



Guía para desarrolladores

Amazon Simple Workflow Service



Versión de API 2012-01-25

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Simple Workflow Service: Guía para desarrolladores

Copyright © 2026 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 Amazon SWF?	1
Componentes del flujo de trabajo	2
Componentes del flujo de trabajo	2
Ejecutando su flujo de trabajo	4
Configuración del entorno de desarrollo de	5
Desarrolle con AWS SDKs	5
Tenga en cuenta las AWS Flow Framework	6
Introducción	7
Acerca del flujo de trabajo	8
Requisitos previos	9
Pasos del tutorial	9
Parte 1: Uso de Amazon SWF con el SDK para Ruby	9
Incluya el AWS SDK para Ruby	10
Configuración de la sesión AWS	10
Registro de un dominio de Amazon SWF	12
Sigüientes pasos	13
Segunda parte: implementación del flujo de trabajo	13
Diseño del flujo de trabajo	13
Configuración del código del flujo de trabajo	14
Registro de un flujo de trabajo	16
Sondeo de decisiones	17
Comienzo de la ejecución del flujo de trabajo	20
Sigüientes pasos	23
Parte 3: implementación de las actividades	23
Definición de un tipo de actividad básica	23
Definiendo GetContactActivity	25
Definir SubscribeTopicActivity	27
¿Definiendo WaitForConfirmationActivity	31
¿Definiendo SendResultActivity	33
Sigüientes pasos	35
Parte 4: implementación del sondeador de tareas de actividades	35
Ejecución del flujo de trabajo	38
¿Qué tengo que hacer ahora?	43
Trabajando en la consola	44

Registrar un dominio	44
Registrar tipos de flujo de trabajo	45
Registrar los tipos de actividad	46
Iniciar un flujo de trabajo	47
Para comenzar una ejecución de flujo de trabajo con la consola	47
Gestión de las ejecuciones del flujo de	47
Conceptos básicos	51
Creación de un flujo de trabajo	52
Modelado del flujo de trabajo y sus actividades	53
Ejecutar flujos de trabajo	54
Historial del flujo de trabajo	54
Identificadores de objetos	59
Dominios	60
Actores	61
¿Qué es un actor en Amazon SWF?	61
Iniciadores de flujo de trabajo	62
Decisores	62
Procesos de trabajo de actividad	64
Intercambio de datos entre actores	64
Tareas	65
Listas de tareas	66
Listas de tareas de decisión	67
Listas de tareas de actividad	67
Direccionamiento de tareas	68
Cierre de ejecución del flujo de trabajo	68
Ciclo de vida de la ejecución del flujo	69
Ciclo de vida de ejecución de flujo de trabajo	70
Sondeo de tareas	76
Conceptos avanzados	78
Control de versiones	78
Señales	79
Flujos de trabajo secundarios	81
Marcadores	83
Etiquetas	85
Administración de etiquetas	85
Etiqueta las ejecuciones del flujo de trabajo	86

Controle el acceso a los dominios con etiquetas	87
Elección exclusiva	88
Temporizadores	91
Cancelar tareas de actividad	91
Seguridad	95
Protección de los datos	95
Cifrado	96
Gestión de identidad y acceso	97
Público	98
Autenticación con identidades	98
Administración del acceso con políticas	100
Control de acceso	101
Acciones de política	102
Recursos de políticas	102
Claves de condición de políticas	103
ACLs	103
ABAC	104
Credenciales temporales	104
Permisos de entidades principales	104
Roles de servicio	105
Roles vinculados a servicios	105
Políticas basadas en identidad	105
Políticas basadas en recursos	106
Cómo funciona Amazon Simple Workflow Service con IAM	106
Ejemplos de políticas basadas en identidades	107
Principios básicos	110
Políticas de IAM para Amazon SWF	112
Resumen de las API	118
Políticas basadas en etiquetas	126
Puntos de conexión de VPC de Amazon	127
Resolución de problemas	129
Registro y supervisión	131
Métricas de Amazon SWF para CloudWatch	131
Visualización de métricas de Amazon SWF	142
Grabando en CloudTrail	146
EventBridge para Amazon SWF	154

Uso AWS User Notifications con Amazon SWF	162
Validación de la conformidad	163
Resiliencia	164
Seguridad de infraestructuras	164
Configuración y análisis de vulnerabilidades	165
Usando el AWS CLI	166
Trabajando con APIs	168
Realización de solicitudes HTTP	168
Contenido de los encabezados HTTP	169
Contenido de cuerpo HTTP	171
Ejemplo de solicitud y respuesta JSON	171
Cálculo de la firma HMAC-SHA	173
Lista de acciones de Amazon SWF	175
Acciones relacionadas con actividades	175
Acciones relacionadas con decisores	176
Acciones relacionadas con ejecuciones de flujos de trabajo	176
Acciones relacionadas con administración	177
Acciones de visibilidad	178
Registro de un dominio	179
Véase también	179
Establecer valores de tiempo de espera	179
Cuotas de los valores de tiempo de espera	180
Tiempos de espera de las ejecuciones de flujos de trabajo y las tareas de decisión	180
Tiempos de espera de las tareas de actividad	181
Véase también	181
Registro de un tipo de flujo de trabajo	181
Véase también	182
Registro de un tipo de actividad	182
Véase también	183
Tareas Lambda	183
Acerca AWS Lambda	183
Beneficios y limitaciones de la utilización de tareas de Lambda	184
Uso de tareas de Lambda en los flujos de trabajo	184
Desarrollo de un proceso de trabajo de actividad	189
Sondeo de tareas de actividad	190
Realización de la tarea de actividad	191

Notificación de los latidos de tareas de actividad	191
Finalización de una tarea de actividad o generación de un error en la misma	192
Lanzamiento de procesos de trabajo de actividad	194
Desarrollo de los decisores	194
Definición de la lógica de coordinación	196
Sondeo de tareas de decisión	196
Aplicación de la lógica de coordinación	198
Respuesta con decisiones	199
Cierre de una ejecución de flujo de trabajo	200
Lanzamiento de decisores	201
Inicio de flujos de trabajo	202
Establecer la prioridad de las tareas	203
Establecimiento de prioridad de las tareas para flujos de trabajo	204
Establecimiento de prioridad de las tareas para actividades	206
Acciones que devuelven información acerca de la prioridad de las tareas	207
Gestión de errores	208
Errores de validación	208
Errores en la aplicación de acciones o decisiones	209
Tiempos de espera	209
Errores planteados el código de usuario	210
Errores relacionados con el cierre de una ejecución de flujo de trabajo	210
Cuotas	211
Cuotas de cuentas generales para Amazon SWF	211
Cuotas de ejecución de flujos de trabajo	212
Cuotas de ejecuciones de tareas	213
Cuotas de limitación de Amazon SWF	214
Cuotas de limitación para todas las regiones	215
Cuotas de decisión para todas las regiones	217
Cuotas a nivel de los flujos de trabajo	217
Solicitud de aumento de cuota	218
Recursos adicionales	219
Tipos de tiempo de espera	219
Tiempos de espera de las tareas de decisión y flujo de trabajo	220
Tiempos de espera de las tareas de actividad	221
puntos de conexión	222
Documentación adicional de	223

Referencia de la API de Amazon Simple Workflow Service	223
AWS Flow Framework Documentación	223
AWS Documentación del SDK	223
AWS CLI Documentación	225
Recursos web	226
Foro de Amazon SWF	226
Preguntas frecuentes sobre Amazon SWF	226
Vídeos de Amazon SWF	226
Opciones de Ruby Flow	226
Seguir usando el Ruby Flow Framework	227
Migrar al Java Flow Framework	227
Migrar a Step Functions	228
Utilizar la API de Amazon SWF directamente	229
Historial de documentos	230
.....	CCXXXV

¿Qué es Amazon Simple Workflow Service?

Con Amazon Simple Workflow Service (Amazon SWF), puede crear, ejecutar y escalar trabajos en segundo plano con pasos paralelos o secuenciales. Puede coordinar el trabajo entre los componentes distribuidos y realizar un seguimiento del estado de las tareas.

En Amazon SWF, una tarea representa una unidad lógica de trabajo que realiza un componente de la aplicación. La coordinación de todas las tareas incluye la gestión de las dependencias entre tareas, la programación y la simultaneidad en el flujo de la aplicación. Con Amazon SWF, puede controlar y coordinar las tareas sin preocuparse por las complejidades subyacentes, como el seguimiento del progreso y el mantenimiento del estado de las tareas.

Cuando se utiliza Amazon SWF, se implementan trabajadores para que realicen tareas. Los trabajadores pueden trabajar en una infraestructura de nube, como Amazon Elastic Compute Cloud (Amazon EC2), o en sus propias instalaciones. Se pueden crear tareas que se ejecuten de forma prolongada o que puedan provocar errores, agotar el tiempo de espera o requerir reinicios, o bien que puedan completarse con un rendimiento y una latencia variables. Amazon SWF almacena las tareas y las asigna a los trabajadores cuando están preparadas, realiza un seguimiento del progreso y mantiene el estado, incluidos los detalles de la finalización de las tareas.

Para coordinar las tareas, debe escribir un programa que obtiene el estado más reciente de la tarea de Amazon SWF y lo utiliza para iniciar las tareas subsiguientes. Amazon SWF mantiene el estado de ejecución de una aplicación de forma duradera, por lo que la aplicación es resistente a los fallos de los componentes individuales. Con Amazon SWF, puede crear, implementar, escalar y modificar los componentes de la aplicación de forma independiente.

Otros servicios de AWS flujo de trabajo

Para la mayoría de los casos de uso, te recomendamos que los tengas en cuenta AWS Step Functions para tus necesidades de flujo de trabajo y orquestación.

Con Step Functions, puede crear flujos de trabajo, también denominados máquinas de estado, para crear aplicaciones distribuidas, automatizar procesos, organizar microservicios y crear canalizaciones de datos y aprendizaje automático. En la consola o el AWS kit de herramientas de Step Functions en VS Code, puede usar el Workflow Studio gráfico para visualizar, editar, probar y depurar el flujo de trabajo de su aplicación.

[Para obtener más información técnica, consulte la AWS Step Functions Guía para desarrolladores.](#)

Desarrollo de componentes de flujo de trabajo con Amazon SWF

El desarrollo de aplicaciones distribuidas requiere coordinar muchos componentes y hacer frente a la latencia y la falta de fiabilidad inherentes a la comunicación remota.

Con Amazon Simple Workflow Service (Amazon SWF), puede desarrollar aplicaciones distribuidas y asíncronas proporcionando un modelo de programación y una infraestructura para coordinar los componentes distribuidos y mantener su estado de ejecución de forma fiable. Al confiar en Amazon SWF, puede concentrarse en la creación de los aspectos de la aplicación que marcarán la diferencia respecto a otras aplicaciones.

Componentes de un flujo de trabajo

[Componentes de un flujo de trabajo](#) El concepto fundamental de Amazon SWF es el flujo de trabajo. Un flujo de trabajo es un conjunto de actividades que se realizan para obtener un objetivo determinado, junto con la lógica que coordina las actividades. Por ejemplo, un flujo de trabajo podría recibir un pedido de un cliente y tomar las medidas necesarias para tramitar el pedido.

Cada flujo de trabajo se ejecuta en un recurso denominado dominio, que controla el alcance del flujo de trabajo. Una cuenta de AWS puede tener varios dominios, y cada uno de ellos puede contener numerosos flujos de trabajo. No obstante, los flujos de trabajo de distintos dominios no pueden interactuar entre sí.

Al diseñar un flujo de trabajo de Amazon SWF, debe definir cada una de las actividades necesarias. A continuación, deberá registrar cada actividad con Amazon SWF como un tipo de actividad. Deberá proporcionar un nombre, una versión y valores de tiempo de espera. Por ejemplo, un cliente puede esperar que un pedido se envíe en un plazo de 24 horas.

En el proceso de llevar a cabo el flujo de trabajo, puede ser necesario realizar algunas actividades más de una vez, quizá con distintas entradas. Por ejemplo, en un flujo de trabajo de pedido de un cliente, podría tener una actividad que se encargue de los artículos adquiridos. Si el cliente adquiere varios artículos, esta actividad tendría que ejecutarse varias veces. Amazon SWF tiene el concepto de una tarea de actividad que representa una invocación de una actividad. En nuestro ejemplo, el procesamiento de cada artículo estaría representado por una sola tarea de actividad.

Un trabajador de actividades es un programa que recibe las tareas de una actividad, las ejecuta y proporciona resultados. En realidad, la tarea podría realizarla una persona. Por ejemplo, un analista estadístico puede recibir conjuntos de datos, analizarlos y, a continuación, enviar su análisis.

Las tareas de actividad y los trabajadores de la actividad que las realizan pueden ejecutarse de forma sincrónica o asíncrona. Los trabajadores pueden trabajar en un solo lugar o estar distribuidos en varios ordenadores, posiblemente en diferentes regiones geográficas. Los diferentes procesos de trabajo de actividad pueden escribirse en distintos lenguajes de programación y ejecutarse en distintos sistemas operativos. Por ejemplo, un trabajador de la actividad podría estar ejecutándose en un servidor en Asia, mientras que otro podría estar ejecutándose en un dispositivo móvil en América del Norte.

La lógica de coordinación en un flujo de trabajo está contenida en un programa de software llamado decisor. Un decisor programa las tareas de una actividad, proporciona información a los trabajadores activos, procesa los eventos que llegan mientras el flujo de trabajo está en curso y finaliza (o cierra) el flujo de trabajo una vez alcanzado el objetivo.

