWS gestionadas para las funciones laborales en la Guía del IAM usuario.
- Aplique permisos con privilegios mínimos: cuando establezca permisos con IAM políticas, 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. Para obtener más información sobre cómo IAM aplicar permisos, consulte Políticas y permisos IAM en la IAM Guía del usuario.
- Utilice las condiciones en IAM las políticas para restringir aún más el acceso: puede añadir una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse medianteSSL. 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, como AWS CloudFormation. Para obtener más información, consulte los elementos IAM JSON de la política: Condición en la Guía del IAM usuario.
- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAMAccess Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarle a crear políticas seguras y funcionales. Para obtener más información, consulte la validación de políticas de IAM Access Analyzer en la Guía del IAM usuario.
- Requerir autenticación multifactorial (MFA): si se encuentra en una situación en la que se requieren IAM usuarios o un usuario raíz Cuenta de AWS, actívela MFA para aumentar la seguridad. Para solicitarlo MFA cuando se convoque a API las operaciones, añada MFA condiciones a sus políticas. Para obtener más información, consulte Configuración del API acceso MFA protegido en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadasIAM, consulte las prácticas recomendadas de seguridad IAM en la Guía del IAM usuario.

Uso de la AWS Batch consola

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 realicen llamadas únicamente al AWS CLI o al AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la AWS Batch consola, adjunte también la política `ReadOnly` AWS gestionada AWS Batch `ConsoleAccess` o la política gestionada a las entidades. Para obtener más información, consulte Añadir permisos a un usuario en la Guía del IAM usuario.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo se muestra cómo se puede crear una política que permita a IAM los usuarios ver las políticas integradas y administradas asociadas a su identidad de usuario. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la tecla o. AWS CLI AWS API

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS políticas administradas para AWS 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 administradas por el IAM 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 IAM usuario.

AWS los servicios mantienen y AWS actualizan las políticas administradas 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 una

descripción de las políticas de funciones laborales, consulte las políticas AWS administradas para las funciones laborales en la Guía del IAMusuario.

AWS política gestionada: BatchServiceRolePolicy

El BatchServiceRolePolicyrol `AWSServiceRoleForBatch`vinculado al servicio utiliza la IAM política gestionada. Esto le permite AWS Batch realizar acciones en su nombre. No puedes adjuntar esta política a tus IAM entidades. Para obtener más información, consulte Utilice funciones vinculadas al servicio 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 Amazon EC2 Auto Scaling. AWS Batch crea y administra grupos de Amazon EC2 Auto Scaling para la mayoría de los entornos de cómputo.
- `ec2`— Permite AWS Batch controlar el ciclo de vida de las EC2 instancias de Amazon, 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 EC2 puntual.
- `ecs`- Permite AWS Batch crear y gestionar ECS clústeres de Amazon, definiciones de tareas y tareas para la ejecución de trabajos.
- `eks`- Permite AWS Batch describir el recurso del EKS clúster de Amazon para las validaciones.
- `iam`- Permite AWS Batch validar y transferir los roles proporcionados por el propietario a AmazonEC2, Amazon EC2 Auto Scaling y AmazonECS.
- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch 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 IAM política AWSBatchServiceRoleadministrada 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 Amazon EC2 Auto Scaling. AWS Batch crea y administra grupos de Amazon EC2 Auto Scaling para la mayoría de los entornos de cómputo.
- `ec2`— Permite AWS Batch gestionar el ciclo de vida de las EC2 instancias de Amazon, 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 EC2 puntual.
- `ecs`- Permite AWS Batch crear y gestionar ECS clústeres de Amazon, definiciones de tareas y tareas para la ejecución de trabajos.
- `iam`- Permite AWS Batch validar y transferir los roles proporcionados por el propietario a AmazonEC2, Amazon EC2 Auto Scaling y AmazonECS.
- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch 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 acciones para describir y enumerar acciones para Amazon EC2ECS, Amazon EKS CloudWatch, Amazon y sus IAM servicios. Esto sirve para que IAM las identidades, ya sean usuarios o roles, puedan ver los recursos AWS Batch gestionados que se crearon en su nombre. Por último, esta política también permite IAM transferir funciones seleccionadas a esos servicios.

Puede adjuntarlos AWSBatchFullAccessa sus IAM entidades. 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 al RSS feed 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 una declaración IDs, otorgar AWS Batch permisos `ec2:DescribeSpotFleetRequestHistory` y `autoscaling:DescribeScalingActivities` .	5 de diciembre de 2023
BatchServiceRolePolicypolítica actualizada	Se ha actualizado para añadir compatibilidad con la descripción de EKS los clústeres de Amazon.	20 de octubre de 2022
AWSBatchFullAccesspolítica actualizada	Se ha actualizado para añadir compatibilidad con la publicación y descripción de EKS los clústeres de Amazon.	20 de octubre de 2022
BatchServiceRolePolicypolítica actualizada	Se actualizó para añadir compatibilidad con los grupos EC2 de reservas de Amazon Capacity gestionados por AWS Resource Groups. Para obtener más	18 de mayo de 2022

Cambio	Descripción	Fecha
	información, consulta Cómo trabajar con grupos de reserva de capacidad en la Guía EC2 del usuario de Amazon.	
BatchServiceRolePolicyy se actualizaron AWSBatchServiceRolelas políticas	Se ha actualizado para añadir compatibilidad con la descripción del estado de las instancias AWS Batch gestionadas en Amazon, de EC2 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 capacidadGPU, Elastic y Elastic Inference en Amazon. EC2	26 de marzo de 2021
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 IAM permisos 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

AWS Batch IAMpolíticas, funciones y permisos

De forma predeterminada, los usuarios no tienen permiso para crear o modificar AWS Batch recursos ni para realizar tareas mediante la AWS Batch API AWS Batch consola o el AWS CLI. Para permitir

que los usuarios realicen estas acciones, cree IAM políticas que concedan a los usuarios permisos para los recursos y API operaciones específicos. 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 IAM usuario. Para obtener más información sobre la administración y la creación de IAM políticas personalizadas, consulte Administración de IAM políticas.

AWS Batch hace llamadas a otras personas 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 un IAM rol y una política que proporcionen estos permisos. A continuación, asocia el rol a sus entornos de computación al crearlos. Para obtener más información, consulte IAMFuncionesFunción de ECS instancia de Amazon, Uso de funciones vinculadas a servicios y Creación de una función para delegar permisos a un AWS servicio en la Guía del IAM usuario.

Temas

- IAMestructura de políticas
- Recurso: ejemplos de políticas para AWS Batch
- Recurso: política AWS Batch gestionada

IAMestructura de políticas

En los temas siguientes se explica la estructura de una IAM política.

Temas

- Sintaxis de la política
- APIacciones para AWS Batch
- Nombres de recursos de Amazon para AWS Batch
- Confirme que los usuarios tengan los permisos necesarios

Sintaxis de la política

Una IAM política es un JSON documento que consta de una o más declaraciones. Cada instrucción tiene la estructura siguiente.

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
    "key":"value"
    }
      }
    }
  ]
}
```

Hay cuatro elementos principales que componen una declaración:

- Effect: el valor de effect puede ser `Allow` o `Deny`. De forma predeterminada, los usuarios no tienen permiso para usar recursos y API acciones. 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 API acción específica para la que concedes o deniegas el permiso. Para obtener instrucciones acerca de cómo especificar la acción, consulte APIacciones para AWS Batch.
- Resource: el recurso al que afecta la acción. Con algunas AWS Batch API acciones, puedes incluir recursos específicos en tu política que la acción puede crear o modificar. Para especificar un recurso en la declaración, utilice su nombre de recurso de Amazon (ARN). Para obtener más información, consulte Se admiten permisos a nivel de recursos para las acciones AWS Batch API y Nombres de recursos de Amazon para AWS Batch. Si la AWS Batch API operación actualmente no admite permisos a nivel de recursos, 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 IAM de política para AWS Batch, consulte. Recurso: ejemplos de políticas para AWS Batch

APIacciones para AWS Batch

En una declaración de IAM política, puede especificar cualquier API acción de cualquier servicio compatibleIAM. Para AWS Batch ello, utilice el siguiente prefijo con el nombre de la API acción: `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 AWS Batch API acciones, incluya un comodín (*).

```
"Action": "batch:*"
```

Para obtener una lista de AWS Batch acciones, consulte Acciones en la AWS Batch APIreferencia.

Nombres de recursos de Amazon para AWS Batch

Cada declaración IAM de política se aplica a los recursos que especifique mediante sus nombres de recursos de Amazon (ARNs).

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

El Región de AWS para el recurso (por ejemplo,`us-east-2`).

account

El Cuenta de AWS ID, 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.

AWS Batch APIActualmente, las operaciones admiten permisos a nivel de recursos en varias operaciones. API Para obtener más información, consulte Se admiten permisos a nivel de recursos para las acciones AWS Batch API. Para especificar todos los recursos, o si una API acción específica no es compatibleARNs, incluya un comodín (*) en el elemento. `Resource`

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

Confirme que los usuarios tengan los permisos necesarios

Antes de poner en marcha una IAM política, asegúrese de que conceda a los usuarios los permisos para usar las API acciones y los recursos específicos que necesitan.

Para ello, cree primero un usuario con fines de prueba y adjunte la IAM política al usuario de prueba. 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 sus políticas mediante el simulador IAM de políticas. Para obtener más información sobre el simulador de políticas, consulte Uso del simulador IAM de políticas en la Guía del IAM usuario.

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 DecodeAuthorizationMessagela AWS Security Token Service APIReferencia y decode-authorization-messagela Referencia de AWS CLI comandos.

Recurso: ejemplos de políticas para AWS Batch

Puedes crear IAM políticas específicas para restringir las llamadas y los recursos a los que tienen acceso los usuarios de tu 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, consulta los permisos y las políticas en la Guía del IAM usuario. Para obtener instrucciones sobre cómo administrar y crear IAM políticas personalizadas, consulte Administración de IAM políticas.

Los siguientes ejemplos muestran instrucciones de política que puede utilizar para controlar los permisos que los usuarios de tienen en AWS Batch.

Ejemplos

- Recurso: acceso de solo lectura para AWS Batch
- Recurso: se limita al POSIX usuario, la imagen de Docker, el nivel de privilegios y la función al enviar el trabajo
- Recurso: Limítese al prefijo de la definición del trabajo al enviar el trabajo
- Recurso: Limítese a una cola de trabajos
- Denegue la acción cuando todas las condiciones coincidan con las cadenas
- Recurso: deniegue la acción cuando alguna clave de condición coincida con una cadena
- Recurso: utilice la clave de batch:ShareIdentifier condición

Recurso: acceso de solo lectura para AWS Batch

La siguiente política otorga a los usuarios permisos para usar todas AWS Batch API las acciones cuyo nombre comience por y. `Describe` `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 API las acciones.

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

Recurso: se limita al POSIX usuario, la imagen de Docker, el nivel de privilegios y la función al enviar el trabajo

La siguiente política permite a los POSIX usuarios administrar su propio conjunto de definiciones de trabajos restringidos.

Utilice las instrucciones primera y segunda para registrar y anular el registro de cualquier nombre de definición de trabajo cuyo nombre lleve el prefijo *`JobDefA_`*.

La primera sentencia también utiliza claves contextuales condicionales para restringir el POSIX usuario, el estado privilegiado y los valores de imagen del contenedor dentro `containerProperties` de la definición de un trabajo. Para obtener más información, consulte RegisterJobDefinitionla AWS Batch APIReferencia. En este ejemplo, las definiciones de trabajo solo se pueden registrar cuando el POSIX usuario está configurado en`nobody`. El indicador privilegiado está establecido en `false`. Por último, la imagen está configurada `myImage` en un ECR repositorio de Amazon.

> **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 mediante el uso de `uid` valores directos tanto en la definición del trabajo como en IAM las políticas asociadas. Tanto las teclas AWS Batch API de operaciones como las `batch:User` IAM condicionales 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"
            ]
```

```
        }
    ]
}
```

Recurso: Limítese al prefijo de la definición del trabajo al enviar el trabajo

Utilice la 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/*"
      ]
    }
  ]
}
```

Recurso: Limítese a una 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"
      ]
    }
  ]
}
```

Denegue la acción cuando todas las condiciones coincidan con las cadenas

La siguiente política deniega el acceso al RegisterJobDefinitionAPIoperación cuando la clave de condición `batch:Image` (ID de imagen del contenedor) es»*`string1`*"y la clave de condición `batch:LogDriver` (controlador del registro del contenedor) es»*`string2`*». AWS Batch evalúa las claves de estado 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 la lista de claves de condición AWS Batch, consulte las claves de condición 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"
        }
      }
    }
  ]
}
```

Recurso: deniegue la acción cuando alguna clave de condición coincida con una cadena

La siguiente política deniega el acceso al RegisterJobDefinitionAPIoperación cuando la clave de condición `batch:Image` (ID de imagen del contenedor) es»*`string1`*«o la clave de condición `batch:LogDriver` (controlador del 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 la lista de claves de condición AWS Batch, consulte Claves de condición AWS Batch en la Referencia de autorización de servicios. 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"
          ]
        }
      }
    }
  ]
}
```

Recurso: utilice la clave de **`batch:ShareIdentifier`** condición

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"
          ]
        }
      }
    }
  ]
}
```

Recurso: política AWS Batch gestionada

AWS Batch proporciona una política gestionada que puede adjuntar a los usuarios. Esta política otorga permiso para usar AWS Batch recursos y API operaciones. Puede aplicar esta política directamente o utilizarla como punto de partida para crear sus propias políticas. Para obtener más información sobre cada API operación mencionada en estas políticas, consulte Acciones en la AWS Batch APIreferencia.

AWSBatchFullAccess

Esta política permite al administrador el acceso total a AWS Batch.

```
{
  "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"
          }
      }
    }
  ]
}
```

Rol de ejecución de AWS Batch IAM

La función de ejecución otorga al ECS contenedor y a AWS Fargate los agentes de Amazon permiso para realizar AWS API llamadas en tu nombre.

> **Note**
>
> La función de ejecución es compatible con el agente ECS contenedor de Amazon, versión 1.16.0 y versiones posteriores.

El rol IAM de ejecución es obligatorio 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 información sobre el rol de ECS instancia de Amazon, consulteFunción de ECS instancia de Amazon. Para obtener más información acerca de los roles de servicio, consulte ¿Cómo AWS Batch funciona con IAM.

Amazon ECS proporciona la política `AmazonECSTaskExecutionRolePolicy` gestionada. 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": "*"
    }
  ]
}
```

Se admiten permisos a nivel de recursos para las acciones AWS Batch API

El término permisos a nivel de recursos se refiere a la capacidad de especificar los recursos en los que los usuarios pueden realizar acciones. AWS Batch admite parcialmente los permisos a nivel de recursos. En el caso de algunas AWS Batch acciones, puede controlar cuándo se permite a los usuarios utilizarlas en función de las condiciones que deben cumplirse. 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.

En la siguiente lista se describen las AWS Batch API acciones que actualmente admiten permisos a nivel de recurso. En la lista también se describen los recursos, los recursos y las claves de condición compatibles para cada acción. ARNs

> **Important**
>
> Si una AWS Batch API acción no aparece en esta lista, significa que no admite permisos a nivel de recurso. Si una AWS Batch API acción no admite permisos a nivel de recursos, puedes conceder permiso a los usuarios para que la usen. 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, GetJobQueueSnapshot, ListTagsForResource, RegisterJobDefinition, SubmitJob, TagResource, TerminateJob, UntagResource, UpdateComputeEnvironment, UpdateSchedulingPolicy, UpdateJobQueue

CancelJob

Cancela un trabajo de una cola. AWS Batch

Resource

Trabajo

arn:aws:batch:*region*:*account*:trabajo/*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 AWS Batch informático.

Resource

Entorno de computación

arn:aws:batch:*region*:*account*:entorno de computo/*compute-environment-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.

CreateJobQueue

Crea una cola de trabajos. AWS Batch

Resource

Entorno de computación

arn:aws:batch:*region*:*account*:entorno de computo/*compute-environment-name*

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*:cola de trabajos/*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*: política de planificación/*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. AWS Batch

Resource

Entorno de computación

arn:aws:batch:*region*:*account*:entorno de computo/*compute-environment-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

CreateSchedulingPolicy

Crea una política de programación. AWS Batch

Resource

Política de programación

arn:aws:batch:*region*:*account*: política de planificación/*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.

Resource

Cola de trabajos

arn:aws:batch:*region*:*account*:cola de trabajos/*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.

Resource

Política de programación

arn:aws:batch:*region*:*account*: política de planificación/*scheduling-policy-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

DeregisterJobDefinition

Anula el registro de una definición de trabajo. AWS Batch

Resource

Definición de trabajo

arn:aws:batch:*region*:*account*:definición de trabajo/*definition-name*:*revision*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

GetJobQueueSnapshot

Proporciona una lista de los primeros 100 `RUNNABLE` trabajos asociados a una sola cola de trabajos.

Resource

Cola de trabajos

arn:aws:batch:*region*:*account*:cola de trabajos/*queue-name*

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.

Resource

Entorno de computación

arn:aws:batch:*region*:*account*:entorno de computo/*compute-environment-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Trabajo

arn:aws:batch:*region*:*account*:trabajo/*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*:definición de trabajo/*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*:cola de trabajos/*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*: política de planificación/*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

Resource

Definición de trabajo

arn:aws:batch:*region*:*account*:definición de trabajo/*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 registro `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 a partir de una definición de trabajo AWS Batch .

Resource

Trabajo

arn:aws:batch:*region*:*account*:trabajo/*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*:definición de trabajo/*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 usar cuando la definición de trabajo Amazon Resource Name (ARN) tiene el formato `arn:aws:batch:region:account_number:job-definition/definition-name:revision`.

Cola de trabajos

arn:aws:batch:*region*:*account*:cola de trabajos/*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.

Resource

Entorno de computación

arn:aws:batch:*region*:*account*:entorno de computo/*compute-environment-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Trabajo

arn:aws:batch:*region*:*account*:trabajo/*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*:definición de trabajo/*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*:cola de trabajos/*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*: política de planificación/*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 trabajos. AWS Batch

Resource

Trabajo

arn:aws:batch:*region*:*account*:trabajo/*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.

Resource

Entorno de computación

arn:aws:batch:*region*:*account*:entorno de computo/*compute-environment-name*

Claves de condición

`aws:ResourceTag/${TagKey}` (Cadena)

Filtra acciones en función de las etiquetas que están asociadas con el recurso.

Trabajo

arn:aws:batch:*region*:*account*:trabajo/*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*:definición de trabajo/*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*:cola de trabajos/*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*: política de planificación/*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

Actualiza un entorno informático. AWS Batch

Resource

Entorno de computación

arn:aws:batch:*region*:*account*:entorno de computo/*compute-environment-name*

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.

Resource

Cola de trabajos

arn:aws:batch:*region*:*account*:cola de trabajos/*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*: política de planificación/*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.

Resource

Política de programación

arn:aws:batch:*region*:*account*: política de planificación/*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 AWS Batch API

AWS Batch define las siguientes claves de condición que se utilizan en el `Condition` elemento de una política. 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 las claves de condición globales disponibles en la Guía del IAM usuario.

`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)

El Región de AWS lugar al que se envían los registros.

`batch:AWSLogsStreamPrefix` (Cadena)

Prefijo de flujo de registro `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 `shareIdentifier` parámetro al que se envían 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.

Tutorial: Crea el rol IAM de ejecución

Si su cuenta aún no tiene un rol de IAM ejecución, siga los siguientes pasos para crear el rol.

1. Abre la IAM consola 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. Para el servicio o el caso de uso, elija EC2. A continuación, EC2vuelva a elegir.
6. Elija Next (Siguiente).
7. Para las políticas de permisos, busque mazonECSTaskExecutionRolePolicyA.
8. Seleccione la casilla de verificación situada a la izquierda de la mazonECSTask ExecutionRolePolicy política A y, a continuación, seleccione Siguiente.
9. En Nombre de rol, escriba `ecsTaskExecutionRole` y luego elija Crear rol.

Tutorial: Compruebe la función IAM de ejecución

Utilice el siguiente procedimiento para comprobar que su cuenta ya tiene la función de IAM ejecución y adjunte la IAM política gestionada, si es necesario.

1. Abra la IAM consola 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 Tutorial: Crea el rol IAM de ejecución. Si el rol existe, selecciónelo para ver sus políticas asociadas.
4. En la pestaña Permisos, compruebe que la política mazonECSTask ExecutionRolePolicy administrada A 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 mazonECSTaskExecutionRolePolicyA.

c. Marque la casilla situada a la izquierda de la mazonECSTask ExecutionRolePolicy política A y elija Adjuntar 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"
    }
  ]
}
```

Utilice funciones vinculadas al servicio para AWS Batch

AWS Batch usa AWS Identity and Access Management (IAM) roles vinculados al servicio. Un rol vinculado a un servicio es un tipo único de IAM rol 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 tu 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 la política de confianza y la política de permisos, y esa política de permisos no se puede adjuntar a ninguna otra IAM entidad.

> **Note**
>
> Realice una de las siguientes acciones para especificar un rol 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 Nombre de función incorrecto o ARN 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 sobre otros servicios que admiten funciones vinculadas a un servicio, consulte AWS Servicios que funcionan con IAM y busque los servicios que tienen la palabra «Sí» en la columna Función vinculada al 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 IAM política denominada BatchServiceRolePolicypermite AWS Batch realizar las siguientes acciones en recursos específicos:

- `autoscaling`— Permite AWS Batch crear y gestionar los recursos de Amazon EC2 Auto Scaling. AWS Batch crea y administra grupos de Amazon EC2 Auto Scaling para la mayoría de los entornos de cómputo.
- `ec2`— Permite AWS Batch controlar el ciclo de vida de las EC2 instancias de Amazon, 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 EC2 puntual.
- `ecs`- Permite AWS Batch crear y gestionar ECS clústeres de Amazon, definiciones de tareas y tareas para la ejecución de trabajos.
- `eks`- Permite AWS Batch describir el recurso del EKS clúster de Amazon para las validaciones.
- `iam`- Permite AWS Batch validar y transferir los roles proporcionados por el propietario a AmazonEC2, Amazon EC2 Auto Scaling y AmazonECS.
- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch trabajos.

Debe configurar los permisos para permitir que una IAM entidad (como un usuario, un grupo o un rol) cree, edite o elimine un rol vinculado a un servicio. Para obtener más información, consulte Permisos de roles vinculados a un servicio en la Guía del usuario. IAM

Crear un rol vinculado a un servicio para AWS Batch

No necesita crear manualmente un rol vinculado a servicios. Si estás CreateComputeEnvironment en el AWS Management Console, el o el AWS CLI 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, consulta Apareció un nuevo rol en mi IAM cuenta.

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 hagas CreateComputeEnvironment, AWS Batch vuelve a 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 utilizando. IAM Para obtener más información, consulte Edición de un rol vinculado a un servicio en la Guía del IAMusuario.

Para permitir que una IAM entidad edite la descripción del rol vinculado al servicio AWSServiceRoleForBatch

Añada la siguiente instrucción a la política de permisos. Esto permite a la IAM entidad editar la descripción de un rol vinculado a un 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 IAM entidad elimine el rol vinculado al servicio AWSServiceRoleForBatch

Añada la siguiente instrucción a la política de permisos. Esto permite a la IAM entidad eliminar un rol vinculado al 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 eliminar un rol vinculado IAM a un servicio, primero debe confirmar que el rol no tiene sesiones activas y eliminar todos los entornos de AWS Batch cómputo que utilizan el rol en todas AWS las regiones de una sola partición.

Para comprobar si el rol vinculado a servicio tiene una sesión activa

1. Abra la IAM consola en. https://console.aws.amazon.com/iam/
2. En el panel de navegación, seleccione 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 el estado cambie a DISABLED.
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.

Eliminar un rol vinculado a un servicio en IAM (Consola)

Puede usar la IAM consola para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Inicie sesión en AWS Management Console y abra la IAM consola en. https://console.aws.amazon.com/iam/
2. En el panel de navegación de la IAM consola, selecciona Funciones. 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. Observe las notificaciones de la IAM consola para supervisar el progreso de la eliminación del rol vinculado al servicio. Como la eliminación del rol IAM vinculado al servicio es asincrónica, una vez enviado el rol para su eliminación, la tarea de eliminación puede realizarse correctamente o fallar.

   - 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 sobre cómo limpiar los recursos de ese servicio, consulte este artículo con.Servicios de AWS 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.

Eliminar un rol vinculado a un servicio en IAM ()AWS CLI

Puede usar IAM los comandos 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 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 `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 sobre cómo limpiar los recursos de un servicio que no informa de ningún recurso, consulte los AWS servicios 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.

Eliminar un rol vinculado a un servicio en IAM ()AWS API

Puede usar el para eliminar un IAM API 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 sobre cómo limpiar los recursos de un servicio que no informa de ningún recurso, consulte Servicios de AWS esa sección sobre cómo trabajar 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.

Función de ECS instancia de Amazon

AWS Batch los entornos de cómputo se rellenan con instancias de ECS contenedor de Amazon. Dirigen el agente de ECS contenedores de Amazon a nivel local. El agente de ECS contenedores de Amazon realiza llamadas a varias AWS API operaciones en tu nombre. Por lo tanto, las instancias contenedoras en las que se ejecuta el agente requieren una IAM política y una función para que estos servicios reconozcan que el agente te pertenece. Debes crear un IAM rol y un perfil de instancia para que las instancias contenedoras los usen cuando se lancen. De lo contrario, no puede crear un entorno de computación y lanzar instancias de contenedor en él. Este requisito se aplica a las instancias contenedoras lanzadas con o sin la versión ECS optimizada de Amazon AMI proporcionada por Amazon. Para obtener más información, consulte Función de ECS instancia de Amazon en la Guía para desarrolladores de Amazon Elastic Container Service.

Temas

- Tutorial: Comprueba el rol de ECS instancia de Amazon de tu cuenta

Tutorial: Comprueba el rol de ECS instancia de Amazon de tu cuenta

El rol y el perfil de instancia de Amazon ECS se crean automáticamente en la primera ejecución de la consola. Sin embargo, puedes seguir estos pasos para comprobar si tu cuenta ya tiene el rol de ECS instancia y el perfil de instancia de Amazon. En los siguientes pasos también se explica cómo adjuntar la IAM política gestionada.

Tutorial: **`ecsInstanceRole`** compruébelo en la IAM consola

1. Abra la IAM consola en https://console.aws.amazon.com/iam/.
2. Seleccione 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 Crear rol.
   b. En Tipo de entidad de confianza, elija Servicio de AWS.
   c. Para casos de uso comunes, elija EC2.
   d. Elija Next (Siguiente).
   e. Para ver las políticas de permisos, busca Amazon EC2ContainerServiceforEC2Role.
   f. Selecciona la casilla de verificación situada junto a Amazon EC2ContainerServiceforEC2Role y, a continuación, selecciona Siguiente.
   g. En Role Name (Nombre de rol), escriba `ecsInstanceRole` y elija Create role (Crear rol).

   > **Note**
   > Si utilizas el AWS Management Console para crear un rol para AmazonEC2, la consola crea un perfil de instancia con el mismo nombre que el rol.

Como alternativa, puede usar el AWS CLI para crear el `ecsInstanceRole` IAM rol. En el siguiente ejemplo, se crea un IAM rol con una política de confianza y una política AWS administrada.

Tutorial: Crear un perfil de IAM rol y de instancia (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. Usa el create-instance-profilecomando para crear un perfil de instancia con un nombre`ecsInstanceRole`.

> **Note**
>
> Debes crear los roles y los perfiles de instancia como acciones independientes en AWS CLI y AWS API.

```
$ 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. Usa el comando add-role-to-instance-profile para añadir el `ecsInstanceRole` rol al perfil de la `ecsInstanceRole` instancia.

```
aws iam add-role-to-instance-profile \
    --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole
```

5. Usa el attach-role-policycomando para adjuntar la política `AmazonEC2ContainerServiceforEC2Role` AWS administrada al `ecsInstanceRole` rol.

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
```

```
    --role-name ecsInstanceRole
```

Función de Amazon EC2 en la flota spot

Si crea un entorno informático gestionado que utiliza Amazon EC2 Spot Fleet Instances, debe crear la `AmazonEC2SpotFleetTaggingRole` política. 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 las funciones AWSServiceRoleForEC2SpotFleetvinculadas al AWSServiceRoleForEC2Spotservicio de Amazon EC2 Spot y Spot Fleet. Utilice la siguiente instrucción para crear todos estos roles. 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. IAM

Temas

- Tutorial: Cree funciones de flota EC2 puntual de Amazon en el AWS Management Console
- Tutorial: Cree funciones de flota EC2 puntual de Amazon con el AWS CLI

Tutorial: Cree funciones de flota EC2 puntual de Amazon en el AWS Management Console

Para crear el rol **`AmazonEC2SpotFleetTaggingRole`** IAM vinculado al servicio para Amazon Spot Fleet EC2

1. Abra la IAM consola 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 los casos de uso de otros Servicios de AWS, selecciona EC2y, a continuación, elige EC2: Spot Fleet Tagging.
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 Crear rol.

> **Note**
>
> En el pasado, había dos políticas gestionadas para el rol de Amazon EC2 Spot Fleet.
>
> - Amazon EC2SpotFleetRole: Esta es la política gestionada original para el rol de Spot Fleet. Sin embargo, ya no recomendamos que lo utilice con AWS Batch. Esta política no admite el etiquetado de instancias de spot en entornos de computación, 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.
> - Amazon EC2SpotFleetTaggingRole: esta función proporciona todos los permisos necesarios para etiquetar las instancias EC2 puntuales de Amazon. Utilice esta función para permitir el etiquetado de instancias puntuales en sus entornos AWS Batch informáticos.

Tutorial: Cree funciones de flota EC2 puntual de Amazon con el AWS CLI

Para crear el EC2SpotFleetTaggingRole IAM rol de Amazon para sus entornos de cómputo de Spot Fleet

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 IAM política EC2SpotFleetTaggingRole gestionada por Amazon a tu EC2SpotFleetTaggingRole rol de Amazon, ejecuta el siguiente comando con AWS CLI.