La función del servicio de Amazon SWF es servir de ubicación central fiable a través de la que se intercambien datos entre el decisor, los procesos de trabajo de la actividad y otras entidades relevantes, como la persona que administra el flujo de trabajo. Amazon SWF también mantiene el estado de ejecución de cada flujo de trabajo, lo que evita que la aplicación tenga que almacenar el estado de forma duradera.

El decisor dirige el flujo de trabajo cuando recibe las tareas de decisión de Amazon SWF y responde a Amazon SWF con decisiones. Una decisión representa una acción o un conjunto de acciones, que son los siguientes pasos del flujo de trabajo. Una decisión típica sería programar una tarea de actividad. Las decisiones también se pueden utilizar para retrasar tareas con temporizadores, solicitar la cancelación de tareas en curso y completar los flujos de trabajo.

El mecanismo por el que los procesos de trabajo de una actividad y del decisor reciben sus tareas (tareas de actividad y tareas de decisión, respectivamente) consiste en sondear el servicio de Amazon SWF.

Amazon SWF informa al decisor del estado del flujo de trabajo e incluye una copia del historial de ejecución del flujo de trabajo actual con cada tarea de decisión. El historial de ejecución del flujo de trabajo está compuesto de eventos, donde un evento representa un cambio significativo en el estado de la ejecución del flujo de trabajo. Algunos ejemplos de eventos son la finalización de una tarea, los tiempos de espera de una tarea o la caducidad de un temporizador. El historial es un registro completo, coherente y fiable del progreso del flujo de trabajo.

El control de acceso de Amazon SWF utiliza AWS Identity and Access Management (IAM) para que pueda controlar el acceso a los recursos. AWS Por ejemplo, puede permitir que un usuario acceda a su cuenta, pero solo para ejecutar ciertos flujos de trabajo en un dominio concreto.

Ejecutando su flujo de trabajo

A continuación, se proporciona una descripción general de los pasos necesarios para desarrollar y ejecutar un flujo de trabajo en Amazon SWF:

1. Escriba a los trabajadores de actividades para que realicen los pasos de procesamiento de su flujo de trabajo.
2. Escribe un decisor para gestionar la lógica de coordinación de tu flujo de trabajo.
3. Registre las actividades y el flujo de trabajo en Amazon SWF.

Puede realizar este paso mediante programación o mediante el. Consola de administración de AWS

4. Inicie los procesos de trabajo de actividad y el decisor.

Estos actores pueden ejecutarse en cualquier dispositivo informático que pueda obtener acceso a un punto de conexión de Amazon SWF. Por ejemplo, podrías usar instancias de cómputo en la nube, como Amazon Elastic Compute Cloud (Amazon EC2); servidores en tu centro de datos; o incluso un dispositivo móvil, para alojar a un decisor o a un trabajador de la actividad. Una vez iniciados, el decisor y los procesos de trabajo de actividad deben comenzar a sondear si hay tareas en Amazon SWF.

5. Comience una o varias ejecuciones de su flujo de trabajo.

Puede iniciar los flujos de trabajo mediante programación o mediante el. Consola de administración de AWS

Cada ejecución se ejecuta independientemente y puede proporcionar su propio conjunto de datos de entrada a cada una. Cuando se inicia una ejecución, Amazon SWF programa la tarea de decisión inicial. En respuesta, la persona que decide comienza a generar decisiones que inician las tareas de actividad. La ejecución continúa hasta que el decisor toma la decisión de cerrar la ejecución.

6. Vea las ejecuciones del flujo de trabajo mediante. Consola de administración de AWS

Puede filtrar y ver los detalles completos de las ejecuciones en ejecución y finalizadas. Por ejemplo, puede seleccionar una ejecución abierta para ver qué tareas se han completado y cuáles han sido sus resultados.

Configuración del entorno de desarrollo de

Tiene la opción de desarrollar para Amazon SWF en cualquiera de los lenguajes de programación compatibles con. AWS Para los desarrolladores de Java, también AWS Flow Framework está disponible. Para obtener más información, consulte el [AWS Flow Framework](#) sitio web y la [Guía AWS Flow Framework para desarrolladores de Java](#).

Para reducir la latencia y almacenar los datos en una ubicación que cumpla sus requisitos, Amazon SWF proporciona puntos de enlace en diferentes regiones.

Cada punto de conexión de Amazon SWF es completamente independiente. Los dominios, flujos de trabajo y actividades que haya registrado en una región no compartirán datos ni atributos con los de otra región.

Cuando registras un dominio, flujo de trabajo o actividad de Amazon SWF, solo existe en la región en la que lo registraste. Por ejemplo, puede registrar un dominio con un nombre SWF-Flows-1 en dos regiones diferentes, pero no compartirán datos ni atributos entre sí, ya que cada una de ellas actúa como un dominio completamente independiente.

Para ver una lista de los puntos de conexión de Amazon SWF, consulte [Regiones y puntos de conexión](#).

Desarrolle con AWS SDKs

Amazon SWF es compatible con Java, .NET, Node.js, PHP, Python y Ruby, lo que proporciona una forma cómoda de utilizar la API HTTP de Amazon SWF en el lenguaje de programación que prefiera. AWS SDKs

Puede desarrollar decisores, trabajadores activos o iniciadores de flujos de trabajo mediante la API expuesta en estas bibliotecas. Además, puede utilizar las operaciones de visibilidad a través de estas bibliotecas para desarrollar sus propias herramientas de supervisión e informes de Amazon SWF.

Para descargar herramientas para desarrollar y administrar aplicaciones AWS, incluso SDKs, vaya al [Centro de desarrolladores](#).

Para obtener información detallada sobre las operaciones de Amazon SWF en cada SDK, consulte la documentación de referencia específica del idioma del SDK.

Tenga en cuenta las AWS Flow Framework

AWS Flow Framework Se trata de un SDK mejorado para escribir programas distribuidos y asíncronos que se ejecutan como flujos de trabajo en Amazon SWF. El marco está disponible para el lenguaje de programación Java y ofrece clases para escribir programas distribuidos complejos.

Con él AWS Flow Framework, utiliza tipos preconfigurados para asignar la definición del flujo de trabajo directamente a los métodos del programa. AWS Flow Framework Es compatible con conceptos estándar orientados a objetos, como el manejo de errores basado en excepciones. Los programas escritos con el se AWS Flow Framework pueden crear, ejecutar y depurar completamente en el editor o IDE que prefiera. Para obtener más información, consulte el [AWS Flow Framework](#) sitio web y la [Guía AWS Flow Framework para desarrolladores de Java](#).

Introducción a Amazon SWF

Puede empezar con la siguiente aplicación de flujo de trabajo de Amazon Simple Workflow Service, que consta de un conjunto de cuatro actividades que funcionan de forma secuencial. El tutorial también cubre los siguientes temas:

- Establecimiento de las opciones de actividad y flujo de trabajo default y execution-time.
- Sondeo de Amazon SWF para tareas de actividad y de decisión.
- Transmisión de datos entre las actividades y el flujo de trabajo con Amazon SWF.
- Espera de tareas humanas y notificación de latidos a Amazon SWF desde una tarea de actividad.
- Uso de Amazon SNS para crear un tema, suscribir un usuario a este y publicar mensajes en puntos de conexión suscritos.

Puede utilizar Amazon SWF y Amazon Simple Notification Service (Amazon SNS) juntos para emular un flujo de trabajo de «tarea humana», en el que un trabajador humano debe realizar alguna acción y, a continuación, comunicarse con Amazon SWF para lanzar la siguiente actividad del flujo de trabajo.

Dado que Amazon SWF es un servicio web basado en la nube, la comunicación con Amazon SWF puede originarse en cualquier lugar donde haya conexión a Internet. En este caso, usaremos Amazon SNS para comunicarnos con el usuario por correo electrónico, mediante un mensaje de texto SMS o de ambas formas.

En este tutorial se utiliza [AWS SDK para Ruby](#) para acceder a Amazon SWF y Amazon SNS, pero hay muchas opciones de desarrollo disponibles, incluida AWS Flow Framework la de Ruby, que facilita la coordinación y la comunicación con Amazon SWF.

Note

[En este tutorial se utiliza el AWS SDK para Ruby, pero le recomendamos que lo utilice para Java.AWS Flow Framework](#)

Temas

- [Acerca del flujo de trabajo](#)

- [Requisitos previos](#)
- [Pasos del tutorial](#)
- [Tutorial sobre el flujo de trabajo de suscripciones, parte 1: Uso de Amazon SWF con AWS SDK para Ruby](#)
- [Segunda parte del tutorial acerca del flujo de trabajo de suscripción: implementación del flujo de trabajo](#)
- [Parte 3 del tutorial acerca del flujo de trabajo de suscripción: implementación de las actividades](#)
- [Parte 4 del tutorial acerca del flujo de trabajo de suscripción: implementación del sondeador de tareas de actividades](#)
- [Tutorial acerca del flujo de trabajo de suscripción: ejecución del flujo de trabajo](#)

Acerca del flujo de trabajo

El flujo de trabajo que desarrollaremos consta de cuatro pasos principales:

1. Obtenga una dirección de suscripción (correo electrónico o SMS) del usuario.
2. Cree un tema de SNS y suscriba los puntos de conexión proporcionados al tema.
3. Espere a que el usuario confirme la suscripción.
4. Si el usuario la confirma, publique un mensaje de felicitación en el tema.

Entre estos pasos se incluyen actividades que están totalmente automatizadas (pasos 2 y 4) y otras que requieren que el flujo de trabajo espere a que un humano proporcione algunos datos a la actividad antes de que el flujo de trabajo pueda progresar (pasos 1 y 3).

Cada paso se basa en datos generados por el paso anterior (debe tener un punto de conexión antes de suscribirlo a un tema y debe tener una suscripción a un tema antes de poder esperar la confirmación, etc.). En este tutorial también se explicará cómo proporcionar los resultados de una actividad una vez finalizada y cómo transferir información a una tarea que se está programando. Amazon SWF gestiona la coordinación y la entrega de información entre las actividades y el flujo de trabajo, y viceversa.

Asimismo, usamos la entrada de teclado y Amazon SNS para controlar la comunicación entre Amazon SWF y la persona que proporciona datos al flujo de trabajo. En la práctica, puede utilizar muchas técnicas diferentes para comunicarse con usuarios humanos, pero Amazon SNS

proporciona una forma muy sencilla de utilizar mensajes de texto o correo electrónico para notificar al usuario acerca de eventos en el flujo de trabajo.

Requisitos previos

Para continuar con este tutorial, necesitará lo siguiente:

- [Cuenta de Amazon Web Services](#)
- [Intérprete de Ruby](#)
- [AWS SDK para Ruby](#)

Si ya tiene estos elementos configurados, estará listo para continuar. Si no quiere ejecutar el ejemplo, puede seguir el tutorial; gran parte del contenido de este tutorial se refiere al uso de Amazon SWF y Amazon SNS, independientemente de la opción de desarrollo que elija.

Pasos del tutorial

Este tutorial se divide en los siguientes pasos:

1. [Tutorial sobre el flujo de trabajo de suscripciones, parte 1: Uso de Amazon SWF con AWS SDK para Ruby](#)
2. [Segunda parte del tutorial acerca del flujo de trabajo de suscripción: implementación del flujo de trabajo](#)
3. [Parte 3 del tutorial acerca del flujo de trabajo de suscripción: implementación de las actividades](#)
4. [Parte 4 del tutorial acerca del flujo de trabajo de suscripción: implementación del sondeador de tareas de actividades](#)
5. [Tutorial acerca del flujo de trabajo de suscripción: ejecución del flujo de trabajo](#)

Tutorial sobre el flujo de trabajo de suscripciones, parte 1: Uso de Amazon SWF con AWS SDK para Ruby

Temas

- [Incluya el AWS SDK para Ruby](#)

- [Configuración de la sesión AWS](#)
- [Registro de un dominio de Amazon SWF](#)
- [Siguiendo los pasos](#)

Incluya el AWS SDK para Ruby

Para empezar, cree un archivo llamado `utils.rb`. El código de este archivo obtendrá, o creará si es necesario, el dominio de Amazon SWF que utiliza el código tanto de las actividades como del flujo de trabajo. También permitirá almacenar el código común a todas nuestras clases.

En primer lugar, tenemos que incluir la biblioteca `aws-sdk-v1` en el código, a fin de poder utilizar las características que proporciona el SDK para Ruby.

```
require 'aws-sdk-v1'
```

Esto nos da acceso al espacio de nombres de AWS, que permite establecer valores globales relacionados con la sesión, como AWS las credenciales y la región, y también proporciona acceso al servicio. AWS APIs

Configuración de la sesión AWS

Configuraremos la sesión de AWS configurando nuestras credenciales de AWS (que son necesarias para acceder a los servicios de AWS) y la región de AWS que vamos a utilizar.

Hay varias formas de [configurar las credenciales de AWS en el SDK para Ruby](#): configurándolas en variables de entorno (`AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`) o configurándolas con `AWS.config`. Utilizaremos el segundo método y las cargaremos a partir de un archivo de configuración YAML, llamado `aws-config.txt`, que tiene este aspecto.

```
---
:access_key_id: REPLACE_WITH_ACCESS_KEY_ID
:secret_access_key: REPLACE_WITH_SECRET_ACCESS_KEY
```

Cree este archivo ahora y sustituya las cadenas que comienzan por `REPLACE_WITH_` por el ID de la clave de acceso y la clave de acceso secreta. Para obtener información sobre las claves de acceso de AWS, consulte [¿Cómo puedo obtener credenciales de seguridad?](#) en la Referencia general de Amazon Web Services.

También necesitamos configurar la AWS región que vamos a usar. Como utilizaremos el [servicio de mensajes cortos \(SMS\)](#) para enviar mensajes de texto al teléfono del usuario con Amazon SNS, debemos asegurarnos de utilizar una región compatible con Amazon SNS. Consulte [las regiones y los países compatibles](#) en la guía para desarrolladores de Amazon Simple Notification Service (Amazon SNS).

Note

Si no tiene acceso a us-east-1, o si no le interesa ejecutar la demostración con la mensajería SMS habilitada, puede utilizar cualquier región que desee. Además, puede eliminar la funcionalidad SMS del ejemplo y utilizar el correo electrónico como único punto de conexión para suscribirse al tema de Amazon SNS.

Para obtener más información sobre el envío de mensajes SMS, consulte [Envío y recepción de notificaciones por SMS mediante Amazon SNS](#) en la Guía para desarrolladores de Amazon Simple Notification Service.

Ahora vamos a añadir código a `utils.rb` para cargar el archivo de configuración, a obtener las credenciales del usuario y a proporcionar las credenciales y la región a [AWS.config](#).

```
require 'yaml'

# Load the user's credentials from a file, if it exists.
begin
  config_file = File.open('aws-config.txt') { |f| f.read }
rescue
  puts "No config file! Hope you set your AWS credentials in the environment..."
end

if config_file.nil?
  options = { }
else
  options = YAML.load(config_file)
end

# SMS Messaging (which can be used by Amazon SNS) is available only in the
# `us-east-1` region.
$SMS_REGION = 'us-east-1'
options[:region] = $SMS_REGION

# Now, set the options
```

```
AWS.config = options
```

Registro de un dominio de Amazon SWF

Para utilizar Amazon SWF, debemos configurar un dominio: una entidad con nombre que contendrá los flujos de trabajo y las actividades. Puede tener registrados muchos dominios de Amazon SWF, pero todos deben tener nombres únicos en su AWS cuenta y los flujos de trabajo no pueden interactuar entre dominios: todos los flujos de trabajo y actividades de su aplicación deben estar en el mismo dominio para poder interactuar entre sí.

Como utilizaremos el mismo dominio en toda la aplicación, crearemos una función en *utils.rb* called *init_domain* que recuperará el dominio Amazon SWF denominado SWFSample Domain.

Una vez que haya registrado un dominio, puede reutilizarlo en numerosas ejecuciones de flujos de trabajo. Sin embargo, es un error intentar registrar un dominio que ya existe, por eso el código comprobará en primer lugar si el dominio existe y utilizará el dominio existente, si lo encuentra. Si no encuentra el dominio, lo crearemos.

Para trabajar con los dominios de Amazon SWF en el SDK para Ruby, utilice [AWS::SimpleWorkflow.domains](#), que devuelve un [DomainCollection](#) que se puede utilizar tanto para enumerar como para registrar dominios:

- [Para comprobar si un dominio ya está registrado, consulte la lista proporcionada por .domains.registered. AWS::Simpleworkflow](#)
- [Para registrar un dominio nuevo, usa .domains.register. AWS::Simpleworkflow](#)

Este es el código para *init_domain* en *utils.rb*.

```
# Registers the domain that the workflow will run in.
def init_domain
  domain_name = 'SWFSampleDomain'
  domain = nil
  swf = AWS::SimpleWorkflow.new

  # First, check to see if the domain already exists and is registered.
  swf.domains.registered.each do | d |
    if(d.name == domain_name)
      domain = d
      break
    end
  end
end
```

```
end

if domain.nil?
  # Register the domain for one day.
  domain = swf.domains.create(
    domain_name, 1, { :description => "#{domain_name} domain" })
end

return domain
end
```

Siguientes pasos

A continuación, creará el código de iniciador y flujo de trabajo en [Segunda parte del tutorial acerca del flujo de trabajo de suscripción: implementación del flujo de trabajo](#).

Segunda parte del tutorial acerca del flujo de trabajo de suscripción: implementación del flujo de trabajo

Hasta ahora, nuestro código ha sido bastante genérico. Esta es la parte donde empezamos a definir realmente lo que hace nuestro flujo de trabajo y qué actividades necesitará para implementarlo.

Temas

- [Diseño del flujo de trabajo](#)
- [Configuración del código del flujo de trabajo](#)
- [Registro de un flujo de trabajo](#)
- [Sondeo de decisiones](#)
- [Comienzo de la ejecución del flujo de trabajo](#)
- [Siguientes pasos](#)

Diseño del flujo de trabajo

Como recordará, la idea inicial de este flujo de trabajo consistía en los siguientes pasos:

1. Obtenga una dirección de suscripción (correo electrónico o SMS) del usuario.
2. Cree un tema de SNS y suscriba los puntos de conexión proporcionados al tema.

3. Espere a que el usuario confirme la suscripción.
4. Si el usuario la confirma, publique un mensaje de felicitación en el tema.

Puede pensar en cada paso del flujo de trabajo como una actividad que debe realizar. El flujo de trabajo es responsable de programar cada actividad en el momento oportuno y de coordinar la transferencia de datos entre actividades.

En este flujo de trabajo, crearemos una actividad independiente para cada uno de estos pasos, y les daremos nombres descriptivos:

1. `get_contact_activity`
2. `subscribe_topic_activity`
3. `wait_for_confirmation_activity`
4. `send_result_activity`

Estas actividades se ejecutarán por orden, y los datos de cada paso se utilizarán en el paso siguiente.

Podríamos diseñar nuestra aplicación de modo que todo el código se encuentre en un archivo de origen, pero esta estrategia iría en contra de la forma en que se ha diseñado Amazon SWF. De hecho, se ha diseñado para flujos de trabajo que abarcan la totalidad de Internet, por eso vamos a desglosar la aplicación en dos ejecutables distintos:

- `swf_sns_workflow.rb`: contiene el flujo de trabajo y el iniciador de flujo de trabajo.
- `swf_sns_activities.rb`: contiene las actividades y el iniciador de las actividades.

Las implementaciones de flujo de trabajo y actividades pueden ejecutarse en distintas ventanas, distintos equipos o incluso distintas partes del mundo. Puesto que Amazon SWF hace un seguimiento de los detalles de los flujos de trabajo y actividades, el flujo de trabajo puede coordinar las programaciones y transferencias de datos de las actividades sin importar donde se ejecuten.

Configuración del código del flujo de trabajo

Para empezar, cree un archivo llamado `swf_sns_workflow.rb`. En este archivo, declare una clase llamada `SampleWorkflow`. Esta es la declaración de clase y su constructor, el método `initialize`.

```
require_relative 'utils.rb'
```

```
# SampleWorkflow - the main workflow for the SWF/SNS Sample
#
# See the file called `README.md` for a description of what this file does.
class SampleWorkflow

  attr_accessor :name

  def initialize(workflowId)

    # the domain to look for decision tasks in.
    @domain = init_domain

    # the task list is used to poll for decision tasks.
    @workflowId = workflowId

    # The list of activities to run, in order. These name/version hashes can be
    # passed directly to AWS::SimpleWorkflow::DecisionTask#schedule_activity_task.
    @activity_list = [
      { :name => 'get_contact_activity', :version => 'v1' },
      { :name => 'subscribe_topic_activity', :version => 'v1' },
      { :name => 'wait_for_confirmation_activity', :version => 'v1' },
      { :name => 'send_result_activity', :version => 'v1' },
    ].reverse! # reverse the order... we're treating this like a stack.

    register_workflow
  end
end
```

Como puede ver, mantenemos los siguientes datos de instancia de clase:

- `domain`: el nombre del dominio recuperado de `init_domain` en `utils.rb`.
- `workflowId`: la lista de tareas transmitida a `initialize`.
- `activity_list`: la lista de actividades, que contiene los nombres y las versiones de las actividades que ejecutaremos.

El nombre de dominio, el nombre de la actividad y la versión de la actividad son suficientes para que Amazon SWF identifique positivamente un tipo de actividad, por lo que estos son todos los datos que necesitamos acerca de nuestras actividades para programarlas.

La lista de tareas utilizada por el código del decisor del flujo de trabajo para realizar sondeos para obtener tareas de decisión y actividades de programación.

Al final de esta función, llamamos a un método que aún no hemos definido: `register_workflow`. Definiremos este método a continuación.

Registro de un flujo de trabajo

Par utilizar un tipo de flujo de trabajo, primero es necesario registrarlo. Al igual que un tipo de actividad, un tipo de flujo de trabajo se identifica por su dominio, nombre y versión. Además, al igual que los dominios y tipos de actividad, no puede volver a registrar un tipo de flujo de trabajo existente. Si necesita cambiar algo de un tipo de flujo de trabajo, debe hacerlo mediante una nueva versión, lo que básicamente crea un nuevo tipo.

Este es el código para `register_workflow`, que se utiliza para recuperar el tipo de flujo de trabajo existente que hemos registrado en una ejecución anterior o para registrar el flujo de trabajo si aún no se ha registrado.

```
# Registers the workflow
def register_workflow
  workflow_name = 'swf-sns-workflow'
  @workflow_type = nil

  # a default value...
  workflow_version = '1'

  # Check to see if this workflow type already exists. If so, use it.
  @domain.workflow_types.each do | a |
    if (a.name == workflow_name) && (a.version == workflow_version)
      @workflow_type = a
    end
  end

  if @workflow_type.nil?
    options = {
      :default_child_policy => :terminate,
      :default_task_start_to_close_timeout => 3600,
      :default_execution_start_to_close_timeout => 24 * 3600 }

    puts "registering workflow: #{workflow_name}, #{workflow_version},
#{options.inspect}"
    @workflow_type = @domain.workflow_types.register(workflow_name, workflow_version,
options)
  end
end
```

```
puts "*** registered workflow: #{workflow_name}"
end
```

En primer lugar, para comprobar si el nombre y la versión del flujo de trabajo ya están registrados, procedemos a la iteración a través de la colección de [workflow_types](#) del dominio. Si se encuentra una coincidencia, utilizaremos el tipo de flujo de trabajo que ya esté registrado.

Si no encontramos ninguna coincidencia, se registra un nuevo tipo de flujo de trabajo (llamando a [register](#) en la misma `workflow_types` colección en la que estábamos buscando el flujo de trabajo) con el nombre 'swf-sns-workflow', la versión '1' y las siguientes opciones.

```
options = {
  :default_child_policy => :terminate,
  :default_task_start_to_close_timeout => 3600,
  :default_execution_start_to_close_timeout => 24 * 3600 }
```

Las opciones transmitidas durante el registro se utilizan para configurar el comportamiento predeterminado del tipo de trabajo. De este modo, no es necesario configurar estos valores cada vez que se inicia una nueva ejecución del flujo de trabajo.

Aquí, se definen solo algunos valores de tiempo de espera: la duración máxima desde el momento en que se inicia la tarea hasta que se cierra (una hora) y la duración máxima hasta que se completa el flujo de trabajo (24 horas). Si se supera cualquiera de estos plazos, la tarea o el flujo de trabajo agotarán el tiempo de espera.

Para obtener más información acerca de los valores de tiempo de espera, consulte [Tipos de tiempo de espera de Amazon SWF](#).

Sondeo de decisiones

En el centro de cada ejecución de flujo de trabajo se encuentra un decisor. La responsabilidad del decisor consiste en administrar la ejecución del flujo de trabajo en sí. El decisor recibe tareas de decisión y responde a ellas bien programando nuevas actividades, cancelando y reiniciando actividades, o definiendo el estado de la ejecución del flujo de trabajo como completo, cancelado o erróneo.

El decisor utiliza el nombre de la lista de tareas de la ejecución del flujo de trabajo para recibir tareas de decisión y responder a ellas. Si desea realizar sondeos para obtener tareas de decisión, llame al [sondeo](#) de la colección de [decision_tasks](#) del dominio para recorrer las tareas de decisión

disponibles. A continuación, podrá buscar nuevos eventos en la tarea de decisión realizando una iteración en su colección de [new_events](#).

Los eventos devueltos son [AWS::SimpleWorkflow::HistoryEvent](#) objetos y puede obtener el tipo de evento utilizando el miembro [event_type](#) del evento devuelto. Para obtener una lista y una descripción de los tipos de eventos del historial, consulte [HistoryEvent](#) la referencia de la API de Amazon Simple Workflow Service.

Este es el principio de la lógica del sondeador de tareas de decisión. Un nuevo método de la clase de flujo de trabajo ha llamado `poll_for_decisions`.

```
def poll_for_decisions
  # first, poll for decision tasks...
  @domain.decision_tasks.poll(@workflowId) do | task |
    task.new_events.each do | event |
      case event.event_type
```

Ahora crearemos ramificaciones de la ejecución del decisor según el `event_type` recibido. El primero que probablemente recibamos es `WorkflowExecutionStarted`. Cuando se recibe este evento, significa que Amazon SWF señala al decisor que debe empezar la ejecución del flujo de trabajo. Para comenzar, programe la primera actividad llamando a [schedule_activity_task](#) en la tarea que se recibió durante el sondeo.

Transmitiremos la primera actividad que declaramos en la lista de actividades, que, como invertimos la lista para poder usarla como una pila, ocupa la posición `last` de la lista. Las “actividades” que definimos son solo mapas compuestos del nombre y el número de versión, pero eso es todo lo que necesita Amazon SWF para identificar la actividad que se va a programar, en el supuesto caso de que la actividad ya se ha registrado.

```
when 'WorkflowExecutionStarted'
  # schedule the last activity on the (reversed, remember?) list to
  # begin the workflow.
  puts "*** scheduling activity task: #{@activity_list.last[:name]}"

  task.schedule_activity_task( @activity_list.last,
    { :workflowId => "#{@workflowId}-activities" } )
```

Cuando se programa una actividad, Amazon SWF envía una tarea de actividad a la lista de tareas de actividad que transmitimos durante su programación, lo que indica que la tarea debe empezar.