```
$ aws iam attach-role-policy \
  --policy-arn \
    arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
    AmazonEC2SpotFleetTaggingRole
```

Para crear el rol **`AWSServiceRoleForEC2Spot`** IAM vinculado al servicio para Amazon Spot EC2

> **Note**
>
> Si el rol `AWSServiceRoleForEC2Spot` IAM vinculado al servicio 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 **`AWSServiceRoleForEC2SpotFleet`** IAM vinculado al servicio para Amazon Spot Fleet EC2

> **Note**
>
> Si el rol `AWSServiceRoleForEC2SpotFleet` IAM vinculado al servicio 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
```

EventBridge IAMrol

Amazon EventBridge ofrece un flujo casi en tiempo real de los eventos del sistema que describen los cambios en los AWS recursos. AWS Batch los trabajos están disponibles como EventBridge objetivos. Mediante reglas sencillas que puede configurar rápidamente, puede asignar eventos y enviar trabajos de AWS Batch como respuesta a ellos. Antes de poder enviar AWS Batch trabajos con EventBridge reglas y objetivos, EventBridge debe tener permisos para ejecutar los AWS Batch trabajos en su nombre.

> **Note**
>
> Si crea una regla en la EventBridge consola que especifica una AWS Batch cola como destino, puede crear este rol. Para ver un tutorial de ejemplo, consulte AWS Batch puestos de trabajo como EventBridge objetivos. Puede crear el EventBridge rol manualmente mediante la IAM consola. Para obtener instrucciones, consulte Crear un rol mediante políticas de confianza personalizadas (consola) en la Guía del IAM usuario.

La relación de confianza para su EventBridge IAM función debe proporcionar al director del `events.amazonaws.com` servicio la capacidad de asumir la función.

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

Asegúrese de que la política asociada a su EventBridge IAM función 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": "*"
    }
  ]
}
```

Creación de una nube virtual privada

Los recursos informáticos de sus entornos informáticos necesitan acceso a una red externa para comunicarse con AWS Batch los puntos finales de ECS servicio de Amazon. Sin embargo, es posible que tenga trabajos que desee ejecutar en subredes privadas. Para tener la flexibilidad de ejecutar trabajos en una subred pública o privada, cree una VPC que tenga subredes públicas y privadas.

Puede usar Amazon Virtual Private Cloud (AmazonVPC) para lanzar AWS recursos en una red virtual que usted defina. En este tema se proporciona un enlace al VPC asistente de Amazon y una lista de las opciones que se pueden seleccionar.

Crea un VPC

Para obtener información sobre cómo crear un AmazonVPC, consulta la Guía del VPC usuario sobre cómo crear un Amazon VPC únicamente en Amazon y utiliza la siguiente tabla para determinar qué opciones seleccionar.

Opción	Valor
Recursos para crear	VPCúnicamente
Nombre	Si lo desea, proporcione un nombre para suVPC.
IPv4CIDRbloquear	IPv4CIDRentrada manual El tamaño del CIDR bloque debe estar comprendido entre /16 y /28.
IPv6CIDRbloque	Sin IPv6 CIDR bloqueo
Propiedad	Predeterminado

Para obtener más información sobre AmazonVPC, consulta ¿Qué es AmazonVPC? en la Guía del VPC usuario de Amazon.

Siguientes pasos

Una vez que haya creado la suyaVPC, tenga en cuenta 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 Trabajar con grupos de seguridad en la Guía del VPC usuario de Amazon.
- Cree un entorno informático AWS Batch gestionado que incorpore recursos informáticos a su nuevo entornoVPC. Para obtener más información, consulte Crear un entorno de computación.

Si utiliza el asistente de creación de entornos informáticos de la AWS Batch consola, puede especificar el entorno VPC que acaba de crear y las subredes públicas o privadas en las que desea lanzar las instancias.

- Crea una cola de AWS Batch trabajos que se asigne a tu nuevo entorno de cómputo. Para obtener más información, consulte Crear una cola de trabajos.
- Cree una definición de trabajo para ejecutar sus trabajos. Para obtener más información, consulte Crear una definición de trabajo de un solo nodo .
- Envíe un trabajo con su definición de trabajo a la nueva cola de trabajos. Este trabajo recae en el entorno informático que creó con su nueva red VPC y sus subredes. Para obtener más información, consulte Tutorial: enviar un trabajo.

Utilice un punto final de interfaz para acceder AWS Batch

Puede usarlo AWS PrivateLink para crear una conexión privada entre su VPC y AWS Batch. Puede acceder AWS Batch como si estuviera en su casaVPC, sin el uso de una pasarela de Internet, NAT dispositivo, VPN conexión o AWS Direct Connect conexión. Las instancias VPC que tengas no necesitan direcciones IP públicas para acceder 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 los VPCpuntos finales de la interfaz en la AWS PrivateLink guía.

Consideraciones sobre 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 API acciones a través del punto final de la interfaz.

Antes de configurar los VPC puntos finales 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 VPC la interfaz de AmazonECS, pero es posible que necesites los puntos de enlace de

VPC la interfaz para AWS Batch Amazon, Secrets ECR Manager o Amazon CloudWatch Logs, que se describen en los puntos siguientes.

- Para ejecutar los trabajos, debe crear los VPC puntos finales de la interfaz de AmazonECS. Para obtener más información, consulte Interface VPC Endpoints (AWS PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.
- Para permitir que tus trabajos extraigan imágenes privadas de AmazonECR, debes crear los VPC puntos finales de la interfaz de AmazonECR. Para obtener más información, consulte Interface VPC Endpoints (AWS PrivateLink) en la Guía del usuario de Amazon Elastic Container Registry.
- Para permitir que sus trabajos extraigan datos confidenciales de Secrets Manager, debe crear los VPC puntos finales de interfaz para Secrets Manager. Para obtener más información, consulte Uso de Secrets Manager con VPC puntos finales en la Guía del AWS Secrets Manager usuario.
- Si 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 VPC punto final de interfaz para CloudWatch Logs. Para obtener más información, consulte Uso de CloudWatch registros con VPC puntos de enlace de interfaz en la Guía del usuario de Amazon CloudWatch Logs.

- Los trabajos que utilizan los EC2 recursos requieren que las instancias de contenedor en las que se lanzan ejecuten una versión `1.25.1` o posterior del agente de ECS contenedores de Amazon. Para obtener más información, consulte las versiones del agente contenedor de Amazon ECS Linux en la Guía para desarrolladores de Amazon Elastic Container Service.
- VPCActualmente, los puntos de enlace no admiten solicitudes entre regiones. Asegúrese de crear su terminal en la misma región a la que planea realizar las API llamadas. AWS Batch
- VPClos puntos de enlace solo son compatibles con los proporcionados por DNS Amazon a través de Amazon Route 53. Si quieres usar los tuyos propiosDNS, puedes usar el reenvío condicional. DNS Para obtener más información, consulta Conjuntos de DHCP opciones en la Guía del VPC usuario de Amazon.
- El grupo de seguridad adjunto al VPC punto final debe permitir las conexiones entrantes en el puerto 443 desde la subred privada delVPC.
- AWS Batch no admite los puntos finales VPC de la interfaz 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 VPC consola de Amazon 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 la opción privada DNS para el punto final de la interfaz, puede realizar API solicitudes AWS Batch utilizando su DNS nombre 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 final es un IAM recurso que se puede adjuntar a un punto final de interfaz. La política de puntos finales predeterminada permite el acceso total a AWS Batch través del punto final de la interfaz. Para controlar el acceso permitido AWS Batch desde su punto de conexiónVPC, adjunte una política de punto final personalizada al punto final de la interfaz.

Una política de punto de conexión especifica la siguiente información:

- Los principales que pueden realizar acciones (Cuentas de AWS usuarios y IAM roles).

- 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 VPC puntos finales para las 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.
- Diseñando una arquitectura basada en la HIPAA seguridad y el cumplimiento en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar las empresas AWS para crear HIPAA aplicaciones aptas.

> **Note**
>
> No todos son aptos. Servicios de AWS HIPAA Para obtener más información, consulta la Referencia de servicios HIPAA aptos.

- 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. En las guías se resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y se orientan a 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 las 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.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, por ejemplo PCIDSS, cumpliendo con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS 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 API llamadas AWS publicadas para acceder a AWS Batch través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Cifre suites con perfecto secreto (PFS), como (Ephemeral Diffie-Hellman) o DHE 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 firmarse con un identificador de clave de acceso y una clave de acceso secreta asociada a un director. IAM También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puede API realizar estas operaciones 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 utilizar AWS Batch políticas para controlar el acceso desde puntos de enlace específicos o específicos VPCs de Amazon Virtual Private Cloud (AmazonVPC). En efecto, esto aísla el acceso a la red a un AWS Batch recurso determinado únicamente de los recursos específicos VPC de la AWS red.

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 `aws:SourceArn` valor no contiene el ID de la cuenta, como un bucket de Amazon S3ARN, debe usar 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 confuso problema de los diputados es utilizar la clave del contexto ARN de la condición `aws:SourceArn` global con todo el recurso. Si no conoce la totalidad ARN del recurso o si está especificando varios recursos, utilice la clave de condición del contexto `aws:SourceArn` global con caracteres comodín (`*`) para las partes desconocidas del ARN recurso. Por ejemplo, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

En los ejemplos siguientes se muestra cómo utilizar las claves de contexto `aws:SourceArn` y de condición `aws:SourceAccount` global AWS Batch para evitar el confuso problema de los diputados.

Ejemplo: función para acceder a un solo entorno de cómputo

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: función para acceder a varios entornos de cómputo

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/*"
          ]
        }
      }
    }
```

```
  ]
}
```

Registrar AWS Batch API llamadas con AWS CloudTrail

AWS Batch está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en AWS Batch. CloudTrail captura todas API las llamadas AWS Batch como eventos. Las llamadas capturadas incluyen las llamadas desde la AWS Batch consola y las llamadas en código a las AWS Batch API operaciones. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para AWS Batch. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por usted CloudTrail, puede determinar el destinatario de la solicitud AWS Batch, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

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

Temas

- AWS Batch información en CloudTrail
- Referencia: Descripción de las entradas de los archivos de AWS Batch registro

AWS Batch información en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crear la cuenta. Cuando se produce una actividad en AWS Batch, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puedes ver, buscar y descargar los eventos recientes en tu AWS cuenta. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de tu cuenta AWS Batch, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando crea una ruta en la consola, la ruta se aplica a todas AWS las regiones. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento

- CloudTrail Integraciones y servicios compatibles
- Configuración de Amazon SNS Notifications para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas AWS Batch las acciones se registran CloudTrail y se documentan en la https://docs.aws.amazon.com/batch/ última versión APIReference /. Por ejemplo, las llamadas a las `SubmitJob` `DescribeJobs` secciones `ListJobs` y generan entradas en los CloudTrail archivos de registro.

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

- Si la solicitud se realizó con credenciales de IAM usuario o raíz.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el CloudTrail userIdentityElemento.

Referencia: Descripción de las entradas de los archivos de AWS Batch registro

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrailLos archivos de registro no son un registro ordenado de las API llamadas públicas, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `CreateComputeEnvironment` acción.

```
{
  "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"
}
```

Solucionar problemas AWS Batch IAM

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con AWS Batch yIAM.

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 el 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.

El siguiente ejemplo de error se produce cuando un IAM usuario denominado `marymajor` intenta utilizar la consola para realizar una acción en ella. 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 AWS el administrador. 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 admiten políticas basadas en recursos o listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus 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 propiedad, consulte Proporcionar acceso a un IAM usuario en otro Cuenta de AWS de su propiedad en la Guía del IAM usuario. Cuentas de AWS
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo permitir el acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del IAM usuario.
- Para obtener información sobre cómo proporcionar acceso mediante la federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (federación de identidades) en la Guía del IAM usuario.
- Para saber la diferencia entre el uso de roles y políticas basadas en recursos para el acceso entre cuentas, consulte el acceso a recursos entre cuentas IAM en la Guía del usuario. IAM

AWS Step Functions

Puede utilizar la AWS Batch consola para ver detalles sobre las máquinas de estado de Step Functions y las funciones que utilizan.

Secciones

- Tutorial: Vea los detalles de las máquinas de estado
- Tutorial: Editar una máquina de estados
- Tutorial: Ejecute una máquina de estados

Tutorial: Vea los detalles de las máquinas de estado

La AWS Batch consola muestra una lista de las máquinas de estado actuales Región de AWS que contienen al menos un paso del flujo de trabajo que envía un AWS Batch trabajo.

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 a un AWS Batch trabajo JsonPath en la definición de la máquina de estados, los detalles de la función no se pueden mostrar en la AWS Batch consola. 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 página de orquestación del flujo de trabajo de la AWS Batch consola con Step Functions.
2. Creación de una máquina de estado.
   <result>

   La AWS Batch consola abre la página de detalles.
   </result>

Para obtener más información, consulte Step Functions en la Guía para desarrolladores de AWS Step Functions .

Tutorial: Editar una máquina de estados

Cuando desee editar una máquina de estados, AWS Batch abre la página Editar definición de la consola Step Functions.

Para editar una máquina de estado

1. Abra la página de orquestación del flujo de trabajo de la AWS Batch consola 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 .

Tutorial: Ejecute una máquina de estados

Cuando desee ejecutar una máquina de estados, AWS Batch abre la nueva página de ejecución de la consola Step Functions.