Las tareas de actividad se abordarán en [Parte 3 del tutorial acerca del flujo de trabajo de suscripción: implementación de las actividades](#), pero merece la pena señalar que la tarea no se ejecuta aquí. Solo indicamos a Amazon SWF que esta debe programarse.

La siguiente actividad que debemos abordar es el `ActivityTaskCompleted` evento, que se produce cuando Amazon SWF recibe una respuesta de actividad completada de una tarea de actividad.

```
when 'ActivityTaskCompleted'
  # we are running the activities in strict sequential order, and
  # using the results of the previous activity as input for the next
  # activity.
  last_activity = @activity_list.pop

  if(@activity_list.empty?)
    puts "!! All activities complete! Sending complete_workflow_execution..."
    task.complete_workflow_execution
    return true;
  else
    # schedule the next activity, passing any results from the
    # previous activity. Results will be received in the activity
    # task.
    puts "*** scheduling activity task: #{@activity_list.last[:name]}"
    if event.attributes.has_key?('result')
      task.schedule_activity_task(
        @activity_list.last,
        { :input => event.attributes[:result],
          :workflowId => "#{@workflowId}-activities" } )
    else
      task.schedule_activity_task(
        @activity_list.last, { :workflowId => "#{@workflowId}-activities" } )
    end
  end
end
```

Como ejecutamos nuestras tareas de forma lineal y solo se ejecuta una actividad a la vez, aprovecharemos esta oportunidad para eliminar de la `activity_list` pila la tarea completada. Si esto resulta en una lista vacía, quiere decir que se ha completado el flujo de trabajo. En ese caso, informamos a Amazon SWF de que se ha completado el flujo de trabajo; para ello, llamamos a [complete_workflow_execution](#) en la tarea.

En caso de que la lista aún tenga entradas, programaremos la siguiente actividad de la lista (de nuevo, en la última posición). Sin embargo, esta vez intentaremos comprobar si la actividad anterior devolvió algún dato de resultados a Amazon SWF tras finalizar, dato que se proporciona al flujo de

trabajo en los atributos del evento, en la clave `result` opcional. Si la actividad generó un resultado, lo transmitiremos como la opción `input` a la siguiente actividad programada, junto con la lista de tareas de actividad.

Al recuperar los valores `result` de las actividades completadas, y al configurar los valores `input` de las actividades programadas, es posible transmitir datos de una actividad a la siguiente, o utilizar datos de una actividad para modificar el comportamiento del decisor en función de los resultados de una actividad.

A efectos de este tutorial, estos dos tipos de eventos son los más importantes a la hora de definir el comportamiento del flujo de trabajo. Sin embargo, una actividad puede generar eventos distintos a `ActivityTaskCompleted`. Para resumir nuestro código de decisión, proporcionaremos un código de controlador de demostración para los `ActivityTaskFailed` eventos `ActivityTaskTimedOut`, además, para el `WorkflowExecutionCompleted` evento, que se generará cuando Amazon SWF procese `complete_workflow_execution` la llamada que realicemos cuando se nos acaben las actividades pendientes.

```
when 'ActivityTaskTimedOut'
  puts "!! Failing workflow execution! (timed out activity)"
  task.fail_workflow_execution
  return false

when 'ActivityTaskFailed'
  puts "!! Failing workflow execution! (failed activity)"
  task.fail_workflow_execution
  return false

when 'WorkflowExecutionCompleted'
  puts "## Yesss, workflow execution completed!"
  task.workflow_execution.terminate
  return false
end
end
end
end
```

Comienzo de la ejecución del flujo de trabajo

Antes de que el flujo de trabajo pueda realizar sondeos para obtener tareas de decisión, es necesario comenzar la ejecución del flujo de trabajo.

Para iniciar la ejecución del flujo de trabajo, llame a [start_execution](#) en el tipo de flujo de trabajo registrado (). [AWS::SimpleWorkflow::WorkflowType](#) Definiremos un pequeño encapsulador a este nivel a fin de utilizar el miembro de instancia `workflow_type` que recuperamos en el constructor de clase.

```
def start_execution
  workflow_execution = @workflow_type.start_execution( {
    :workflowId => @workflowId } )
  poll_for_decisions
end
end
```

Una vez que el flujo de trabajo se esté ejecutando, los eventos de decisión empezarán a aparecer en la lista de tareas del flujo de trabajo, que se transmite como una opción de ejecución del flujo de trabajo de [start_execution](#).

A diferencia de las opciones que se ofrecen cuando se registra el tipo de flujo de trabajo, las opciones que se transmiten a `start_execution` no se consideran parte del tipo de flujo de trabajo. Puede cambiarlas para cada ejecución de flujo de trabajo sin cambiar la versión del flujo de trabajo.

Como queremos que el flujo de trabajo comience a ejecutarse cuando ejecutemos el archivo, añadimos código que cree una instancia de la clase y, a continuación, llame al `start_execution` método que acabamos de definir.

```
if __FILE__ == $0
  require 'securerandom'

  # Use a different task list name every time we start a new workflow execution.
  #
  # This avoids issues if our pollers re-start before SWF considers them closed,
  # causing the pollers to get events from previously-run executions.
  workflowId = SecureRandom.uuid

  # Let the user start the activity worker first...

  puts ""
  puts "Amazon SWF Example"
  puts "-----"
  puts ""
  puts "Start the activity worker, preferably in a separate command-line window, with"
  puts "the following command:"
```

```
puts ""
puts "> ruby swf_sns_activities.rb #{workflowId}-activities"
puts ""
puts "You can copy & paste it if you like, just don't copy the '>' character."
puts ""
puts "Press return when you're ready..."

i = gets

# Now, start the workflow.

puts "Starting workflow execution."
sample_workflow = SampleWorkflow.new(workflowId)
sample_workflow.start_execution
end
```

Para evitar conflictos de nombres en la lista de tareas, utilice `SecureRandom.uuid` para generar UUID aleatorios que se utilizarán como nombre de la lista de tareas, lo que garantiza que se empleará un nombre de tarea distinto para cada ejecución del flujo de trabajo.

Note

Las listas de tareas se utilizan para registrar eventos en torno a la ejecución del flujo de trabajo, por lo que si utiliza la misma lista de tareas para varias ejecuciones del mismo tipo de flujo de trabajo, podría obtener eventos que se generaron durante una ejecución anterior, en especial si las ejecuta de forma casi consecutiva, lo que es a menudo el caso cuando se prueba nuevo código o se realizan otras pruebas.

Para evitar el problema de tener que tratar con elementos de ejecuciones anteriores, es posible utilizar una nueva lista de tareas para cada ejecución, especificándola al empezar la ejecución del flujo de trabajo.

También es necesario algo de código para proporcionar instrucciones a la persona a cargo de la ejecución (probablemente usted), y proporcionar la versión de la "actividad" de la lista de tareas. El decisor utilizará el nombre de esta lista de tareas para programar actividades para el flujo de trabajo, en tanto que la implementación de actividades prestará atención a los eventos de actividad en el nombre de esta lista de tareas para saber cuándo empezar las actividades programadas y proporcionar actualizaciones sobre la ejecución de la actividad.

El código también espera a que el usuario comience a ejecutar el iniciador de actividades antes de comenzar la ejecución del flujo de trabajo. El iniciador de actividades estará entonces listo para responder cuando las tareas de actividad empiecen a aparecer en la lista de tareas proporcionada.

Siguientes pasos

Ha implementado el flujo de trabajo. A continuación, definirá las actividades y un iniciador de actividades en [Parte 3 del tutorial acerca del flujo de trabajo de suscripción: implementación de las actividades](#).

Parte 3 del tutorial acerca del flujo de trabajo de suscripción: implementación de las actividades

Ahora implementaremos cada una de las actividades en nuestro flujo de trabajo, empezando con una clase base que proporciona algunas características comunes para el código de actividad.

Temas

- [Definición de un tipo de actividad básica](#)
- [Definiendo GetContactActivity](#)
- [Definir SubscribeTopicActivity](#)
- [¿Definiendo WaitForConfirmationActivity](#)
- [¿Definiendo SendResultActivity](#)
- [Siguientes pasos](#)

Definición de un tipo de actividad básica

Al diseñar el flujo de trabajo, identificamos las siguientes actividades:

- `get_contact_activity`
- `subscribe_topic_activity`
- `wait_for_confirmation_activity`
- `send_result_activity`

Implementaremos cada una de estas actividades ahora. Como nuestras actividades compartirán algunas características, hagamos un poco de trabajo preliminar y creemos un código común

que puedan compartir. Lo llamaremos `BasicActivity` y lo definiremos en un nuevo archivo llamado `basic_activity.rb`.

Al igual que con los demás archivos de código fuente, incluiremos `utils.rb` para obtener acceso a la función `init_domain` para configurar el dominio de ejemplo.

```
require_relative 'utils.rb'
```

A continuación, declararemos la clase de actividad básica y algunos datos comunes que serán de nuestro interés para cada actividad. Guardaremos la [AWS::SimpleWorkflow::ActivityType](#) instancia, el nombre y los resultados de la actividad en los atributos de la clase.

```
class BasicActivity

  attr_accessor :activity_type
  attr_accessor :name
  attr_accessor :results
```

Estos atributos obtienen acceso a los datos de la instancia que se definen en el método `initialize` de la clase, que toma un nombre de actividad, una versión opcional y un mapa de opciones que se van a utilizar al registrar la actividad en Amazon SWF.

```
def initialize(name, version = 'v1', options = nil)

  @activity_type = nil
  @name = name
  @results = nil

  # get the domain to use for activity tasks.
  @domain = init_domain

  # Check to see if this activity type already exists.
  @domain.activity_types.each do | a |
    if (a.name == @name) && (a.version == version)
      @activity_type = a
    end
  end

  if @activity_type.nil?
    # If no options were specified, use some reasonable defaults.
    if options.nil?
      options = {
```

```
      # All timeouts are in seconds.
      :default_task_heartbeat_timeout => 900,
      :default_task_schedule_to_start_timeout => 120,
      :default_task_schedule_to_close_timeout => 3800,
      :default_task_start_to_close_timeout => 3600 }
    end
    @activity_type = @domain.activity_types.register(@name, version, options)
  end
end
```

Al igual que ocurre con el registro del tipo de flujo de trabajo, si ya se ha registrado un tipo de actividad, podemos recuperarla consultando la colección de [activity_types](#) del dominio. Si no se encuentra la actividad, se registrará.

También, al igual que ocurre con los tipos de flujo de trabajo, puede establecer opciones predeterminadas que se almacenen con su tipo de actividad al registrarla.

Lo último que obtiene nuestra actividad básica es una forma coherente de ejecutarla. Definiremos un método `do_activity` que tome una tarea de actividad. Tal como se indica, podemos usar la tarea de actividad transmitida para recibir datos a través de su atributo de instancia de `input`.

```
def do_activity(task)
  @results = task.input # may be nil
  return true
end
```

Con esto termina la `BasicActivity` clase. Ahora la usaremos para hacer que la definición de nuestras actividades sea sencilla y coherente.

Definiendo `GetContactActivity`

La primera actividad que se ejecuta durante una ejecución de flujo de trabajo es `get_contact_activity`, que recupera la información de suscripción a un tema de Amazon SNS del usuario.

Creará un nuevo archivo llamado `get_contact_activity.rb` y requiere `ambosyaml`, que usaremos para preparar una cadena para pasarla a Amazon SWF y `basic_activity.rb` que usaremos como base para esta `GetContactActivity` clase.

```
require 'yaml'
```

```
require_relative 'basic_activity.rb'

# **GetContactActivity** provides a prompt for the user to enter contact
# information. When the user successfully enters contact information, the
# activity is complete.
class GetContactActivity < BasicActivity
```

Como hemos introducido el código de registro de la actividad `BasicActivity`, el `initialize` método `GetContactActivity` es bastante sencillo. Nos limitamos a llamar al constructor de clases base con el nombre de actividad, `get_contact_activity`. Esto es todo lo que hace falta para registrar nuestra actividad.

```
# initialize the activity
def initialize
  super('get_contact_activity')
end
```

Ahora definiremos el `do_activity` método, que solicita el número de and/or teléfono de correo electrónico del usuario.

```
def do_activity(task)
  puts ""
  puts "Please enter either an email address or SMS message (mobile phone) number
to"
  puts "receive SNS notifications. You can also enter both to use both address
types."
  puts ""
  puts "If you enter a phone number, it must be able to receive SMS messages, and
must"
  puts "be 11 digits (such as 12065550101 to represent the number
1-206-555-0101)."
```



```
  input_confirmed = false
  while !input_confirmed
    puts ""
    print "Email: "
    email = $stdin.gets.strip

    print "Phone: "
    phone = $stdin.gets.strip

    puts ""
```

```
    if (email == '') && (phone == '')
      print "You provided no subscription information. Quit? (y/n)"
      confirmation = $stdin.gets.strip.downcase
      if confirmation == 'y'
        return false
      end
    else
      puts "You entered:"
      puts "  email: #{email}"
      puts "  phone: #{phone}"
      print "\nIs this correct? (y/n): "
      confirmation = $stdin.gets.strip.downcase
      if confirmation == 'y'
        input_confirmed = true
      end
    end
  end
end

# make sure that @results is a single string. YAML makes this easy.
@results = { :email => email, :sms => phone }.to_yaml
return true
end
end
```

Al final de `do_activity`, tomamos el correo electrónico y el número de teléfono recuperados del usuario, los colocamos en un mapa y, a continuación, usamos `to_yaml` para convertir todo el mapa a una cadena YAML. Existe un motivo importante para esto: los resultados que pase a Amazon SWF al completar una actividad deben ser datos de cadena únicamente. La capacidad de Ruby de convertir fácilmente objetos a cadenas YAML y, posteriormente, a objetos una vez más es, por suerte, adecuada para este fin.