Para ejecutar una máquina de estado

1. Abra la página de orquestación del flujo de trabajo de la AWS Batch consola 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 transmisión de eventos 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 de control. 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
- Tutorial: Usa las notificaciones AWS de usuario con AWS Batch
- AWS Batch puestos de trabajo como EventBridge objetivos
- Tutorial: Escuche los eventos de AWS Batch trabajo usando EventBridge

- Tutorial: Envío de alertas de Amazon Simple Notification Service sobre eventos de trabajo fallidos

AWS Batch eventos

AWS Batch envía los eventos de cambio de estado del trabajo a EventBridge. AWS Batch realiza un seguimiento del 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 vas a deserializar JSON datos de eventos mediante programación, asegúrate de que tu aplicación esté preparada para gestionar propiedades desconocidas. Esto es para evitar problemas si se agregan estas propiedades adicionales y cuando se agregan.

Temas

- Recurso: Eventos de cambio de estado de trabajo
- Recurso: Eventos bloqueados por la cola de trabajos

Recurso: Eventos de cambio de estado de trabajo

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 DescribeJobsAPIoperación en la AWS Batch APIReferencia. 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": []
    }
}
```

Recurso: Eventos bloqueados por 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 bloqueadas, consulte. Trabajos bloqueados en estado `RUNNABLE` El mismo motivo también está disponible en el `statusReason` campo de la `DescribeJobs`APIacción.

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 DescribeJobsAPIoperación en la AWS Batch APIReferencia. 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": []
    }
}
```

Tutorial: Usa las notificaciones AWS de 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 Batch.
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 cron trabajo solo se realiza durante las horas de bajo consumo, cuando las instancias EC2 puntuales de Amazon son menos costosas.
- Un AWS Batch trabajo se ejecuta en respuesta a una API operación 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 el principal de servicio y IAM los permisos necesarios para este rol, consulteEventBridge IAMrol.

Temas

- Tutorial: Crear un AWS Batch trabajo programado
- Tutorial: Crea una regla con un patrón de eventos
- Tutorial: Transfiera la información del evento a un AWS Batch objetivo según un cronograma utilizando el transformador EventBridge de entrada

Tutorial: Crear un AWS Batch trabajo programado

El siguiente procedimiento explica cómo crear un AWS Batch trabajo programado y el EventBridge IAM rol requerido.

Para crear un AWS Batch trabajo programado con EventBridge

> **Note**
>
> Este procedimiento funciona para todos AWS Batch los trabajos de Amazon ECSEKS, Amazon 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 miembro 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. PSTel primer lunes de cada mes y, a continuación, introduce una 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 Next (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: introduce el nombre del recurso de Amazon (ARN) de la cola de trabajos para programar tu trabajo.
    - Definición de trabajo: introduzca el nombre y la revisión o la totalidad ARN de la definición de trabajo que desee utilizar en su 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 el IAM rol necesario para que se ejecute la regla. Realice una de las siguientes acciones siguientes:
    - Para crear un IAM rol automáticamente, elija Crear un nuevo rol para este recurso específico.
    - Para usar un IAM rol que ya haya creado, elija Usar un 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, especifica la SQS cola de Amazon que vas a utilizar como cola de letra muerta.

17. (Opcional) Seleccione 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 Next (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.

Tutorial: Crea una regla con un patrón de eventos

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 AWS Batch los trabajos de Amazon ECSEKS, Amazon 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 miembro 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 Next (Siguiente).
11. En Event Source, 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 Next (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: introduce el nombre del recurso de Amazon (ARN) de la cola de trabajos para programar tu trabajo.
    - Definición de trabajo: introduzca el nombre y la revisión o la totalidad ARN de la definición de trabajo que desee utilizar en su 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 el IAM rol necesario para que se ejecute la regla. Realice una de las siguientes acciones siguientes:

    - Para crear un IAM rol automáticamente, elija Crear un nuevo rol para este recurso específico.
    - Para usar un IAM rol que haya creado anteriormente, elija Usar un rol 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, especifica la SQS cola de Amazon que vas a utilizar como cola de letra muerta.

20. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.

21. Elija Next (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 Next (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.

Tutorial: Transfiera la información del evento a un AWS Batch objetivo según un cronograma utilizando 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. PSTel primer lunes de cada mes y, a continuación, introduce una 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 Next (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: introduce el nombre del recurso de Amazon (ARN) de la cola de trabajos para programar tu trabajo.
    - Definición de trabajo: introduzca el nombre y la revisión o la totalidad ARN de la definición de trabajo que desee utilizar en su 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 el IAM rol necesario para que se ejecute la regla. Realice una de las siguientes acciones siguientes:
    - Para crear un IAM rol automáticamente, elija Crear un nuevo rol para este recurso específico.
    - Para usar un IAM rol que ya haya creado, elija Usar un 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 Batch Job State Change, utilice el siguiente JSON formato.

```
{
    "instance": "$.detail.jobId",
    "state": "$.detail.status"
}
```

21. En Plantilla, introduzca lo siguiente.

```
{
    "instance": <jobId> ,
    "status": <status>
}
```

22. Elija 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, especifica la SQS cola de Amazon que vas a utilizar como cola de letra muerta.
26. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.
27. Elija Next (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 Next (Siguiente).

30. En Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, elija Crear regla.

Tutorial: Escuche los eventos de AWS Batch trabajo usando EventBridge

En este tutorial, configurará una AWS Lambda función sencilla que escucha los eventos de AWS Batch trabajo y los escribe en un flujo de registro de CloudWatch registros.

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 los tutoriales 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.

Temas

- Tutorial: Creación de la función Lambda
- Tutorial: Registre la regla de eventos
- Tutorial: Pruebe su configuración

Tutorial: Creación de la función Lambda

En este procedimiento, se crea una función Lambda sencilla que sirva de destino para los mensajes de flujo de AWS Batch eventos.

Para crear una función de Lambda de destino

1. Abra la AWS Lambda consola 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 la función, introduzca batch-event-stream-handler.
4. En Tiempo de ejecución, seleccione Python 3.8.
5. Seleccione 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))
```

   Esta es 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, los detalles del evento aparecen en el flujo de registro de CloudWatch Logs asociado a esta función de Lambda.

7. Elija Implementar.

Tutorial: Registre la regla de eventos

En esta sección, creará una regla de EventBridge eventos que capture los eventos de trabajo que provienen de sus AWS Batch recursos. Esta regla captura todos los eventos que se producen AWS Batch dentro de la cuenta en la que está definida. 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 la AWS Management Console para crear una regla de eventos, la consola añade automáticamente los IAM permisos para llamar EventBridge a la función Lambda. Sin embargo, si va a crear una regla de eventos mediante AWS CLI, debe conceder los permisos de forma explícita. Para obtener más información, consulta Eventos y patrones de eventos en la Guía del EventBridge usuario de Amazon.

Para crear 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. Elija 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, selecciona Patrones personalizados (JSONeditor).
10. Pegue el siguiente patrón de eventos en el área de texto.

```
{
  "source": [
    "aws.batch"
  ]
}
```

    Esta regla se aplica a todos los AWS Batch grupos y a todos los AWS Batch eventos. También puede crear una regla más específica para filtrar algunos resultados.
11. Elija Next (Siguiente).
12. En Tipos de destino, seleccione Servicio de AWS .
13. En Seleccionar un objetivo, elija Función de Lambda y seleccione su función de Lambda.
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, elige si deseas utilizar una cola estándar de Amazon como SQS 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 siguientes acciones siguientes:

- Seleccione Ninguna para no usar una cola de mensajes fallidos.
- Selecciona Selecciona una SQS cola de Amazon en la AWS cuenta corriente para usarla como cola de letra muerta y, a continuación, selecciona la cola que quieres usar en el menú desplegable.
- Selecciona Selecciona una SQS cola de Amazon en otra AWS cuenta como cola de letra muerta y, a continuación, introduce la ARN cola que quieres usar. Debes adjuntar a la cola una política basada en recursos que 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. Elija Next (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. Elija Next (Siguiente).
18. Revise los detalles de la regla y seleccione Crear regla.

Tutorial: Pruebe su configuración

Ahora puede probar su EventBridge configuración enviando un trabajo a su cola de trabajos. Si todo está configurado correctamente, la función Lambda se activa y escribe los datos del evento en un flujo de registro de CloudWatch Logs para la función.

Para probar la configuración

1. Abra la AWS Batch consola en. https://console.aws.amazon.com/batch/
2. Envíe un nuevo AWS Batch trabajo. Para obtener más información, consulte Tutorial: enviar un trabajo.
3. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
4. En el panel de navegación, elija Registros y seleccione el grupo de registros para la función 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 sobre eventos de trabajo fallidos

En este tutorial, configurará una regla de EventBridge eventos de Amazon que solo captura los eventos de trabajo en los que el trabajo ha 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 has configurado tus SNS alertas de Amazon correctamente.

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 los tutoriales para crearlos.

Temas

- Tutorial: Crear un SNS tema de Amazon y suscribirse a él
- Tutorial: Registra una regla de evento
- Tutorial: Pon a prueba tu regla
- Regla alternativa: cola de trabajos por lotes bloqueada

Tutorial: Crear un SNS tema de Amazon y suscribirse a él

En este tutorial, configurarás un SNS tema de Amazon para que sirva como objetivo de eventos para tu nueva regla de eventos.

Para crear un SNS tema de Amazon

1. Abre la SNS consola Amazon en la https://console.aws.amazon.com/sns/versión 3/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, selecciona 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.

Tutorial: Registra una regla de evento

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. Elija 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, selecciona Patrones personalizados (JSONeditor).
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 trabajo`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. Elija Next (Siguiente).
12. En Tipos de destino, seleccione Servicio de AWS .
13. En Seleccione un objetivo, elija un SNStema 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, elige si deseas utilizar una cola estándar de Amazon como SQS 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 siguientes acciones siguientes:
        - Seleccione Ninguna para no usar una cola de mensajes fallidos.
        - Selecciona Selecciona una SQS cola de Amazon en la AWS cuenta corriente para usarla como cola de letra muerta y, a continuación, selecciona la cola que quieres usar en el menú desplegable.
        - Selecciona Selecciona una SQS cola de Amazon en otra AWS cuenta como cola de letra muerta y, a continuación, introduce la ARN cola que quieres usar. Debes adjuntar a la cola una política basada en recursos que 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. Elija Next (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. Elija Next (Siguiente).
18. Revise los detalles de la regla y seleccione Crear regla.

Tutorial: Pon a prueba tu 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 Tutorial: 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 si la cola de trabajos por lotes está bloqueada, repita estos tutoriales con las siguientes modificaciones:

1. En Tutorial: Crear un SNS tema de Amazon y suscribirse a él, utilice `BlockedJobQueue` como nombre del tema.
2. En Tutorial: Registra una regla de evento, utilice el siguiente patrón en el JSON editor:

   ```
   {
      "detail-type": [
        "Batch Job Queue Blocked"
      ],
      "source": [
        "aws.batch"
      ]
   }
   ```

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 instancias que admiten las EFA, consulte los tipos de instancias compatibles en la Guía del usuario de Amazon EC2.

  > **Tip**
  >
  > Para ver una lista de los tipos de instancias que admiten las EFA en una instancia Región de AWS, ejecuta el siguiente comando. A continuación, compara la lista que aparece con la lista de tipos de instancias disponibles en la AWS Batch consola.
  >
  > ```
  > $ 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 LinuxParametersmiembros es diferente para los trabajos paralelos de varios nodos y los trabajos de contenedores 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 de Amazon EC2.

Supervisar AWS Batch

El monitoreo es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de AWS Batch su AWS solución.

Le recomendamos encarecidamente que recopile datos de supervisión de todas las partes de su AWS solución para facilitar la depuración de un fallo multipunto, en caso de que se produzca. Comience por crear un plan de monitoreo que responda a las siguientes preguntas. Si no estás seguro de cómo responder a estas preguntas, puedes seguir utilizando Amazon CloudWatch Logs para establecer tus niveles de referencia de rendimiento.

- ¿Cuáles son los objetivos de la monitorización?
- ¿Qué recursos va a monitorizar?
- ¿Con qué frecuencia va a monitorizar estos recursos?
- ¿Qué herramientas de monitorización va a utilizar?
- ¿Quién se encargará de realizar las tareas de monitoreo?
- ¿Quién debería recibir una notificación cuando surjan problemas?

El siguiente paso es establecer una línea base de AWS Batch rendimiento normal en su entorno midiendo el rendimiento en distintos momentos y bajo diferentes condiciones de carga. Mientras monitorea AWS Batch, guarde los datos de monitoreo históricos para poder compararlos con los datos de rendimiento actuales. Esto le ayudará a identificar patrones de rendimiento normales y anomalías de rendimiento, así como a idear métodos para abordarlos.

Los temas de esta sección pueden ayudarlo a iniciar el registro y el monitoreo de AWS Batch.

Temas

- Uso CloudWatch de registros con AWS Batch
- AWS Batch CloudWatch Información sobre contenedores
- Usa CloudWatch Logs para monitorear AWS Batch los EKS trabajos de Amazon

Uso CloudWatch de registros con AWS Batch

Puede configurar sus AWS Batch trabajos en EC2 los recursos para enviar información de registro y métricas detalladas a CloudWatch los registros. De esta forma, puede ver distintos registros

desde sus trabajos en una ubicación cómoda. Para obtener más información sobre CloudWatch los registros, consulta ¿Qué es Amazon CloudWatch Logs? en la Guía del CloudWatch usuario de Amazon.

> **Note**
>
> De forma predeterminada, CloudWatch los registros están activados para los AWS contenedores Fargate.

Para activar y personalizar el registro de CloudWatch registros, revisa las siguientes tareas de configuración únicas:

- Para los entornos de AWS Batch procesamiento que se basan en EC2 recursos, añada una IAM política al `ecsInstanceRole` rol. Para obtener más información, consulte the section called "Tutorial: Añadir una política de CloudWatch registros IAM".
- Cree una plantilla de EC2 lanzamiento de Amazon que incluya una CloudWatch supervisión detallada y, a continuación, especifique la plantilla cuando cree su entorno AWS Batch informático. También puede instalar el CloudWatch agente en una imagen existente y, a continuación, especificar la imagen en el asistente de AWS Batch primera ejecución.
- (Opcional) Configure el controlador awslogs. Puede añadir parámetros que cambien el comportamiento predeterminado tanto en los recursos como en EC2 los de Fargate. Para obtener más información, consulte the section called "Utilice el controlador de registro awslogs".

Temas

- Tutorial: Añadir una política de CloudWatch registros IAM
- Instale y configure el CloudWatch agente
- Tutorial: Ver CloudWatch registros

Tutorial: Añadir una política de CloudWatch registros IAM

Antes de que sus trabajos puedan enviar datos de registro y métricas detalladas a CloudWatch Logs, debe crear una IAM política que utilice los CloudWatch registrosAPIs. Después de crear la IAM política, adjúntela al `ecsInstanceRole` rol.

> **Note**
>
> Si la `ECS-CloudWatchLogs` política no está asociada al `ecsInstanceRole` rol, las métricas básicas aún se pueden enviar a CloudWatch los registros. Sin embargo, las métricas básicas no incluyen datos de registro ni métricas detalladas, como el espacio libre en disco.

AWS Batch los entornos de cómputo utilizan EC2 los recursos de Amazon. Al crear un entorno de cómputo mediante el asistente de AWS Batch primera ejecución, AWS Batch crea el `ecsInstanceRole` rol y configura el entorno con él.

Si no utiliza el asistente de primera ejecución, puede especificar la `ecsInstanceRole` función al crear un entorno informático en el quirófano. AWS Command Line Interface AWS Batch API Para obtener más información, consulta la Referencia o AWS Batch APIReferencia de AWS CLI comandos.

Para crear la política de IAM **`ECS-CloudWatchLogs`**

1. Abra la IAM consola en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Políticas.
3. Elija Create Policy (Crear política).
4. Elija JSONy, a continuación, introduzca la siguiente política:

```
{
    "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. Abre la IAM consola 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 ECS instancia de Amazon para crear el rol.
4. Elija Agregar permisos y luego Adjuntar políticas.
5. Elija la CloudWatchLogs política ECS- y, a continuación, elija Adjuntar política.

Instale y configure el CloudWatch agente

Puedes crear una plantilla de EC2 lanzamiento de Amazon que incluya la CloudWatch supervisión. Para obtener más información, consulta Lanzar una instancia a partir de una plantilla de lanzamiento y Detalles avanzados en la Guía del EC2 usuario de Amazon.

También puedes instalar el CloudWatch agente en un Amazon existente EC2 AMI y, a continuación, especificar la imagen en el asistente de AWS Batch primera ejecución. Para obtener más información, consulte Instalación del CloudWatch agente yCómo empezar con AWS Batch los tutoriales.

> **Note**
>
> Los AWS Fargate recursos no admiten plantillas de lanzamiento.

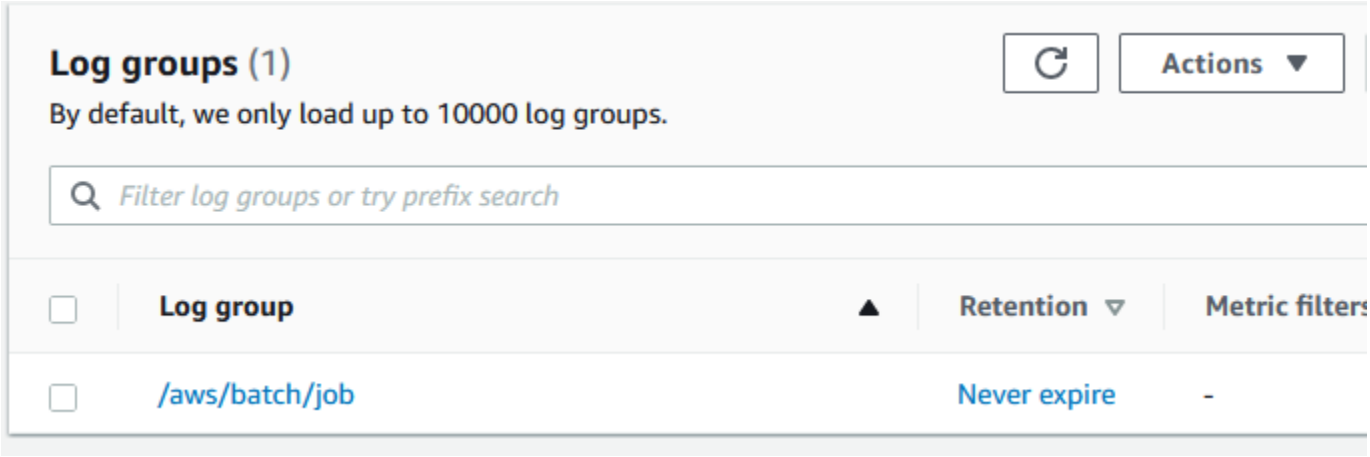
Tutorial: Ver CloudWatch registros

Puede ver y buscar CloudWatch registros en el AWS Management Console.

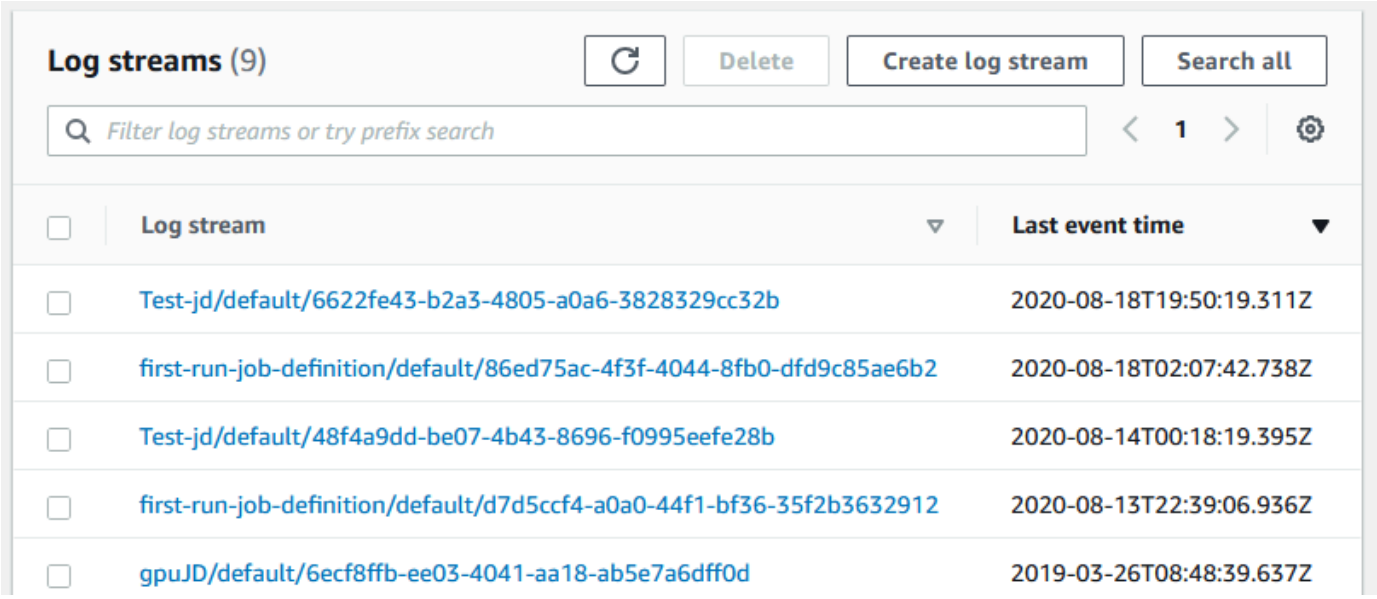
> **Note**
>
> Los datos pueden tardar unos minutos en mostrarse en CloudWatch los registros.

Para ver los datos CloudWatch de tus registros

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, seleccione Registros y, a continuación, Grupos de registros.

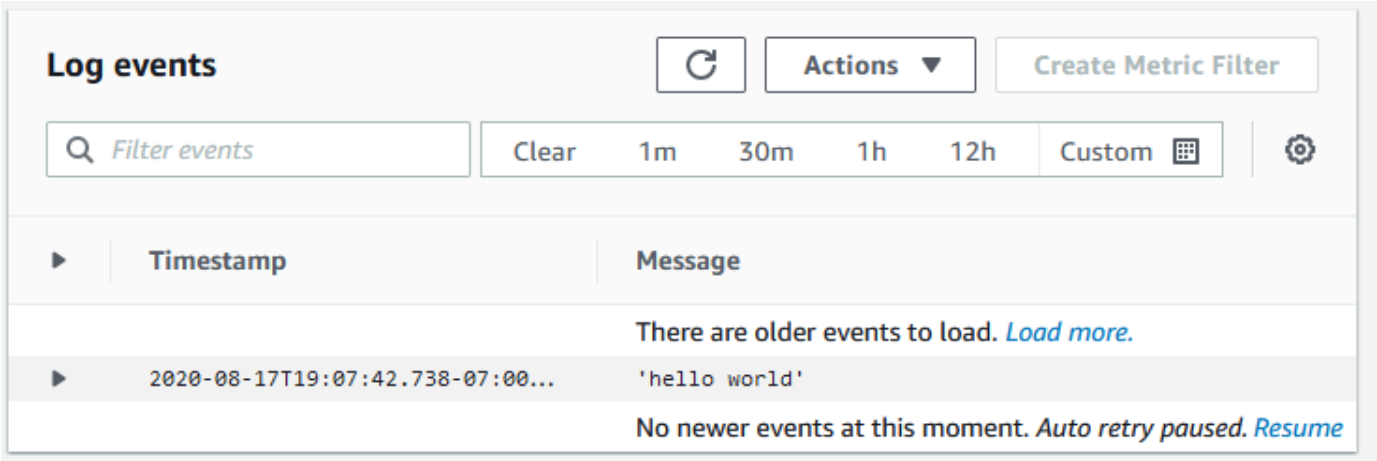


3. Seleccione un grupo de registros que desea ver.



4. Elija el flujo de registros que desea ver. De forma predeterminada, las secuencias se identifican con los primeros 200 caracteres del nombre del trabajo y el ID de la ECS tarea de Amazon.

> **Tip**
>
> Para descargar los datos del flujo de registro, elija Acciones.



AWS Batch CloudWatch Información sobre contenedores

CloudWatch Container Insights recopila, agrega y resume las métricas y los registros de sus trabajos y entornos AWS Batch informáticos. Las métricas incluyen la utilización de la memoriaCPU, el disco y la red. Puede añadir estas métricas a los CloudWatch paneles.

Los datos operativos se recopilan como eventos de registro de rendimiento. Se trata de entradas que utilizan un JSON esquema estructurado que permite ingerir y almacenar datos de alta cardinalidad a escala. A partir de estos datos, CloudWatch crea métricas agregadas de nivel superior en el entorno informático y a nivel de trabajo como métricas. CloudWatch Para obtener más información, consulta Registros estructurados de Container Insights para Amazon ECS en la Guía del CloudWatch usuario de Amazon.

> **Important**
>
> CloudWatch Los datos de Container Insights se cobran como métricas personalizadas por CloudWatch. Para obtener más información, consulta los precios de Amazon CloudWatch Events

Temas

- Tutorial: activar Container Insights

Tutorial: activar Container Insights

Complete los siguientes pasos para activar Container Insights en los entornos de AWS Batch cómputo.

1. Abra la consola de AWS Batch.
2. Elija Entornos de computación.
3. Elija el entorno de desarrollo que quiera.
4. En el caso de 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, consulta las métricas de Amazon ECS Container Insights en la Guía del CloudWatch usuario de Amazon.

- **`JobCount`** — El número de trabajos que se ejecutan en el entorno de computación.
- **`ContainerInstanceCount`**— El número de instancias de Amazon Elastic Compute Cloud que ejecutan el ECS agente de Amazon y están registradas en el entorno informático.
- **`MemoryReserved`** — La memoria que se reserva para los trabajos del entorno de computación. 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 de computación. Esta métrica se recopila únicamente para las tareas que tienen una reserva de memoria definida en su definición de tarea.
- **`CpuReserved`**— Las CPU unidades reservadas por las tareas del entorno de cómputo. Esta métrica se recopila solo para los trabajos que tienen una CPU reserva definida en su definición de trabajo.
- **`CpuUtilized`**— Las CPU unidades utilizadas por los trabajos en el entorno informático. Esta métrica se recopila solo para los trabajos que tienen una CPU reserva definida en su definición de trabajo.
- **`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.

Usa CloudWatch Logs para monitorear AWS Batch los EKS trabajos de Amazon

Puedes usar Amazon CloudWatch Logs para monitorear, almacenar y ver todos tus archivos de registro en un solo lugar. Con CloudWatch Logs, puede buscar, filtrar y analizar los datos de registro de varias fuentes.

Puede descargar un AWS formulario Fluent Bit imagen que incluye un complemento para monitorear los EKS trabajos de Amazon AWS Batch en CloudWatch Logs. Fluent Bit es un procesador y

reenviador de registros de código abierto que es a la vez Docker y Kubernetes compatible. Le recomendamos que utilice Fluent Bit como router de registro porque consume menos recursos que Fluentd. Para obtener más información, consulte Uso de la imagen AWS para Fluent Bit.

Requisitos previos

Adjunte la `CloudWatchAgentServerPolicy` política a la AWS Identity and Access Management política de sus nodos de trabajo. Para obtener más información, consulte Verificación de los requisitos previos.

Instálelo AWS para Fluent Bit

Para obtener instrucciones sobre cómo instalar AWS para Fluent Bit y crear los CloudWatch grupos, consulte Configuración Fluent Bito Inicio rápido con el CloudWatch agente y Fluent Bit.

> **Tip**
>
> Recuerda que Fluent Bit utiliza 0,5 CPU y 100 MB de memoria en AWS Batch los nodos. Esto reduce la capacidad total disponible para los AWS Batch trabajos. Tenga esto en cuenta a la hora de dimensionar sus trabajos.

Active Fluent Bit para los nodos AWS Batch

Para garantizar la Fluent Bit el registro DaemonSet se ejecuta en los nodos AWS Batch gestionados, modifique la Fluent Bit DaemonSet tolerancias:

```
tolerations:
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
```

Etiquete sus AWS Batch recursos

Para ayudarte a administrar tus AWS Batch recursos, puedes asignar tus propios metadatos a cada recurso en forma de etiquetas. En este tema se describe qué son las etiquetas y cómo crearlas.

Temas

- Conceptos básicos de etiquetas
- Etiquetar los recursos
- Restricciones de las etiquetas
- Tutorial: Administra las etiquetas mediante la consola
- Gestione las etiquetas mediante CLI o API

Conceptos básicos de etiquetas

Una etiqueta es una etiqueta que se asigna a un AWS recurso. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario.

Las etiquetas te permiten clasificar tus AWS recursos, por ejemplo, por su propósito, 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, puedes definir un conjunto de etiquetas para tus AWS Batch servicios que te ayuden 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 AWS Batch y se interpretan estrictamente como una cadena 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 las AWS Management Console AWS CLI, las y las AWS Batch API.

Si utilizas AWS Identity and Access Management (IAM), puedes controlar qué usuarios de tu AWS cuenta tienen permiso para crear, editar o eliminar etiquetas.

Etiquetar los recursos

Puede etiquetar entornos AWS Batch informáticos, trabajos, definiciones de trabajos, colas de trabajos y políticas de programación nuevos o existentes.