Ese es el final de la implementación `get_contact_activity`. Estos datos se usarán a continuación en la implementación de `subscribe_topic_activity`.

Definir `SubscribeTopicActivity`

Ahora profundizaremos en Amazon SNS y crearemos una actividad que utilice la información que genera `get_contact_activity` para suscribir al usuario a un tema de Amazon SNS.

Cree un nuevo archivo llamado `subscribe_topic_activity.rb`, añada los mismos requisitos que usamos para `get_contact_activity`, declare su clase y proporcione su método `initialize`.

```
require 'yaml'
require_relative 'basic_activity.rb'

# **SubscribeTopicActivity** sends an SMS / email message to the user, asking for
# confirmation. When this action has been taken, the activity is complete.
class SubscribeTopicActivity < BasicActivity

  def initialize
    super('subscribe_topic_activity')
  end
end
```

Ahora que tenemos el código para que la actividad se configure y registre, añadiremos código para crear un tema de Amazon SNS. Para ello, utilizaremos el método [create_topic](#) del [AWS::SNS::Client](#) objeto.

Añada el método `create_topic` a la clase, que toma un objeto de cliente de Amazon SNS transmitido.

```
def create_topic(sns_client)
  topic_arn = sns_client.create_topic(:name => 'SWF_Sample_Topic')[[:topic_arn]]

  if topic_arn != nil
    # For an SMS notification, setting `DisplayName` is *required*. Note that
    # only the *first 10 characters* of the DisplayName will be shown on the
    # SMS message sent to the user, so choose your DisplayName wisely!
    sns_client.set_topic_attributes( {
      :topic_arn => topic_arn,
      :attribute_name => 'DisplayName',
      :attribute_value => 'SWFSample' } )
  else
    @results = {
      :reason => "Couldn't create SNS topic", :detail => "" }.to_yaml
    return nil
  end

  return topic_arn
end
```

Una vez que tengamos el nombre de recurso de Amazon (ARN) del tema, podemos usarlo con el método `set_topic_attributes` [del cliente de](#) Amazon SNS para establecer el tema, que `DisplayName` es necesario para enviar mensajes SMS con Amazon SNS.

Por último, definiremos el método `do_activity`. Comenzaremos recopilando los datos que se han pasado mediante la opción `input` al programarse la actividad. Tal como se ha mencionado anteriormente, estos deben transferirse como cadena, que creamos mediante `to_yaml`. Una vez recuperados, usaremos `YAML.load` para convertir los datos en objetos Ruby.

Este es el principio de `do_activity`, donde recuperamos los datos de entrada.

```
def do_activity(task)
  activity_data = {
    :topic_arn => nil,
    :email => { :endpoint => nil, :subscription_arn => nil },
    :sms => { :endpoint => nil, :subscription_arn => nil },
  }

  if task.input != nil
    input = YAML.load(task.input)
    activity_data[:email][:endpoint] = input[:email]
    activity_data[:sms][:endpoint] = input[:sms]
  else
    @results = { :reason => "Didn't receive any input!", :detail => "" }.to_yaml
    puts(" #{@results.inspect}")
    return false
  end

  # Create an SNS client. This is used to interact with the service. Set the
  # region to $SMS_REGION, which is a region that supports SMS notifications
  # (defined in the file `utils.rb`).
  sns_client = AWS::SNS::Client.new(
    :config => AWS.config.with(:region => $SMS_REGION))
end
```

Si no hemos recibido ninguna entrada, no hay mucho que se pueda hacer, de modo que produciremos un error en la actividad.

Sin embargo, suponiendo que todo vaya bien, seguiremos rellenando nuestro `do_activity` método, buscaremos un cliente de Amazon SNS con el AWS SDK para Ruby y lo pasaremos a nuestro `create_topic` método para crear el tema Amazon SNS.

```
# Create the topic and get the ARN
activity_data[:topic_arn] = create_topic(sns_client)

if activity_data[:topic_arn].nil?
  return false
end
```

```
end
```

Hay un par de aspectos que merece la pena señalar aquí:

- Utilizamos [AWS.config.with](#) para establecer la región de nuestro cliente de Amazon SNS. Dado que deseamos enviar mensajes SMS, usaremos la región habilitada para SMS que declaramos en `utils.rb`.
- Guardamos el ARN del tema en nuestro mapa `activity_data`. Este forma parte de los datos que se pasarán a la siguiente actividad de nuestro flujo de trabajo.

Por último, esta actividad suscribe al usuario al tema de Amazon SNS mediante los puntos de conexión transmitidos (correo electrónico y SMS). No exigimos que el usuario escriba ambos puntos de conexión, pero necesitamos al menos uno.

```
# Subscribe the user to the topic, using either or both endpoints.
[:email, :sms].each do | x |
  ep = activity_data[x][:endpoint]
  # don't try to subscribe an empty endpoint
  if (ep != nil && ep != "")
    response = sns_client.subscribe( {
      :topic_arn => activity_data[:topic_arn],
      :protocol => x.to_s, :endpoint => ep } )
    activity_data[x][:subscription_arn] = response[:subscription_arn]
  end
end
```

[AWS::SNS::Client.subscribe](#) toma el tema ARN, el protocolo (que, hábilmente, disfrazamos de clave de mapa para `activity_data` el punto final correspondiente).

Por último, podemos volver a empaquetar la información para la siguiente actividad en formato YAML, de modo que podamos devolverla a Amazon SWF.

```
# if at least one subscription arn is set, consider this a success.
if (activity_data[:email][:subscription_arn] != nil) or (activity_data[:sms]
[:subscription_arn] != nil)
  @results = activity_data.to_yaml
else
  @results = { :reason => "Couldn't subscribe to SNS topic", :detail =>
"" }.to_yaml
  puts(" #{@results.inspect}")
```

```
        return false
      end
      return true
    end
  end
```

De ese modo se completa la implementación de `subscribe_topic_activity`. A continuación, definiremos `wait_for_confirmation_activity`.

¿Definiendo `WaitForConfirmationActivity`

Una vez que un usuario se suscriba a un tema de Amazon SNS, este deberá confirmar la solicitud de suscripción. En este caso, esperaremos la confirmación del usuario por correo electrónico o mediante un mensaje SMS.

La actividad que espera a que el usuario confirme la suscripción se llama `wait_for_confirmation_activity` y la definiremos aquí. Para empezar, cree un nuevo archivo llamado `wait_for_confirmation_activity.rb` y configúrelo cuando hayamos configurado las actividades anteriores.

```
require 'yaml'
require_relative 'basic_activity.rb'

# **WaitForConfirmationActivity** waits for the user to confirm the SNS
# subscription. When this action has been taken, the activity is complete. It
# might also time out...
class WaitForConfirmationActivity < BasicActivity

  # Initialize the class
  def initialize
    super('wait_for_confirmation_activity')
  end
end
```

A continuación, empezaremos a definir el método `do_activity` y recuperaremos los datos de entrada de una variable local llamada `subscription_data`.

```
def do_activity(task)
  if task.input.nil?
    @results = { :reason => "Didn't receive any input!", :detail => "" }.to_yaml
    return false
  end
end
```

```
subscription_data = YAML.load(task.input)
```

Ahora que tenemos el tema ARN, podemos recuperarlo creando una nueva instancia de [AWS::SNS::Topic](#) pasándole el ARN.

```
topic = AWS::SNS::Topic.new(subscription_data[:topic_arn])

if topic.nil?
  @results = {
    :reason => "Couldn't get SWF topic ARN",
    :detail => "Topic ARN: #{topic.arn}" }.to_yaml
  return false
end
```

Ahora, comprobaremos el tema para ver si el usuario ha confirmado la suscripción mediante uno de los puntos de conexión. Solo exigiremos que se haya confirmado un punto de conexión para considerar la actividad un éxito.

Cualquier tema de Amazon SNS mantiene una lista de las [suscripciones](#) a dicho tema, y podemos verificar si el usuario ha confirmado o no una suscripción determinada al comprobar si el ARN de la suscripción está establecido en otra opción distinta a `PendingConfirmation`.

```
# loop until we get some indication that a subscription was confirmed.
subscription_confirmed = false
while(!subscription_confirmed)
  topic.subscriptions.each do | sub |
    if subscription_data[sub.protocol.to_sym][:endpoint] == sub.endpoint
      # this is one of the endpoints we're interested in. Is it subscribed?
      if sub.arn != 'PendingConfirmation'
        subscription_data[sub.protocol.to_sym][:subscription_arn] = sub.arn
        puts "Topic subscription confirmed for (#{sub.protocol}:
#{sub.endpoint})"
        @results = subscription_data.to_yaml
        return true
      else
        puts "Topic subscription still pending for (#{sub.protocol}:
#{sub.endpoint})"
      end
    end
  end
end
```

Si obtenemos un ARN para la suscripción, lo guardaremos en los datos de resultados de la actividad, lo convertiremos a YAML y devolveremos true de `do_activity`, que indica que la actividad se ha completado correctamente.

Como esperar a que se confirme una suscripción puede llevar un tiempo, de vez en cuando suspendemos la tarea de `record_heartbeat` la actividad. Esto indica a Amazon SWF que la actividad se sigue procesando y que puede utilizarse para proporcionar actualizaciones acerca del progreso de la actividad (si realiza alguna tarea, como el procesamiento de archivos, de cuyo progreso pueda informar).

```
task.record_heartbeat!(
  { :details => "#{topic.num_subscriptions_confirmed} confirmed,
#{topic.num_subscriptions_pending} pending" })
  # sleep a bit.
  sleep(4.0)
end
```

Esto pone fin a nuestro bucle `while`. Si de algún modo salimos del bucle `while` sin éxito, notificaremos el error y pondremos fin al método `do_activity`.

```
if (subscription_confirmed == false)
  @results = {
    :reason => "No subscriptions could be confirmed",
    :detail => "#{topic.num_subscriptions_confirmed} confirmed,
#{topic.num_subscriptions_pending} pending" }.to_yaml
  return false
end
end
end
```

Eso pone fin a la implementación de `wait_for_confirmation_activity`. Solo tenemos una actividad más que definir: `send_result_activity`.

¿Definiendo `SendResultActivity`

Si el flujo de trabajo ha progresado hasta este punto, quiere decir que hemos suscrito correctamente al usuario a un tema de Amazon SNS y que el usuario ha confirmado la suscripción.

Nuestra última actividad, `send_result_activity`, envía al usuario una confirmación de la suscripción correcta al tema, mediante el tema al que se ha suscrito al usuario y el punto de conexión con el que el usuario ha confirmado la suscripción.

Cree un nuevo archivo llamado `send_result_activity.rb` y configúrelo cuando hayamos configurado todas las actividades hasta el momento.

```
require 'yaml'
require_relative 'basic_activity.rb'

# **SendResultActivity** sends the result of the activity to the screen, and, if
# the user successfully registered using SNS, to the user using the SNS contact
# information collected.
class SendResultActivity < BasicActivity

  def initialize
    super('send_result_activity')
  end
end
```

Nuestro `do_activity` método también comienza de manera similar: obtiene los datos de entrada del flujo de trabajo, los convierte de YAML y, a continuación, usa el tema ARN para crear una [AWS::SNS::Topic](#) instancia.

```
def do_activity(task)
  if task.input.nil?
    @results = { :reason => "Didn't receive any input!", :detail => "" }
    return false
  end

  input = YAML.load(task.input)

  # get the topic, so we publish a message to it.
  topic = AWS::SNS::Topic.new(input[:topic_arn])

  if topic.nil?
    @results = {
      :reason => "Couldn't get SWF topic",
      :detail => "Topic ARN: #{topic.arn}" }
    return false
  end
end
```

Una vez que tengamos el tema, [publicaremos](#) un mensaje en él (y también lo repetiremos en la pantalla).

```
@results = "Thanks, you've successfully confirmed registration, and your
workflow is complete!"
```

```
# send the message via SNS, and also print it on the screen.
topic.publish(@results)
puts(@results)

return true
end
end
```

Al publicar en un tema de Amazon SNS, se envía el mensaje que el usuario haya proporcionado a todos los puntos de conexión suscritos y confirmados que existan para ese tema. Así pues, si el usuario ha confirmado con ambos un correo electrónico y un número de SMS, él o ella recibirá dos mensajes de confirmación, uno en cada punto de conexión.

Siguientes pasos

De este modo se completa la implementación de `send_result_activity`. Ahora vinculará todas estas actividades en una aplicación de actividad que controla las tareas de actividad y puede lanzar actividades como respuesta, en [Parte 4 del tutorial acerca del flujo de trabajo de suscripción: implementación del sondeador de tareas de actividades](#).

Parte 4 del tutorial acerca del flujo de trabajo de suscripción: implementación del sondeador de tareas de actividades

En Amazon SWF, las tareas de actividad de una ejecución de flujo de trabajo en ejecución aparecen en la lista de tareas de actividad, que se proporciona al programar una actividad en el flujo de trabajo.

Implementaremos un sondeador de actividades básico para gestionar estas tareas para nuestro flujo de trabajo y lo usaremos para lanzar nuestras actividades cada vez que Amazon SWF coloque una tarea en la lista de tareas de actividad para comenzar la actividad.