Si utiliza la AWS Batch consola, puede aplicar etiquetas a los recursos nuevos cuando se creen o a los recursos existentes en cualquier momento mediante la pestaña Etiquetas de la página de recursos correspondiente.

Si utilizas las AWS Batch API, las o una AWS CLI AWS SDK, puedes aplicar etiquetas a los nuevos recursos mediante el `tags` parámetro de la API acción correspondiente o a los recursos existentes mediante la `TagResource` API acción. 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 siguiente tabla se describen los AWS Batch recursos que se pueden etiquetar y los recursos que se pueden etiquetar al crearlos.

Etiqueta el soporte para AWS Batch los recursos

Recurso	Admite etiquetas	Admite la propagación de etiquetas	Admite el etiquetado al crear (AWS Batch API, AWS CLI, AWS SDK)
AWS Batch entornos de cómputo	Sí	No. Las etiquetas del entorno de computación no se propagan a ningún otro recurso. Las etiquetas de los recursos se	Sí

Recurso	Admite etiquetas	Admite la propagación de etiquetas	Admite el etiquetado al crear (AWS Batch API, AWS CLI, AWS SDK)
		especifican en el elemento de etiquetas del computeRe sources objeto transferido en la CreateComputeEnvir onmentAPIoperación.	
AWS Batch puestos de trabajo	Sí	Sí	Sí
AWS Batch definicio nes de puestos	Sí	No	Sí
AWS Batch colas de trabajos	Sí	No	Sí
AWS Batch políticas de programación	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 clave: 128 caracteres Unicode en -8 UTF
- Longitud máxima del valor: 256 caracteres Unicode en -8 UTF
- Si su esquema de etiquetado se usa en varios AWS servicios y recursos, recuerde que otros servicios pueden tener restricciones en cuanto a los caracteres permitidos. Los caracteres generalmente permitidos son letras, números, espacios representables en UTF -8 y 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 como prefijo para las claves o los valores, ya que está reservado para su uso. AWS Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas con este prefijo no se tienen en cuenta para el límite. tags-per-resource

Tutorial: Administra las etiquetas mediante la consola

Con la AWS Batch consola, puede administrar las etiquetas asociadas a los entornos informáticos, los trabajos, las definiciones de trabajos y las colas de trabajos nuevos o existentes.

Añada etiquetas a un recurso individual al crearlo

Puede añadir etiquetas a los entornos AWS Batch informáticos, los trabajos, las definiciones de trabajos, las colas de trabajos y las políticas de programación al crearlos.

Agregar y eliminar etiquetas en un recurso individual

AWS Batch le permite añadir o eliminar las etiquetas asociadas a sus clústeres directamente desde la página del recurso.

Para agregar o eliminar una etiqueta en un recurso individual

1. Abra la AWS Batch consola 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.

Gestione las etiquetas mediante CLI o API

Use los siguientes AWS CLI comandos u AWS Batch API operaciones para agregar, actualizar, enumerar y eliminar las etiquetas de sus recursos.

Etiqueta el soporte para AWS Batch los recursos

Tarea	APIacción	AWS CLI	AWS Tools for Windows PowerShell
Agregar o sobrescribir una o varias etiquetas.	TagResource	tag-resource	Añadir- BATResourceTag
Eliminar una o varias etiquetas.	UntagResource	untag-resource	Eliminar- BATResourceTag
Enumera las etiquetas de un recurso	ListTagsForResource	list-tags-for-resource	Obtenga- 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	APIacción	AWS CLI	AWS Tools for Windows PowerShell
Crear un entorno de computación	CreateComputeEnvironment	create-compute-environment	Nuevo- BATComputeEnvironment
Crear una cola de trabajos	CreateJobQueue	create-job-queue	Nuevo- BATJobQueue
Crear una política de programación	CreateSchedulingPolicy	create-scheduling-policy	Nuevo- BATSchedulingPolicy
Registrar una definición de trabajo	RegisterJobDefinition	register-job-definition	Registrarse-BATJobDefinition
Enviar un trabajo	SubmitJob	submit-job	Enviar- BATJob

Mejores prácticas para AWS Batch

Se puede utilizar AWS Batch para ejecutar una variedad de cargas de trabajo computacionales exigentes a escala sin administrar una arquitectura compleja. AWS Batch Los trabajos se pueden utilizar en una amplia gama de casos de uso en áreas como la epidemiología, los juegos y el aprendizaje automático.

En este tema se describen las prácticas recomendadas que se deben tener en cuenta a la hora de utilizarlas AWS Batch y las directrices sobre cómo ejecutar y optimizar las cargas de trabajo al utilizarlas. AWS Batch

Temas

- ¿Cuándo usar AWS Batch
- Lista de comprobación para ejecutarla a escala
- Optimice los contenedores y AMIs
- Elija el recurso de entorno de computación adecuado
- Amazon EC2 On-Demand o Amazon EC2 Spot
- Utilice las prácticas recomendadas de Amazon EC2 Spot para AWS Batch
- Errores comunes y solución de problemas

¿Cuándo usar AWS 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. 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 junte sus tareas antes de enviarlas. AWS Batch A continuación, configure sus AWS Batch trabajos 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 ti binpack los trabajos, recorre los grupos de tareas dentro de tu AWS Batch trabajo.

- Trabajos que deben ejecutarse inmediatamente: AWS Batch pueden procesar los trabajos rápidamente. Sin embargo, AWS Batch es un programador y optimiza 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 EKS es un enfoque basado en servicios con Amazon ECS o Amazon.

Lista de comprobación para ejecutarla a escala

Antes de ejecutar una gran carga de trabajo con 50 mil o másvCPUs, considere la siguiente lista de verificación.

> **Note**
>
> Si planea gestionar una gran carga de trabajo con un millón o más vCPUs o si necesita orientación para trabajar a gran escala, póngase en contacto con su AWS equipo.

- Comprueba tus EC2 cuotas de Amazon: comprueba tus EC2 cuotas de Amazon (también conocidas como límites) en el panel Service Quotas del AWS Management Console. Si es necesario, solicita un aumento de cuota para tu número máximo de EC2 instancias de Amazon. Recuerde que las instancias Amazon EC2 Spot y Amazon On-Demand tienen cuotas distintas. Para obtener más información, consulte Introducción a Service Quotas.
- Verifica tu cuota de Amazon Elastic Block Store para cada región: cada instancia usa un GP3 volumen GP2 o 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: si se trata de un trabajo que cueste un millón o másvCPUs, comience con menos y aumente gradualmente para poder identificar los cuellos de botella en una fase temprana. Por ejemplo, comience por

contar con 50 mil. vCPUs Luego, aumente el conteo a 200 milvCPUs, luego a 500 milvCPUs, y así sucesivamente. En otras palabras, continúe aumentando gradualmente el CPU recuento de v hasta alcanzar el número deseado devCPUs.

- 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 rupturas incluso al escalar de 1 mil a 5 milvCPUs. Puedes usar Amazon CloudWatch Logs para revisar los datos de registro o usar CloudWatch Embedded Metrics mediante una biblioteca cliente. Para obtener más información, consulte la referencia del agente de CloudWatch Logs y `aws-embedded-metrics`

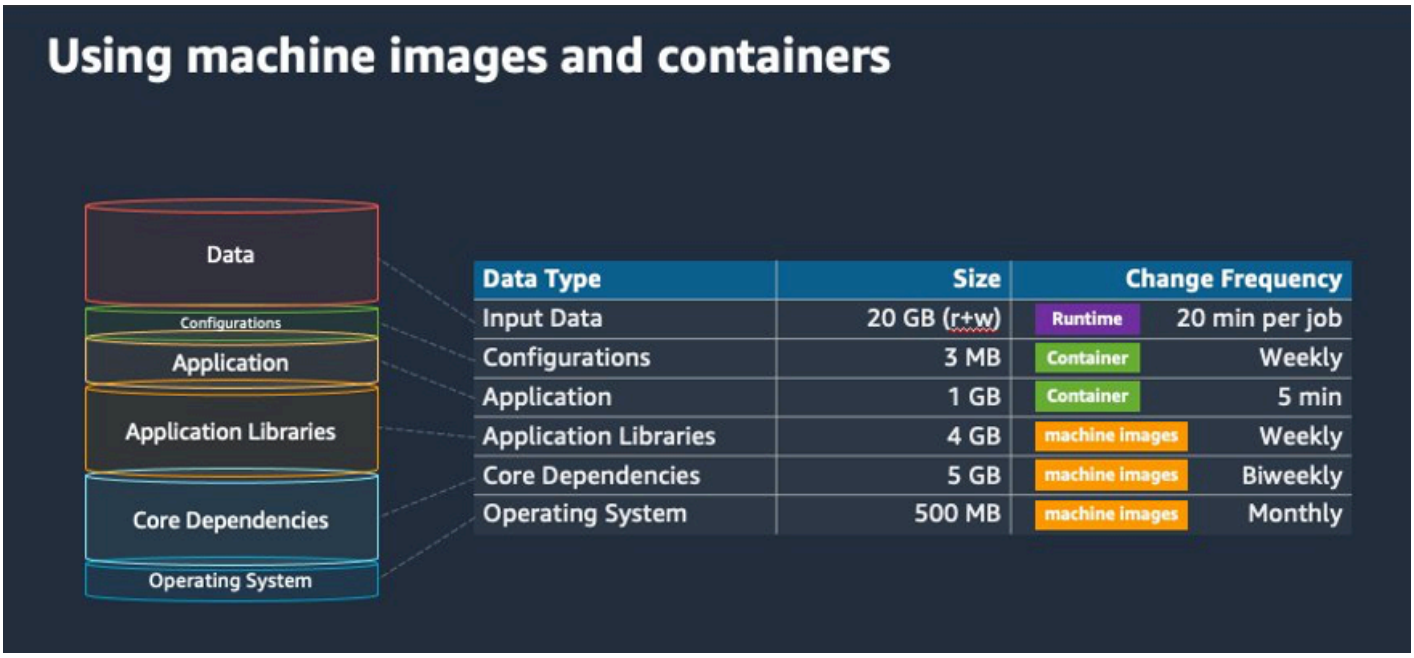
Optimice los contenedores y AMIs

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, descarga 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 trabaja a escala y puede gestionar cargas de trabajo de hasta más de un millónvCPUs.



Elija el recurso de entorno de computación 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 vCPUs o menosGPUs, no y 120 GiB de memoria o menos.

Para obtener más información, consulte Cuándo usar Fargate.

Si tienes los siguientes requisitos, te recomendamos que utilices EC2 instancias de Amazon:

- Necesita un mayor control sobre la selección de instancias o necesita utilizar tipos de instancias específicos.
- Sus trabajos requieren recursos que no AWS Fargate puede proporcionar, como más memoriaGPUs, un adaptador personalizado AMI o el Amazon Elastic Fabric Adapter.

- Necesita un alto nivel de rendimiento o simultaneidad.
- Debe personalizar su AMI plantilla de Amazon EC2 Launch o acceder a parámetros especiales de Linux.

Con AmazonEC2, puede ajustar con mayor precisión su carga de trabajo a sus requisitos específicos y ejecutar a escala si es necesario.

Amazon EC2 On-Demand o Amazon EC2 Spot

La mayoría de AWS Batch los clientes utilizan las instancias de Amazon EC2 Spot debido a los ahorros que suponen 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 de Amazon EC2 On-Demand:

- 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 objetivo estricto SLO (de nivel de servicio) 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 Amazon EC2 Spot:

- 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 instancia 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, envía tus trabajos en una cola conectada a entornos informáticos que se ejecutan en instancias de Amazon EC2 Spot. Si se interrumpe un trabajo, busca el evento de Amazon EventBridge y correlaciónalo con la recuperación de una instancia puntual. A continuación, vuelva a enviar el trabajo a una cola bajo demanda mediante una función o. AWS Lambda AWS Step Functions Para obtener más

información[Tutorial: Envío de alertas de Amazon Simple Notification Service sobre eventos de trabajo fallidos](), consulte [las prácticas recomendadas para gestionar las interrupciones de Amazon EC2 Spot Instance]() y [Manage AWS Batch with 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.

Utilice las prácticas recomendadas de Amazon EC2 Spot para AWS Batch

Si elige las instancias puntuales de Amazon Elastic Compute Cloud (EC2), es probable que pueda optimizar su flujo de trabajo para ahorrar costos, a veces de manera significativa. Para obtener más información, consulta [las prácticas recomendadas para Amazon EC2 Spot]().

Para optimizar tu flujo de trabajo y ahorrar costes, ten en cuenta las siguientes prácticas recomendadas de Amazon EC2 Spot para AWS Batch:

- Elija la estrategia **`SPOT_CAPACITY_OPTIMIZED`** de asignación: AWS Batch elija las EC2 instancias de Amazon de entre los grupos de capacidad más profundos de Amazon EC2 Spot. Si le preocupan las interrupciones, esta es una opción adecuada. Para obtener más información, consulte [Estrategias de asignación de tipos de instancia para AWS Batch]().
- 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 del trabajo: recomendamos no ejecutar trabajos que tarden una hora o más cuando se utilizan instancias de Amazon EC2 Spot 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 AWS Batch trabajos, 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 de Amazon EC2 Spot, consulta el panel de control de interrupciones de Amazon Spot.

Pues 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 el AWS CLI, puede configurar una estrategia de reintentos. 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`.