Para empezar, cree un nuevo archivo llamado `swf_sns_activities.rb`. Lo usaremos para:

- Crear instancias de las clases de actividad que creamos.
- Registrar cada actividad en Amazon SWF.
- Sondear actividades y llamar a `do_activity` para cada actividad cuando su nombre aparezca en la lista de tareas de actividad.

En `swf_sns_activities.rb`, añade las siguientes instrucciones para exigir cada una de las clases de actividad que definimos.

```
require_relative 'get_contact_activity.rb'  
require_relative 'subscribe_topic_activity.rb'  
require_relative 'wait_for_confirmation_activity.rb'  
require_relative 'send_result_activity.rb'
```

Ahora, crearemos la clase y proporcionaremos código de inicialización.

```
class ActivitiesPoller  
  
  def initialize(domain, workflowId)  
    @domain = domain  
    @workflowId = workflowId  
    @activities = {}  
  
    # These are the activities we'll run  
    activity_list = [  
      GetContactActivity,  
      SubscribeTopicActivity,  
      WaitForConfirmationActivity,  
      SendResultActivity ]  
  
    activity_list.each do | activity_class |  
      activity_obj = activity_class.new  
      puts "*** initialized and registered activity: #{activity_obj.name}"  
      # add it to the hash  
      @activities[activity_obj.name.to_sym] = activity_obj  
    end  
  end  
end
```

Además de guardar el dominio y la lista de tareas transferidos, este código crea instancias de cada una de las clases de actividad que creamos. Dado que cada clase registra su actividad asociada (consulte `basic_activity.rb` si necesita revisar ese código), esto es suficiente para informar a Amazon SWF acerca de todas las actividades que ejecutaremos.

Para cada actividad de la que se crea una instancia, la almacenamos en un mapa usando el nombre de actividad (como `get_contact_activity`) como clave, por lo que podemos consultarlas fácilmente en el código de sondeador de actividades, que definiremos a continuación.

Cree un nuevo método denominado `poll_for_activities` y llame al [sondeo](#) en la instancia de [activity_tasks](#) del dominio para obtener tareas de actividad.

```
def poll_for_activities
  @domain.activity_tasks.poll(@workflowId) do | task |
    activity_name = task.activity_type.name
```

Podemos obtener el nombre de actividad del miembro [activity_type](#) de la tarea. A continuación, usaremos el nombre de actividad asociado a esta tarea para consultar la clase en la que se ejecutará `do_activity`, pasando la tarea (que incluye los datos de entrada que deben transferirse a la actividad).

```
# find the task on the activities list, and run it.
if @activities.key?(activity_name.to_sym)
  activity = @activities[activity_name.to_sym]
  puts "*** Starting activity task: #{activity_name}"
  if activity.do_activity(task)
    puts "++ Activity task completed: #{activity_name}"
    task.complete!({ :result => activity.results })
    # if this is the final activity, stop polling.
    if activity_name == 'send_result_activity'
      return true
    end
  else
    puts "-- Activity task failed: #{activity_name}"
    task.fail!(
      { :reason => activity.results[:reason],
        :details => activity.results[:detail] } )
  end
else
  puts "couldn't find key in @activities list: #{activity_name}"
  puts "contents: #{@activities.keys}"
end
end
end
end
```

El código se limita a esperar a que `do_activity` se complete y, a continuación, llama a la instancia [complete!](#) o a [fail!](#) de la tarea, en función del código devuelto.

Note

Este código sale del encuestador una vez que se ha lanzado la actividad final, porque ha completado su misión y ha lanzado todas las actividades. En su propio código de Amazon SWF, si sus actividades pueden volver a ejecutarse, es posible que desee mantener el sondeador de actividades en ejecución de forma indefinida.

Este es el final del código de nuestra `ActivitiesPoller` clase, pero añadiremos un poco más de código al final del archivo para que el usuario pueda ejecutarlo desde la línea de comandos.

```
if __FILE__ == $0
  if ARGV.count < 1
    puts "You must supply a task-list name to use!"
    exit
  end
  poller = ActivitiesPoller.new(init_domain, ARGV[0])
  poller.poll_for_activities
  puts "All done!"
end
```

Si el usuario ejecuta el archivo en la línea de comandos (pasando una lista de tareas de actividad como primer argumento), este código creará una instancia de la clase de sondeador y la iniciará sondeando actividades. Una vez que se complete el sondeador (tras haberse lanzado la actividad final), basta con imprimir un mensaje y salir.

Eso es todo para el sondeador de actividades. Todo lo que le queda por hacer es ejecutar el código y ver cómo funciona, en [Tutorial acerca del flujo de trabajo de suscripción: ejecución del flujo de trabajo](#).

Tutorial acerca del flujo de trabajo de suscripción: ejecución del flujo de trabajo

Ahora que ha completado la implementación de su flujo de trabajo, actividades y los sondeadores de flujo de trabajo y actividades, está listo para ejecutar el flujo de trabajo.

Si aún no lo has hecho, tendrás que proporcionar tus claves de AWS acceso en el `aws-config.txt` archivo, como [Configuración de la sesión AWS](#) en la primera parte del tutorial.

Ahora, acceda a la línea de comandos y modifique el directorio donde se encuentran los archivos de origen del tutorial. Debería tener los siguientes archivos:

```
.
|-- aws-config.txt
|-- basic_activity.rb
|-- get_contact_activity.rb
|-- send_result_activity.rb
|-- subscribe_topic_activity.rb
|-- swf_sns_activities.rb
|-- swf_sns_workflow.rb
|-- utils.rb
`-- wait_for_confirmation_activity.rb
```

Ahora, inicie el flujo de trabajo con el comando siguiente.

```
ruby swf_sns_workflow.rb
```

Esta operación iniciará el flujo de trabajo, y debería imprimir un mensaje con una línea que puede copiar y pegar en una nueva ventana de línea de comandos (o incluso en otro equipo, si ha copiado en él los archivos de origen del tutorial).

Amazon SWF Example

Start the activity worker, preferably in a separate command-line window, with the following command:

```
> ruby swf_sns_activities.rb 87097e76-7c0c-41c7-817b-92527bb0ea85-activities
```

You can copy & paste it if you like, just don't copy the '>' character.

Press return when you're ready...

El código de flujo de trabajo esperará pacientemente a que usted inicie el sondeador de actividad en una ventana diferente.

Abra una nueva ventana de línea de comandos, cambie de nuevo al directorio donde se encuentran los archivos de origen y utilice el comando provisto por el archivo `swf_sns_workflow.rb` para iniciar el sondeador de actividad. Por ejemplo, si recibe la salida anterior, escriba (o pegue) lo siguiente.

```
ruby swf_sns_activities.rb 87097e76-7c0c-41c7-817b-92527bb0ea85-activities
```

Una vez que comienza a ejecutar su sondeador de actividades, este comienza a producir información sobre el registro de actividades.

```
** initialized and registered activity: get_contact_activity
** initialized and registered activity: subscribe_topic_activity
** initialized and registered activity: wait_for_confirmation_activity
** initialized and registered activity: send_result_activity
```

Ahora puede volver a la ventana de línea de comandos original y pulsar Intro para comenzar la ejecución del flujo de trabajo. Registrará el flujo de trabajo y programará la primera actividad.

```
Starting workflow execution.
** registered workflow: swf-sns-workflow
** scheduling activity task: get_contact_activity
```

Vuelva a la otra ventana, donde se está ejecutando el sondeador de actividades. Se muestra el resultado de la primera actividad que se ejecuta, con una pregunta para que introduzca su correo electrónico o el número de teléfono SMS. Introduzca uno de ellos, o ambos, y confírmelos.

```
activity task received: <AWS::SimpleWorkflow::ActivityTask>
** Starting activity task: get_contact_activity
```

```
Please enter either an email address or SMS message (mobile phone) number to
receive Amazon SNS notifications. You can also enter both to use both address types.
```

```
If you enter a phone number, it must be able to receive SMS messages, and must
be 11 digits (such as 12065550101 to represent the number 1-206-555-0101).
```

```
Email: me@example.com
Phone: 12065550101
```

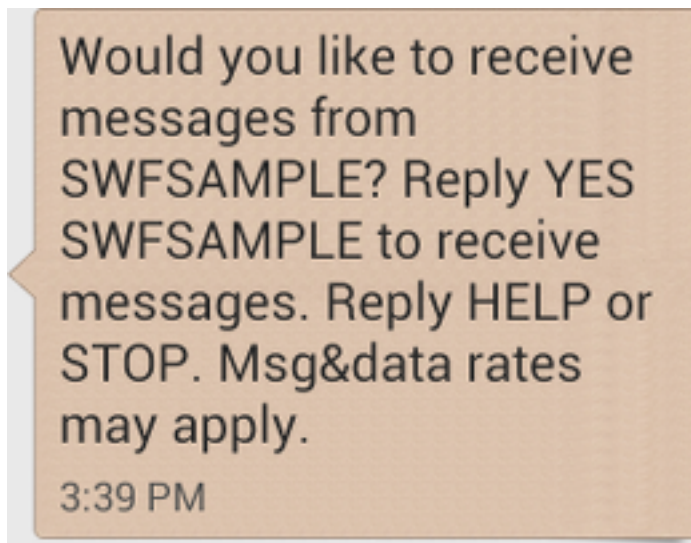
```
You entered:
  email: me@example.com
  phone: 12065550101
```

```
Is this correct? (y/n): y
```

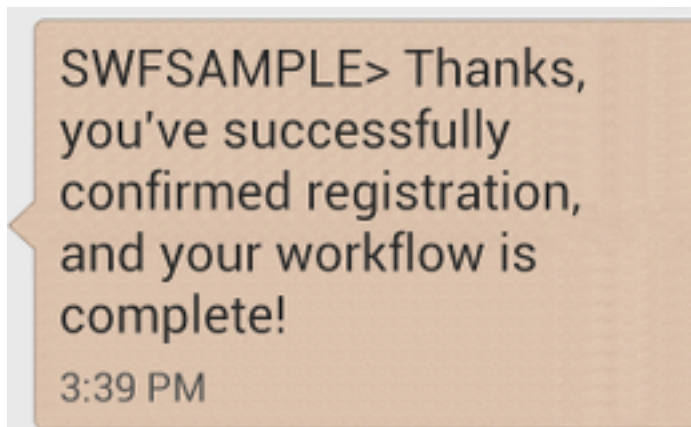
Note

El número de teléfono que se indica aquí es ficticio, solo tiene fines ilustrativos. Escriba su propio número de teléfono y dirección de correo electrónico aquí.

Al poco tiempo de introducir esta información, recibirá un correo electrónico o mensaje de texto de Amazon SNS en el que se le pedirá que confirme la suscripción al tema. Si introdujo un número de SMS, verá en su teléfono algo parecido a lo siguiente:



Si responde YES a este mensaje, recibirá la respuesta que hemos proporcionado en `send_result_activity`.



Mientras tenía lugar todo esto, ¿se fijó en lo que estaba ocurriendo en su ventana de línea de comandos? Tanto el sondeador de flujo de trabajo como el de actividad han estado muy ocupados.

Esta es la salida del sondeador de flujo de trabajo.

```
** scheduling activity task: subscribe_topic_activity
** scheduling activity task: wait_for_confirmation_activity
** scheduling activity task: send_result_activity
!! All activities complete! Sending complete_workflow_execution...
```

Esta es la salida del sondeador de actividades, que se ha generado al mismo tiempo en otra ventana de línea de comandos.

```
++ Activity task completed: get_contact_activity
** Starting activity task: subscribe_topic_activity
++ Activity task completed: subscribe_topic_activity
** Starting activity task: wait_for_confirmation_activity
Topic subscription still pending for (email: me@example.com)
Topic subscription confirmed for (sms: 12065550101)
++ Activity task completed: wait_for_confirmation_activity
** Starting activity task: send_result_activity
Thanks, you've successfully confirmed registration, and your workflow is complete!
++ Activity task completed: send_result_activity
All done!
```

Enhorabuena, su flujo de trabajo se ha completado y este tutorial ha finalizado.

Puede volver a ejecutar el flujo de trabajo para ver cómo funcionan los tiempos de espera, o probar a introducir datos diferentes. Recuerde que una vez que se suscriba a un tema, seguirá suscrito hasta que cancele la suscripción. Si vuelves a ejecutar el flujo de trabajo antes de cancelar la suscripción a los temas, es probable que se complete automáticamente, ya que `wait_for_confirmation_activity` verá que tu suscripción ya está confirmada.

Para cancelar la suscripción a un tema de Amazon SNS

- Responda de forma negativa (envíe STOP) al mensaje de texto.
- Elija el enlace para cancelar la suscripción que recibió en su correo electrónico.

Ahora ya puede volver a suscribirse al tema.

¿Qué tengo que hacer ahora?

Este tutorial ha abarcado muchos temas, pero aún hay mucho más que puedes aprender sobre Amazon SWF o Amazon SNS. AWS SDK para Ruby Para obtener más información y muchos más ejemplos, consulte la documentación oficial de cada producto:

- [Documentación de AWS SDK para Ruby](#)
- [Documentación de Amazon Simple Notification Service \(Amazon SNS\)](#)
- [Documentación de Amazon Simple Workflow Service](#)

Trabajo en la consola Amazon SWF

La consola Amazon SWF ofrece opciones para configurar, iniciar y gestionar las ejecuciones de flujos de trabajo.

Con la consola de Amazon SWF, puede:

- Registrar dominios de flujo de trabajo.
- Registre los tipos de flujo de trabajo y los tipos de actividad.
- Inicie, visualice, señalice, cancele, finalice y reinicie las ejecuciones del flujo de trabajo.

Registrar un dominio

Los flujos de trabajo se ejecutan en un AWS recurso denominado dominio, que controla el alcance del flujo de trabajo. Una cuenta de AWS puede tener varios dominios, y cada uno de ellos puede contener numerosos flujos de trabajo. No obstante, los flujos de trabajo de distintos dominios no pueden interactuar entre sí.

El registro de dominios es la única funcionalidad disponible inicialmente en la consola. Después de registrar al menos un dominio, puede realizar las siguientes acciones para el dominio:

- Registra los tipos de flujo de trabajo y actividad.
- Iniciar ejecuciones de flujos de trabajo.
- Cancelar, terminar y enviar señales para ejecutar ejecuciones de flujos de trabajo.
- Reiniciar ejecuciones de flujos de trabajo cerradas.

También puedes realizar acciones de administración de dominios, como dejar dominios obsoletos o no en desuso.

Después establecer un dominio como obsoleto, no podrá utilizarlo para crear nuevas ejecuciones de flujos de trabajo ni para registrar nuevos flujos de trabajo. Al establecer un dominio como obsoleto, también se quedarán en desuso todas las actividades y los flujos de trabajo registrados en dicho dominio. Las ejecuciones que se hubieran iniciado antes de que se estableciera el dominio como obsoleto seguirán ejecutándose.

Tras eliminar la obsolescencia de un dominio que anteriormente estaba en desuso, puedes volver a usarlo para registrar los tipos de flujo de trabajo e iniciar nuevas ejecuciones de flujos de trabajo.

Para obtener más información sobre estas acciones de administración de dominios, consulte y. [DeprecateDomainUndeprecateDomain](#)

Registrar tipos de flujo de trabajo

Puede registrar los tipos de flujo de trabajo en la consola de Amazon SWF después de haber registrado al menos un dominio.

Un tipo de flujo de trabajo es un conjunto de tipos de actividades que cumplen un objetivo y contienen la lógica que coordina las actividades. Los tipos de flujo de trabajo coordinan y gestionan la ejecución de actividades que se pueden ejecutar de forma asíncrona en varios dispositivos informáticos y cuentan con métodos de procesamiento secuencial y paralelo.

Para registrar un tipo de flujo de trabajo de Amazon SWF mediante la consola

1. Abra el dominio en el que desee registrar un flujo de trabajo.
2. Seleccione Registrar y, a continuación, seleccione Registrar flujo de trabajo.
3. En la página Registrar el flujo de trabajo, introduzca el Nombre del flujo de trabajo y la Versión del flujo de trabajo. Opcionalmente, también puede especificar una [Lista de tareas predeterminada](#) que se utilizará para programar las tareas de decisión para las ejecuciones de este flujo de trabajo.
4. (Opcional) Seleccione Opciones avanzadas para especificar los siguientes detalles para el flujo de trabajo:
 - [Prioridad de la tarea predeterminada](#): la prioridad de la tarea predeterminada que se va a asignar al flujo de trabajo.
 - [Tiempo de espera predeterminado de inicio a cierre de las ejecuciones](#): la duración máxima predeterminada de las ejecuciones de este flujo de trabajo.
 - [Tiempo de espera predeterminado de inicio a cierre de la tarea](#): la duración máxima predeterminada de las tareas de decisión para este flujo de trabajo.
 - [Política secundaria predeterminada](#): la política predeterminada que se utiliza para las ejecuciones de flujos de trabajo secundarios.
 - [Función de Lambda predeterminada](#): rol de IAM predeterminado asociado a este flujo de trabajo.
5. Seleccione Registrar flujo de trabajo.

Registrar los tipos de actividad

Las actividades son tareas que desea que su tipo de flujo de trabajo coordine y ejecute (por ejemplo: verificar el pedido del cliente, hacer un cargo a la tarjeta de crédito, etc.). El orden en que se realizan las actividades viene determinado por la lógica de coordinación del tipo de flujo de trabajo.

Puede registrar los tipos de actividad después de registrar al menos un dominio.

Cómo registrar un tipo de actividad de Amazon SWF con la consola

1. Abra el dominio en el que quiera registrar una actividad.
2. Seleccione Registrar y, a continuación, seleccione Registrar actividad.
3. En la página Registrar actividad, introduzca el [Nombre de la actividad](#) y la [Versión de la actividad](#). Opcionalmente, también puede especificar una [Lista de tareas predeterminada](#) que se utilizará para programar las tareas de esta actividad.
4. (Opcional) Seleccione Opciones avanzadas para especificar los siguientes detalles de la actividad:
 - [Prioridad de la tarea predeterminada](#): la prioridad de la tarea predeterminada que se va a asignar a la actividad.
 - [Programación de la tarea predeterminada para el tiempo de espera de inicio](#): el tiempo máximo predeterminado que puede esperar una tarea de esta actividad antes de asignarse a un proceso de trabajo.
 - [Tiempo de espera predeterminado de inicio a cierre de las tareas](#): la duración máxima predeterminada que puede tardar un proceso de trabajo en procesar las tareas de esta actividad.
 - [Programación de la tarea predeterminada para el tiempo de espera de cierre](#): la duración máxima predeterminada de una tarea de esta actividad.
 - Tiempo de [espera predeterminado para una tarea](#): es el tiempo máximo predeterminado antes del cual un trabajador que procese una tarea de este tipo debe informar del progreso mediante una llamada [RecordActivityTaskHeartbeat](#).
5. Seleccione Registrar actividad.

Iniciar un flujo de trabajo

Puede comenzar una ejecución de flujo de trabajo desde la consola de Amazon SWF. No puede comenzar una ejecución de flujo de trabajo hasta que haya registrado al menos un flujo de trabajo.

Para comenzar una ejecución de flujo de trabajo con la consola

1. Abra la consola de Amazon SWF y, en el panel de navegación izquierdo, seleccione Dominios.
2. Debajo del nombre de dominio, seleccione Flujos de trabajo.
3. En la página Flujos de trabajo, elija el flujo de trabajo que desee ejecutar.
4. Seleccione Iniciar ejecución.
5. En la página Iniciar la ejecución, introduzca el [Nombre del flujo de trabajo](#) y el ID de ejecución para identificar la ejecución por un nombre. Si lo desea, también puede especificar una [Lista de tareas](#) que se utilizará para las tareas de decisión generadas para la ejecución de este flujo de trabajo.
6. (Opcional) Seleccione Opciones avanzadas para especificar los siguientes detalles para la ejecución del flujo de trabajo:
 - [Prioridad de la tarea](#): la prioridad de la tarea que se va a utilizar para la ejecución de este flujo de trabajo.
 - [Tiempo de espera de inicio a cierre de la ejecución](#): la duración total de la ejecución de este flujo de trabajo.
 - [Tiempo de espera de inicio a cierre de las tareas](#): la duración máxima de las tareas de decisión para la ejecución de este flujo de trabajo.
 - [Política secundaria](#): la política que se utilizará para las ejecuciones de este flujo de trabajo secundario si se termina, mediante una llamada a la [TerminateWorkflowExecution](#) acción de forma explícita o si se ha agotado el tiempo de espera.
 - Función [Lambda: la función](#) de IAM que se va a adjuntar a la ejecución de este flujo de trabajo.
7. Seleccione Iniciar ejecución.

Gestión de las ejecuciones del flujo de

Puede filtrar las ejecuciones de sus flujos de trabajo por nombre, estado, ID y etiqueta. Puede enviar señales con entradas a las ejecuciones de flujos de trabajo activos. Si necesita cancelar o

finalizar un flujo de trabajo, puede utilizar la opción Intentar cancelar. La cancelación es preferible a la finalización de la ejecución de un flujo de trabajo, ya que la cancelación da al flujo de trabajo la oportunidad de realizar cualquier tarea de limpieza y, a continuación, cerrarse correctamente.

En la consola, puede gestionar las ejecuciones del flujo de trabajo que actualmente se están ejecutando de forma cerrada. and/or

Para administrar las ejecuciones de flujos de trabajo

1. Abra un dominio para administrar las ejecuciones de flujos de trabajo.
2. Seleccione Buscar ejecución.
3. En la página Ejecuciones del flujo de trabajo, elija Filtrar las ejecuciones por propiedad y, a continuación, en Propiedades, elija uno de los siguientes filtros:

Elija	Cómo aplicar este filtro
Flujo de trabajo	<p>Elija este filtro para ver una lista de las ejecuciones de un flujo de trabajo específico. Por ejemplo, para ver las ejecuciones de <code>fiction-books-order-workflow</code> , haga lo siguiente:</p> <ol style="list-style-type: none"> 1. Elija Flujo de trabajo. 2. En Operadores, elija Igual a. 3. En Flujos de trabajo, elija <code>fiction-books-order-workflow</code>. 4. (Opcional) Seleccione Borrar filtros para eliminar el filtro e iniciar una nueva búsqueda de ejecuciones.
Estado	<p>Elija este filtro para ver una lista de las ejecuciones con un estado específico. Por ejemplo, para ver las ejecuciones con el estado Error, haga lo siguiente:</p> <ol style="list-style-type: none"> 1. Elija Estado: 2. En Operadores, elija Igual a. 3. En Estados, elija Error. 4. (Opcional) Seleccione Borrar filtros para eliminar el filtro e iniciar una nueva búsqueda de ejecuciones.

Elija	Cómo aplicar este filtro
ID de ejecución	<p>Elija este filtro para ver la ejecución de un flujo de trabajo en función de su ID. Por ejemplo, para ver la ejecución con el ID <code>fiction-books-order-category1</code> , haga lo siguiente:</p> <ol style="list-style-type: none"> 1. Elija ID de ejecución. 2. En Operadores, elija Igual a. 3. En Ejecución IDs, elija <code>fiction-books-order-category1</code>. 4. (Opcional) Seleccione Borrar filtros para eliminar el filtro e iniciar una nueva búsqueda de ejecuciones.
Etiqueta	<p>Elija este filtro para ver una lista de las ejecuciones con una etiqueta específica. Por ejemplo, para ver las ejecuciones con la etiqueta <code>purchaseOrder</code> , haga lo siguiente:</p> <ol style="list-style-type: none"> 1. Seleccione Etiquetas. 2. En Operadores, elija Igual a. 3. En Etiqueta, seleccione <code>purchaseOrder</code>. 4. (Opcional) Seleccione Borrar filtros para eliminar el filtro e iniciar una nueva búsqueda de ejecuciones.

4. (Opcional) Tras aplicar el filtro necesario para ver una lista de las ejecuciones de flujos de trabajo, puede realizar las siguientes operaciones en una ejecución con estado Activo:
 - Señal: utilice esta opción para enviar datos adicionales sobre la ejecución de un flujo de trabajo en ejecución. Para ello:
 1. Elija la ejecución a la que desee enviar datos adicionales.
 2. Seleccione Señal y, a continuación, especifique los datos en el cuadro de diálogo Ejecución de señal.
 3. Elija Señal.
 - Probar-cancelar: utilice esta opción para intentar cancelar la ejecución de un flujo de trabajo. Es preferible cancelar una ejecución de flujo de trabajo a terminarla. La cancelación ofrece a la ejecución de flujo de trabajo una oportunidad de realizar tareas de limpieza y, a continuación, cerrarse correctamente.
 1. Elija la ejecución que desee cancelar.

2. Seleccione Probar-cancelar.
 - Terminar: utilice esta opción para terminar la ejecución de un flujo de trabajo. Tenga en cuenta que es preferible cancelar la ejecución de un flujo de trabajo a terminarla.
 1. Elija la ejecución que desee terminar.
 2. En el caso de Política secundaria, asegúrese de que esté seleccionada la opción Terminar.
 3. (Opcional) Especifique el Motivo y los Detalles para terminar la ejecución.
 4. Elija Terminar.
5. (Opcional) Volver a ejecutar: utilice esta opción para volver a ejecutar una ejecución de flujo de trabajo cerrada.
 1. En la lista de ejecuciones de flujo de trabajo, seleccione la ejecución cerrada que se va a volver a ejecutar. Al seleccionar una ejecución cerrada, se habilitará el botón Volver a ejecutar. Elija Volver a ejecutar.
 2. En la página Volver a ejecutar la ejecución, especifique los detalles de la ejecución del flujo de trabajo tal y como se menciona en [Iniciar un flujo de trabajo](#).