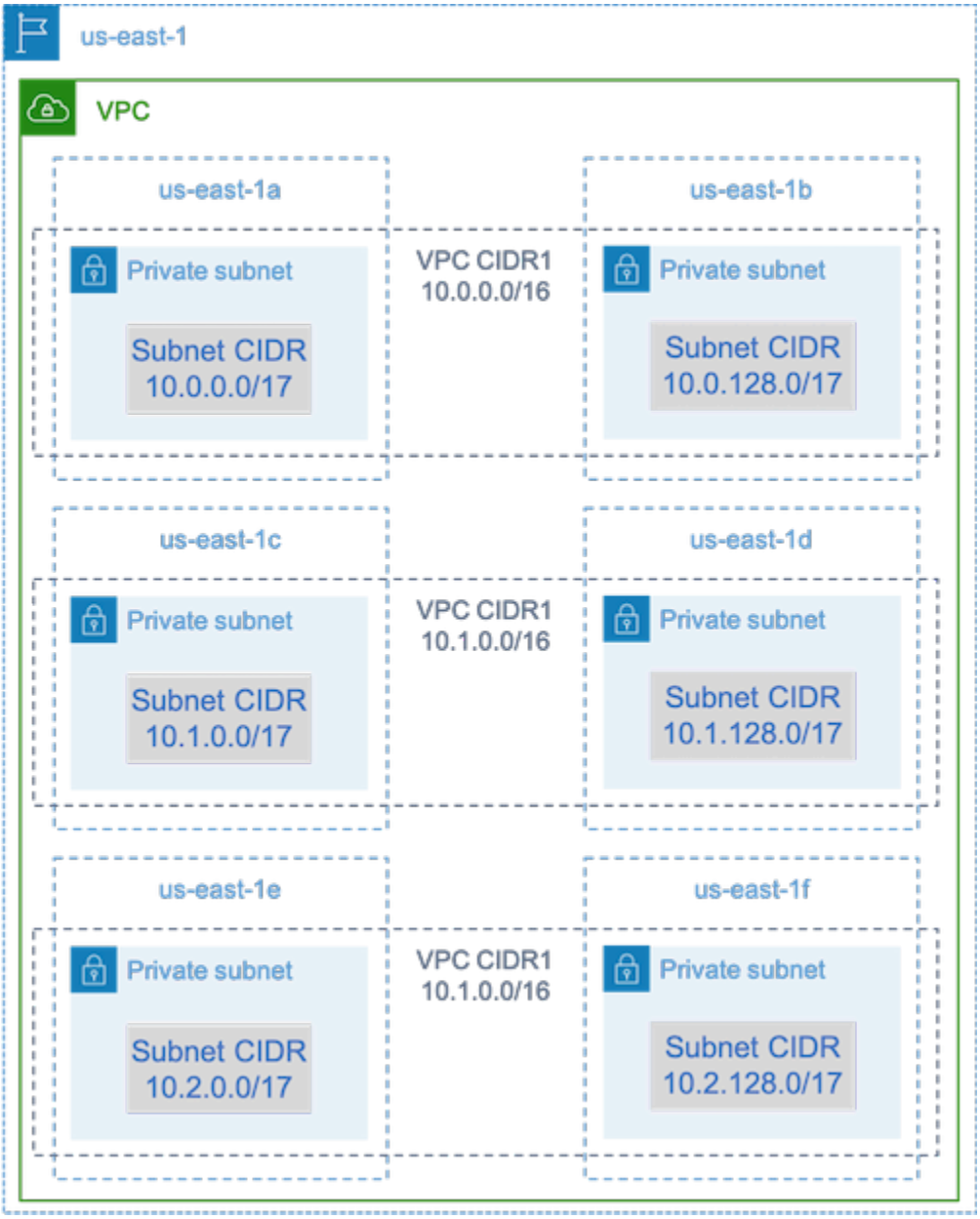
```
"retryStrategy": {
    "attempts": 5,
    "evaluateOnExit":
    [{
        "onStatusReason" :"Host EC2*",
        "action": "RETRY"
    },{
      "onReason" : "*"
        "action": "EXIT"
    }]
}
```

- 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 Amazon EC2 Spot que se recuperan y las zonas de disponibilidad en las que se encuentran las instancias Spot. Para obtener más información, consulte Interrupciones de instancias de spot.

Errores comunes y solución de problemas

Los errores AWS Batch suelen producirse en el nivel de la aplicación o se deben a configuraciones de instancias que no cumplen los requisitos específicos de tu trabajo. Otros problemas incluyen que los trabajos se atasquen en un estado `RUNNABLE` o que los entornos de computación 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 de computación en un estado `INVALID`, consulte Entorno de computación `INVALID`.

- Comprueba CPU las cuotas de Amazon EC2 Spot v: comprueba que tus cuotas de servicio actuales cumplen con los requisitos del trabajo. Por ejemplo, supongamos que tu cuota de servicio actual es de 256 vCPUs y el trabajo requiere 10 000vCPUs. Entonces, la cuota de servicio no cumple con el requisito del trabajo. Para obtener más información e instrucciones de solución de problemas, consulta las cuotas de EC2 servicio de Amazon y ¿Cómo aumento la cuota de servicio de mi AmazonEC2resources? .
- 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 en sus subredes VPC y en sus subredes puede limitar la cantidad de instancias que puede crear. Utilice los enrutamientos entre dominios sin clase (CIDRs) para proporcionar más direcciones IP de las necesarias para ejecutar sus cargas de trabajo. Si es necesario, también puede crear una dedicada VPC con un gran espacio de direcciones. Por ejemplo, puede crear una VPC con múltiples entradas CIDRs `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 EC2 consola de Amazon, pero no ve ninguna instancia contenedora de Amazon Elastic Container Service en su ECS clúster de Amazon, es posible que el ECS agente de Amazon no esté instalado en una Amazon Machine Image (AMI). Es posible que el Amazon ECS Agent, EC2 los datos de Amazon de tu AMI plantilla o de lanzamiento no estén configurados correctamente. Para aislar la causa raíz, crea una EC2 instancia de Amazon independiente o conéctate a una instancia existente medianteSSH. Para obtener más información, consulte la configuración del agente ECS contenedor de Amazon, las ubicaciones de los archivos de ECS registro de Amazon yRecurso informático AMIs.

- Revise el AWS panel de control: revise el AWS panel para comprobar el estado del trabajo esperado y que el entorno de cómputo se escala según lo esperado. También puede revisar los registros de trabajos CloudWatch.
- Compruebe que la instancia esté creada: si se crea una instancia, significa que su entorno de computación se ha escalado según lo previsto. Si sus instancias no se crean, busque las subredes asociadas en su entorno de computación 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 de computación 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 Auto Scaling para verificar que sus instancias estén siendo solicitadas por AWS Batch. Esto indica cómo Amazon EC2 intenta adquirir instancias. Si recibes un error que indica que Amazon EC2 Spot no puede adquirir una instancia en una zona de disponibilidad específica, puede deberse a que la zona de disponibilidad no ofrece una familia de instancias específica.
- Verifica que las instancias se registren en Amazon ECS: si ves instancias en la EC2 consola de Amazon, pero no hay instancias de ECS contenedor de Amazon en tu ECS clúster de Amazon, es posible que el ECS agente de Amazon no esté instalado en Amazon Machine Image (AMI). Además, es posible que el ECS agente de Amazon, EC2 los datos de Amazon de tu AMI plantilla o de lanzamiento no estén configurados correctamente. Para aislar la causa raíz, crea una EC2 instancia de Amazon independiente o conéctate a una instancia existente medianteSSH. Para obtener más información, consulte el archivo de configuración del CloudWatch agente: sección Logs, Ubicaciones de los archivos de ECS registro de Amazon yRecurso informático AMIs.
- 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.
- Revisa los HPC foros AWS Batch y: para obtener más información, consulta los HPCforos AWS Batchy.
- Revise el panel AWS Batch de supervisión del tiempo de ejecución: este panel utiliza una arquitectura sin servidor para capturar eventos de Amazon ECS y Amazon EC2 a fin de proporcionar información sobre los trabajos y las instancias. AWS Batch Para obtener más información, consulte Solución de paneles de monitoreo de tiempo de ejecución de AWS Batch.

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 IAM políticas, funciones y permisos, y se ejecuta en la infraestructura de Amazon EC2 ECS AWS Fargate, Amazon y Amazon Elastic Kubernetes Service. Para solucionar problemas relacionados con estos servicios, consulte lo siguiente:

- Solución de problemas IAM en la guía del usuario IAM
- ECSSolución de problemas de Amazon en la guía para desarrolladores de Amazon Elastic Container Service
- EKSSolución de problemas de Amazon en la Guía del EKS usuario de Amazon
- Solución de problemas de EC2 instancias en la Guía del EC2usuario de Amazon

Contenido

- AWS Batch
  - Entorno de computación INVALID
    - Nombre de función incorrecto o ARN
    - Repara un entorno INVALID informático
  - Trabajos bloqueados en estado RUNNABLE
  - Instancias de spot no etiquetadas en el momento de su creación
  - Las instancias de spot no se están reduciendo verticalmente
    - Adjunta la política EC2SpotFleetTaggingRole gestionada por Amazon a tu función de Spot Fleet en el AWS Management Console
    - Adjunta la política EC2SpotFleetTaggingRole gestionada por Amazon a tu puesto de Spot Fleet con el AWS CLI
  - No puedo recuperar los secretos de Secrets Manager
  - No se pueden anular los requisitos de recursos de la definición del trabajo
  - Mensaje de error al actualizar la configuración de desiredvCpus
- AWS Batch en Amazon EKS
  - Entorno de computación INVALID

    - No se admite Kubernetes versión
    - El perfil de instancia no existe
    - Invalid (No válido) Kubernetes namespace
    - Entorno de computación eliminado
    - Los nodos no se unen al EKS clúster de Amazon
- AWS Batch en Amazon, el EKS trabajo está RUNNABLE estancado
- Compruebe que aws-auth ConfigMap se ha configurado correctamente
- RBACLos permisos o los enlaces no están configurados correctamente

AWS Batch

Consulta los siguientes temas para encontrar los procesos de revisión y las posibles soluciones a los problemas comunes que puedas encontrar al AWS Batch utilizarlos.

Temas

- Entorno de computación INVALID
- Trabajos bloqueados en estado RUNNABLE
- Instancias de spot no etiquetadas en el momento de su creación
- Las instancias de spot no se están reduciendo verticalmente
- No puedo recuperar los secretos de Secrets Manager
- No se pueden anular los requisitos de recursos de la definición del trabajo
- Mensaje de error al actualizar la configuración de desiredvCpus

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 función incorrecto o ARN

La causa más común por la que un entorno de cómputo 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 de cómputo que se crean con AWS CLI o AWS SDKs. Cuando crea un entorno informático en AWS Management Console, le AWS Batch ayuda a elegir el servicio correcto o las funciones de Spot Fleet. Sin embargo, supongamos que introduce el nombre manualmente o ARN 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 IAM recurso en un AWS CLI comando o en su SDK código. 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 un IAM rol cuando se utiliza el AWS CLI o el AWS SDKs, en lugar del nombre completo del recurso de Amazon (ARN). Según cómo haya creado el rol, es ARN posible que contenga un prefijo de `aws-service-role` ruta. Por ejemplo, si crea manualmente el rol de AWS Batch servicio mediante los procedimientos descritos en[Utilice funciones vinculadas al servicio para AWS Batch](), el rol de servicio ARN podría tener el siguiente aspecto.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

Sin embargo, si hoy creó el rol de servicio como parte del asistente de primera ejecución de la consola, su rol de servicio ARN podría tener el siguiente aspecto.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Este problema también puede producirse si asocias 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 IAM rol cuando usas el AWS CLI o el AWS SDKs, se AWS Batch asume que ARN no usas el prefijo de `aws-service-role` ruta. Por este motivo, le recomendamos que especifique el valor completo ARN de sus IAM funciones al crear entornos de procesamiento.

Para reparar un entorno de computación configurado de forma incorrecta, consulte Repara un entorno `INVALID` informático.

Repara un entorno **`INVALID`** informático

Si el entorno de computación se encuentra en estado `INVALID`, actualícelo para reparar el parámetro no válido. Para un Nombre de función incorrecto o ARN, actualice el entorno de computación con el rol de servicio correcto.

Para reparar un entorno de computación mal configurado

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. 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 el entorno de cómputo, en Rol de servicio, elija el IAM rol que se va a usar con su entorno de cómputo. 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 bloquea la cola, recibirás un Recurso: Eventos bloqueados por la cola de trabajos evento de Amazon CloudWatch Events con el motivo. El mismo motivo también se actualiza en el `statusReason` campo como parte de `ListJobs` and `DescribeJobs` API calls.

Si lo desea, puede configurar el `jobStateTimeLimitActions` parámetro mediante `UpdateJobQueue`APIacciones `CreateJobQueue` y acciones.

> **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. En esta lista se muestran los mensajes devueltos por las `DescribeJobs`

API acciones `ListJobs` y. También son los mismos valores que puede definir para el `jobStateLimitActions.statusReason` parámetro.

1. Motivo: todos los entornos de cómputo conectados tienen errores de capacidad insuficientes. Cuando se solicita, AWS Batch detecta EC2 las instancias de Amazon que experimentan errores de capacidad insuficiente. Si se cancela el trabajo, ya sea manualmente o activando el `jobStateTimeLimitActions` parámetro`statusReason`, se permite que el siguiente trabajo pase al principio de la lista.
   - **`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 usa el rol Permisos de rol vinculados al servicio para AWS Batch vinculado al servicio (SLR) o la política AWS política gestionada: AWSBatchServiceRolepolítica administrada, no necesita realizar ninguna acción porque sus políticas de permisos están actualizadas.

   b. Si utilizas la política gestionada SLR o la política gestionada, debes añadir los `ec2:DescribeSpotFleetRequestHistory` permisos `autoscaling:DescribeScalingActivities` y de forma que puedas recibir los eventos de las colas de trabajos bloqueadas y el estado de los trabajos actualizados 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 parallel (MNP) de varios nodos, si el entorno de EC2 cómputo de Amazon de alta prioridad adjunto experimenta `insufficient capacity` errores, bloquea la cola incluso si un entorno de cómputo de menor prioridad experimenta este error.

2. Motivo: todos los entornos de cómputo 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 AWS políticas administradas para AWS Batch y solucione las brechas.

   Se recomienda utilizar el para Permisos de rol vinculados al servicio para AWS Batch 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 computación `INVALID`. 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 Why is my AWS Batch job stuck AWS in RUNNABLE 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 su recurso informático AMI en Amazon ECS Optimized AMI (o Amazon Linux). A continuación, este controlador se registra de forma predeterminada en el paquete `ecs-init`. Ahora supongamos que utiliza una base diferenteAMI. A continuación, debe comprobar que el controlador de `awslogs` registro esté especificado como un controlador de registro disponible con la variable de `ECS_AVAILABLE_LOGGING_DRIVERS` entorno cuando se inicie el

agente ECS contenedor de Amazon. Para obtener más información, consulte Especificación de recursos AMI informáticos y Tutorial: Crear un recurso de cómputo AMI.

Recursos insuficientes

Si las definiciones de sus tareas especifican más recursos CPU o recursos de memoria de los que pueden asignar sus recursos informáticos, sus tareas nunca se colocan. 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. Parte de la memoria está reservada para el agente ECS contenedor de Amazon y otros procesos críticos del sistema. Para obtener más información, consulte Administración de memoria de recursos informáticos.

No hay acceso a Internet para los recursos informáticos

Los recursos informáticos necesitan acceso para comunicarse con el punto final del ECS servicio de Amazon. Esto puede hacerse a través de un VPC punto final de interfaz o a través de los recursos de cómputo de que tienen direcciones IP públicas.

Para obtener más información sobre los VPC puntos de enlace de la interfaz, consulte Amazon ECS Interface VPC Endpoints (AWS PrivateLink) en la Guía para desarrolladores de Amazon Elastic Container Service.

Si no tiene configurado un VPC punto final de interfaz y los recursos informáticos de sus no tienen direcciones IP públicas, deberán utilizar la traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulta NATlas pasarelas en la Guía del VPC usuario de Amazon . Para obtener más información, consulte the section called “Tutorial: Crear un VPC”.

Se alcanzó el límite de EC2 instancias de Amazon

La cantidad de EC2 instancias de Amazon en las que puede lanzar tu cuenta Región de AWS viene determinada por tu cuota de EC2 instancias. Algunos tipos de instancias también tienen una per-instance-type cuota. Para obtener más información sobre la cuota de EC2 instancias de Amazon de tu cuenta, incluida la forma de solicitar un aumento del límite, consulta Amazon EC2 Service Limits en la Guía del EC2 usuario de Amazon.

El agente de ECS contenedores de Amazon no está instalado

El agente ECS contenedor de Amazon debe estar instalado en Amazon Machine Image (AMI) para poder AWS Batch ejecutar los trabajos. El agente de ECS contenedores de Amazon se

instala de forma predeterminada en Amazon ECS OptimizedAMIs. Para obtener más información sobre el agente de ECS contenedores de Amazon, consulte el agente de ECScontenedores de Amazon 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, la política IAM gestionada recomendada (`AmazonEC2SpotFleetRole`) para el rol Amazon EC2 Spot Fleet no incluía permisos para etiquetar instancias puntuales en el momento del lanzamiento. La nueva política IAM gestionada recomendada se denomina`AmazonEC2SpotFleetTaggingRole`. Admite el etiquetado de instancias de spot en el momento del lanzamiento.

Para corregir el etiquetado de las instancias puntuales al crearlas, siga el siguiente procedimiento para aplicar la política IAM gestionada recomendada actualmente a su función de Amazon EC2 Spot Fleet. 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 IAM gestionada actual a tu función de Amazon EC2 Spot Fleet

1. Abre la IAM consola en https://console.aws.amazon.com/iam/.
2. Elige roles y elige tu rol en Amazon EC2 Spot Fleet.
3. Elija Asociar política.
4. Selecciona Amazon EC2SpotFleetTaggingRole y elige Adjuntar política.
5. Vuelve a elegir tu rol en Amazon EC2 Spot Fleet para eliminar la política anterior.
6. Selecciona la x situada a la derecha de la EC2SpotFleetRole política de Amazon y selecciona 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 usa en un entorno de computación EC2 puntual, pero la función puntual utilizada no incluye la política EC2SpotFleetTaggingRole gestionada por Amazon. 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 un servicio y Creación de un rol para delegar permisos a un AWS servicio en la Guía del usuario. IAM

Temas

- Adjunta la política EC2SpotFleetTaggingRole gestionada por Amazon a tu función de Spot Fleet en el AWS Management Console
- Adjunta la política EC2SpotFleetTaggingRole gestionada por Amazon a tu puesto de Spot Fleet con el AWS CLI

Adjunta la política EC2SpotFleetTaggingRole gestionada por Amazon a tu función de Spot Fleet en el AWS Management Console

Para aplicar la política IAM gestionada actual a tu función de Amazon EC2 Spot Fleet

1. Abre la IAM consola en https://console.aws.amazon.com/iam/.
2. Elige roles y elige tu rol en Amazon EC2 Spot Fleet.
3. Elija Asociar política.
4. Selecciona Amazon EC2SpotFleetTaggingRole y elige Adjuntar política.
5. Vuelve a elegir tu rol en Amazon EC2 Spot Fleet para eliminar la política anterior.
6. Selecciona la x situada a la derecha de la EC2SpotFleetRole política de Amazon y selecciona Separar.

Adjunta la política EC2SpotFleetTaggingRole gestionada por Amazon a tu puesto de Spot Fleet con el AWS CLI

Los comandos de ejemplo asumen que su rol en Amazon EC2 Spot Fleet se denomina *`AmazonEC2SpotFleetRole`*. Si su función usa un nombre diferente, ajuste los comandos para que coincidan.

Para adjuntar la política EC2SpotFleetTaggingRole gestionada por Amazon a su función de Spot Fleet

1. Para adjuntar la IAM política EC2SpotFleetTaggingRole gestionada por Amazon a tu *AmazonEC2SpotFleetRole* rol, ejecuta el siguiente comando con AWS CLI.

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole
 \
    --role-name AmazonEC2SpotFleetRole
```

2. Para separar la IAM política EC2SpotFleetRole gestionada por Amazon de tu *AmazonEC2SpotFleetRole* rol, ejecuta el siguiente comando con. 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 utilizas una versión anterior a la 1.16.0-1 AMI con un ECS agente de Amazon, debes usar la variable de configuración del ECS agente de Amazon `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` para usar esta función. 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 añades a una instancia existente, debes reiniciar el ECS agente después de añadirlo. Para obtener más información, consulte Configuración de Amazon ECS Container Agent en la Guía para desarrolladores de Amazon Elastic Container Service.

No se pueden anular los requisitos de recursos de la definición del trabajo

La memoria y las CPU anulaciones v que se especifican en los `vcpus` miembros `memory` y de la containerOverridesestructura, que se pasan a SubmitJob, no pueden anular los CPU requisitos de memoria y v que se especifican en la resourceRequirementsestructura 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 CPU requisitos de memoria y v en el resourceRequirementselemento de. containerOverrides Por ejemplo, si las CPU anulaciones de memoria y v se especifican en las siguientes líneas.

```
"containerOverrides": {
   "memory": 8192,
   "vcpus": 4
}
```

Cámbielos a lo siguiente:

```
"containerOverrides": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "8192"
      },
      {
         "type": "VCPU",
         "value": "4"
      }
   ],
}
```

Realice el mismo cambio en los CPU requisitos de memoria y v que se especifican en el containerPropertiesobjeto en la definición del trabajo. Por ejemplo, si sus CPU requisitos de memoria y v 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 utiliza el AWS Batch API para actualizar la configuración vCPUs (`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
- AWS Batch en Amazon, el EKS trabajo está RUNNABLE estancado
- Compruebe que aws-auth ConfigMap se ha configurado correctamente
- RBACLos permisos o los enlaces no están configurados correctamente

Consulte los siguientes temas para conocer los procesos de revisión y las posibles soluciones a los problemas comunes que pueden surgir al utilizar AWS Batch Amazon Elastic Kubernetes Service.

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.

No se admite Kubernetes versión

Es posible que aparezca un mensaje de error similar al siguiente cuando utilice la `CreateComputeEnvironment` API operación o la `UpdateComputeEnvironment` API operación para crear o actualizar un entorno informático. Este problema se produce si especificas un elemento no compatible Kubernetes versión en`EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Para resolver este problema, elimine el entorno informático y, a continuación, vuelva a crearlo con un entorno compatible Kubernetes versión.

Puedes realizar una actualización de versión menor en tu EKS clúster de Amazon. 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 compatible Kubernetes versión cuando se utiliza una API operación para crear o actualizar un entorno informático.

AWS Batch en Amazon EKS actualmente admite lo siguiente Kubernetes versiones:

- `1.30`
- `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, AWS Batch el estado del entorno de EKS cómputo de Amazon 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 el IAMrol de EKS nodo de Amazon en la Guía del EKS usuario de Amazon.

Invalid (No válido) Kubernetes namespace

Si AWS Batch Amazon no EKS 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 Kubernetes la cadena de espacio de nombres de la `CreateComputeEnvironment` llamada 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 a Amazon EKS Kubernetes APIpunto final del servidor.

Para resolver este problema, consulte Compruebe que `aws-auth ConfigMap` se ha configurado correctamente. Para obtener más información, consulte Tutorial: Cómo empezar AWS Batch en Amazon EKS.

Entorno de computación eliminado

Suponga que elimina un EKS clúster de Amazon antes de eliminar el entorno EKS informático adjunto AWS Batch en Amazon. A continuación, el estado del entorno de computación cambia a `INVALID`. En este escenario, el entorno informático no funciona correctamente si vuelves a crear el EKS clúster de Amazon con el mismo nombre.

Para resolver este problema, elimine y, a continuación, vuelva a crear AWS Batch el entorno EKS informático de Amazon.

Los nodos no se unen al EKS clúster de Amazon

AWS Batch on Amazon reduce la EKS escala de un entorno de cómputo si determina que no todos los nodos se han unido al EKS clúster de Amazon. Cuando AWS Batch Amazon reduce la EKS escala del entorno de cómputo, el estado del entorno de cómputo cambia a`INVALID`.

> **Note**
>
> AWS Batch no cambia el estado del entorno de cómputo de forma inmediata para que puedas 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 un Amazon predeterminado EKSAMI, las causas más comunes de este problema son las siguientes:

- El rol de la instancia no está configurado correctamente. Para obtener más información, consulte el IAMrol de EKS nodo de Amazon en la Guía del EKS usuario de Amazon.
- Las subredes no están configuradas correctamente. Para obtener más información, consulta los requisitos EKS VPC y consideraciones de Amazon y las subredes en la Guía del EKS usuario de Amazon.
- El grupo de seguridad no está configurado correctamente. Para obtener más información, consulta los requisitos y consideraciones de los grupos de EKS seguridad de Amazon en la Guía del EKS usuario de Amazon.

> **Note**
>
> También puede ver una notificación de error en el Personal Health Dashboard (PHD).

AWS Batch en Amazon, el EKS 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 añadir el control de acceso basado en roles (RBAC) para acceder 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 puedes revisar los registros del plano EKS de control de Amazon. Para obtener más información, consulta el registro del plano EKS de control de Amazon en la Guía del EKS usuario de Amazon.

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 Prepara tu EKS clúster de Amazon para AWS Batch. O bien, puede usar `kubectl` para editar manualmente el `aws-authConfigMap`. Para obtener más información, consulta Cómo habilitar IAM el acceso de usuarios y roles a tu clúster en la Guía del EKS usuario de Amazon.

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**
>
> Se `aws-service-role/batch.amazonaws.com/` ha eliminado la ruta ARN del rol vinculado al 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 sus ARN aws-authconfigmap.

> **Note**
>
> También puedes revisar los registros del plano EKS de control de Amazon. Para obtener más información, consulta el registro del plano EKS de control de Amazon en la Guía del EKS usuario de Amazon.

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 Prepara tu EKS clúster de Amazon para AWS Batch. O bien, puede usar `kubectl` para editar manualmente el `aws-authConfigMap`. Para obtener más información, consulta Cómo habilitar IAM el acceso de usuarios y roles a tu clúster en la Guía del EKS usuario de Amazon.

RBAClos permisos o los enlaces no están configurados correctamente

Si tiene algún problema con RBAC los permisos o los enlaces, compruebe que `aws-batch` Kubernetes el rol puede acceder al Kubernetes espacio de nombres:

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

También puede usar el **`kubectl describe`** comando para ver las autorizaciones de un rol de clúster o Kubernetes espacio de nombres.

```
$ 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 RBAC permisos y comandos. `rolebinding` Para obtener más información, consulte Prepara tu EKS clúster de Amazon para AWS Batch.

Recurso: cuotas AWS Batch de servicio

En la siguiente tabla se muestran las cuotas de servicio AWS Batch que no se pueden cambiar. 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 de cómputo en Amazon ECS y AmazonEKS. Para obtener más información, consulte Entornos informáticos para AWS Batch.	50
Número máximo de entornos de cómputo por EKS clúster de Amazon.	5
Número máximo de entornos de computación para cada cola de trabajos	3
Cantidad máxima de dependencias de un trabajo	20
Tamaño máximo de definición de trabajo (para `RegisterJobDefinition` APIoperaciones)	24 KiB
Tamaño máximo de carga útil del trabajo (para `SubmitJob` APIoperaciones)	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

En función del uso que se utilice AWS Batch, es posible que se apliquen cuotas adicionales. Para obtener más información sobre EC2 las cuotas de Amazon, consulte Amazon EC2 Service Quotas en Referencia general de AWS. Para obtener más información sobre ECS las cuotas de Amazon, consulte Amazon ECS Service Quotas en Referencia general de AWS. Para obtener más información sobre EKS las cuotas de Amazon, consulte Amazon EKS Service Quotas en Referencia general de AWS.

Historial de documentos

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
EKSVersiones de Amazon AWS Batch compatibles actualizadas	Se actualizaron las EKS versiones de Amazon AWS Batch compatibles para eliminar la versión 1.22.	11 de marzo de 2024
EKSVersiones de Amazon AWS Batch compatibles actualizadas	Se actualizaron las EKS versiones de Amazon 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
EKSVersiones de Amazon AWS Batch compatibles actualizadas	Se actualizaron las EKS versiones de Amazon 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 una declaraciónIDs, conceder AWS Batch permisos `ec2:DescribeSpotFleetRequestHistory` y `autoscaling:DescribeScalingActivities` .	
AWS Batch en Amazon EKS	AWS Batch añade soporte para ejecutar trabajos en EKS clústeres de Amazon.	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 finales de la interfaz VPC ()AWS PrivateLink	Se agregó soporte para configurar los VPC puntos finales de la interfaz con la tecnología de. AWS PrivateLink Esto significa que puede crear una conexión privada entre su VPC y AWS Batch sin necesidad de acceder a través de una NAT instancia , una VPN conexión 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 existentes.	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 soporte para añadir los sistemas de EFS archivos de Amazon a tus definiciones de trabajo.	1 de abril de 2021
Rol vinculado al servicio añadido	AWS Batch añade el rol AWSServiceRoleForBatchvinculado 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

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
EFAsoporte	AWS Batch añade compatibilidad con los dispositivos Elastic Fabric Adapter (EFA).	2 de agosto de 2019
GPUprogramación	AWS Batch añade GPU programación. Con esta función, puede especificar la cantidad que requiere GPUs cada trabajo y ampliar AWS Batch las instancias en consecuencia.	4 de abril de 2019

Trabajos paralelos de varios nodos	AWS Batch añade soporte para trabajos paralelos de varios nodos. Puedes usar esta función para ejecutar trabajos individuales que abarquen varias EC2 instancia s de Amazon.	19 de noviembre de 2018
Permisos de nivel de recursos	AWS Batch admite permisos a nivel de recursos en varias API operaciones.	12 de noviembre de 2018
Soporte de plantillas EC2 de Amazon Launch	AWS Batch añade soporte para el uso de plantillas de lanzamiento con entornos de cómputo.	12 de noviembre de 2018
AWS Batch tiempos de espera de los trabajos	AWS Batch añade compatibi lidad 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 a la AWS Batch API acción realizadas.	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 etiquetad o. Puede utilizar esta función para especificar etiquetas para las instancias EC2 puntuales de Amazon lanzadas en entornos informáti cos gestionados.	26 de octubre de 2017
AWS Batch transmisión 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 compatibilidad con los 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.