Conceptos básicos de flujo de trabajo en Amazon SWF

Note

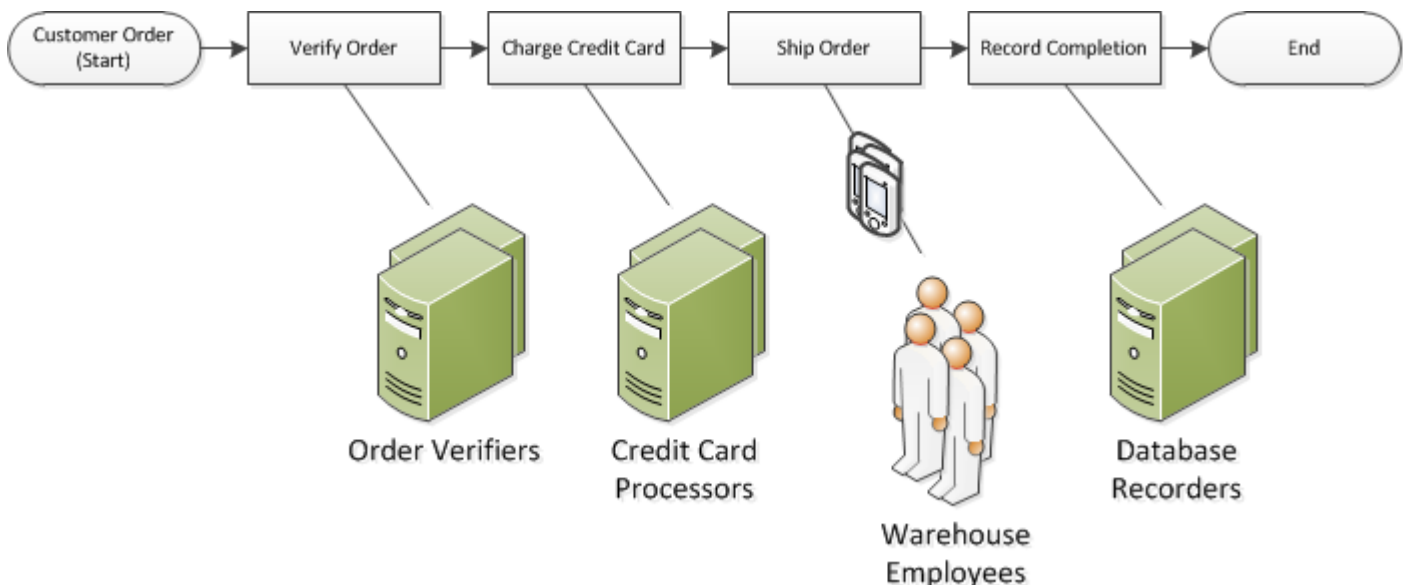
Los conceptos de este capítulo proporcionan información general acerca de Amazon Simple Workflow Service y describen sus características principales. Si busca ejemplos, consulte [Uso de Amazon SWF APIs](#).

Mediante Amazon Simple Workflow Service (Amazon SWF), puede implementar aplicaciones asíncronas distribuidas como flujos de trabajo. Los flujos de trabajo coordinan y administran la ejecución de actividades que se pueden ejecutar de forma asíncrona en varios dispositivos informáticos y que pueden realizar procesamiento secuencial y paralelo.

Al diseñar un flujo de trabajo, puede analizar su aplicación para identificar sus tareas de componente. En Amazon SWF, estas tareas se representan mediante actividades. El orden en el que se realizan las actividades viene determinado por la lógica de coordinación del flujo de trabajo.

Ejemplo de flujo de trabajo para una aplicación de comercio electrónico

La siguiente figura muestra un flujo de trabajo de procesamiento de pedidos de comercio electrónico en el que participan personas y procesos automatizados:



El flujo de trabajo de la aplicación de comercio electrónico comienza cuando un cliente realiza un pedido e incluye cuatro tareas:

1. Verifique el pedido.
2. Si el pedido es válido, cargue el cliente.
3. Si se realiza el pago, envíe el pedido.
4. Si se envía el pedido, guarde los detalles del pedido.

Las tareas de este flujo de trabajo son secuenciales: un pedido debe verificarse antes de que pueda cargarse una tarjeta de crédito; una tarjeta de crédito debe cargarse correctamente antes de que pueda enviarse un pedido y un pedido debe enviarse antes de que pueda registrarse. Sin embargo, como Amazon SWF admite procesos distribuidos, estas tareas se pueden llevar a cabo en diversas ubicaciones. Si las tareas tienen naturaleza programática, también pueden escribirse en diferentes lenguajes de programación o mediante diversas herramientas.

Además del procesamiento secuencial de tareas, Amazon SWF también admite flujos de trabajo con procesamiento paralelo de tareas. Las tareas paralelas se realizan al mismo tiempo y diferentes aplicaciones o procesos de trabajo humanos pueden llevarlas a cabo de forma independiente. Su flujo de trabajo toma decisiones acerca de cómo continuar una vez completadas una o más de las tareas paralelas.

Conceptos adicionales

- [Creación de un flujo de trabajo en Amazon SWF](#)
- [Ejecución de flujos de trabajo en Amazon SWF](#)
- [Historial del flujo de trabajo en Amazon SWF](#)
- [Identificadores de objetos en Amazon SWF](#)
- [Dominios en Amazon SWF](#)
- [Actores en Amazon SWF](#)
- [Tareas en Amazon SWF](#)
- [Listas de tareas en Amazon SWF](#)
- [Cierre de la ejecución del flujo de trabajo en Amazon SWF](#)
- [Ciclo de vida de un flujo de trabajo de Amazon SWF](#)
- [Sondeo de tareas en Amazon SWF](#)

Creación de un flujo de trabajo en Amazon SWF

La creación de un flujo de trabajo secuencial básico implica las siguientes etapas.

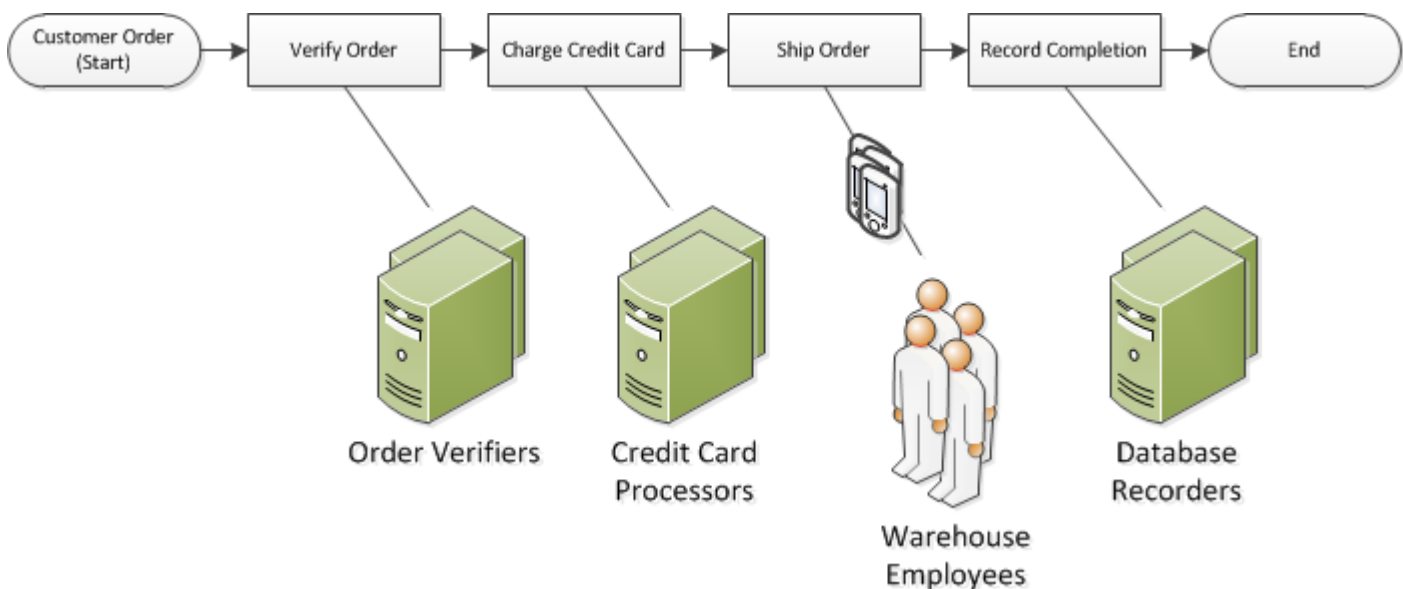
- Modelado de un flujo de trabajo, registro de su tipo y registro de sus tipos de actividad
- Desarrollo y lanzamiento de procesos de trabajo de actividad que realizan las tareas de actividad
- Desarrollo y lanzamiento de decisores que utilizan el historial del flujo de trabajo para determinar cómo proceder a continuación
- Desarrollo y lanzamiento de iniciadores del flujo de trabajo; es decir, aplicaciones que dan comienzo a las ejecuciones de flujos de trabajo

Modelado del flujo de trabajo y sus actividades

Para utilizar Amazon SWF, modele los pasos lógicos en la aplicación como actividades. Una actividad representa un solo paso lógico o tarea de su flujo de trabajo. Por ejemplo, la autorización de una tarjeta de crédito es una actividad que incluye proporcionar un número de tarjeta de crédito y otra información, y recibir un código de aprobación o un mensaje que indica que no se ha aceptado la tarjeta.

Además de la definición de actividades, también tiene que definir la lógica de coordinación que gestiona los puntos de decisión. Por ejemplo, la lógica de coordinación podría programar una actividad de seguimiento diferente en función de si se ha aceptado o rechazado la tarjeta de crédito.

La siguiente figura muestra un ejemplo de flujo de trabajo secuencial para el pedido de un cliente con cuatro actividades (verificación del pedido, cobro a la tarjeta de crédito, envío del pedido y registro de conclusión del pedido).



Ejecución de flujos de trabajo en Amazon SWF

Después de diseñar la lógica de coordinación y las actividades, puede registrar estos componentes como tipos de actividad y de flujo de trabajo en Amazon SWF. Durante el registro, se especifica un nombre, una versión y valores de configuración predeterminados para cada tipo.

Solo se pueden utilizar con Amazon SWF los tipos de actividad y de flujo de trabajo registrados. En el ejemplo del comercio electrónico, registraría el tipo `CustomerOrder` de flujo de trabajo y los tipos `VerifyOrder` `ChargeCreditCard` `ShipOrder`, y de `RecordCompletion` actividad.

Después de registrar su tipo de flujo de trabajo, puede ejecutarlo tanto como desee. Una ejecución de flujo de trabajo es una instancia en ejecución de un flujo de trabajo.

Cualquier proceso o aplicación puede iniciar una ejecución de flujo de trabajo, incluso otra ejecución de flujo de trabajo. En el ejemplo de E-Commerce, una nueva ejecución de flujo de trabajo se inicia con cada pedido del cliente. El tipo de aplicación que inicia el flujo de trabajo depende de la forma en que el cliente haga el pedido. Un sitio web o aplicación móvil, o bien un representante del servicio de atención al cliente podrían iniciar el flujo de trabajo mediante una aplicación de la empresa interna.

Con Amazon SWF, puede asociar un identificador, denominado `workflowId`, a las ejecuciones de flujo de trabajo, de modo que pueda integrar los identificadores existentes de su negocio en el flujo de trabajo. En el ejemplo de E-Commerce, cada ejecución de flujo de trabajo podría identificarse mediante el número de factura del cliente.

Además del identificador que proporcione, Amazon SWF asocia un identificador único generado por el sistema (un `runId`) a cada ejecución del flujo de trabajo. Amazon SWF solo permite ejecutar una única ejecución de flujo de trabajo con este identificador en un momento dado; aunque el usuario pueda ejecutar varios flujos de trabajo del mismo tipo, cada ejecución de flujo de trabajo tendrá un `runId` diferente.

Historial del flujo de trabajo en Amazon SWF

Amazon SWF registra el progreso de cada ejecución del flujo de trabajo en el historial del flujo de trabajo: un registro detallado, completo y coherente de todos los eventos que se han producido desde que se inició la ejecución del flujo de trabajo.

Un evento representa un cambio discreto en el estado de ejecución del flujo de trabajo, como la programación de una nueva actividad o la finalización de una actividad en ejecución. El historial del flujo de trabajo contiene todos los eventos que entrañan la modificación del estado de la ejecución

del flujo de trabajo, como actividades programadas y completadas, tiempos de espera de la tarea y señales.

Las operaciones que no modifican el estado de la ejecución del flujo de trabajo no suelen aparecer en el historial del flujo de trabajo. Por ejemplo, el historial del flujo de trabajo no muestra los intentos de sondeo o la utilización de operaciones de visibilidad.

El historial del flujo de trabajo ofrece varios beneficios clave:

- Las aplicaciones pueden no tener estado, ya que toda la información sobre la ejecución de un flujo de trabajo se almacena en su historial de flujo de trabajo.
- Para cada ejecución de flujo de trabajo, el historial proporciona un registro de qué actividades se programaron, su estado actual y sus resultados. La ejecución del flujo de trabajo utiliza esta información para determinar los siguientes pasos.
- El historial ofrece un seguimiento de auditoría detallado que puede utilizar para monitorizar las ejecuciones de flujos de trabajo en curso y verificar las ejecuciones de flujos de trabajo completadas.

A continuación se muestra una vista conceptual de un historial de flujo de trabajo de e-commerce:

```
Invoice0001
```

```
Start Workflow Execution
```

```
Schedule Verify Order
```

```
Start Verify Order Activity
```

```
Complete Verify Order Activity
```

```
Schedule Charge Credit Card
```

```
Start Charge Credit Card Activity
```

```
Complete Charge Credit Card Activity
```

```
Schedule Ship Order
```

```
Start Ship Order Activity
```

En el ejemplo anterior, el pedido está a la espera de ser enviado. En el siguiente ejemplo, el pedido se ha completado. Como el historial del flujo de trabajo es acumulativo, los eventos más nuevos se añaden al final:

```
Invoice0001
```

```
Start Workflow Execution

Schedule Verify Order
Start Verify Order Activity
Complete Verify Order Activity

Schedule Charge Credit Card
Start Charge Credit Card Activity
Complete Charge Credit Card Activity

Schedule Ship Order
Start Ship Order Activity

Complete Ship Order Activity

Schedule Record Order Completion
Start Record Order Completion Activity
Complete Record Order Completion Activity

Close Workflow
```

Mediante programación, los eventos del historial de ejecución del flujo de trabajo se representan como JavaScript objetos de notación de objetos (JSON). El historial en sí es una matriz JSON de estos objetos. Cada evento contiene lo siguiente:

- Un tipo, como o [WorkflowExecutionStartedActivityTaskCompleted](#)
- Una marca temporal en formato de tiempo Unix
- Un ID que identifica el evento de forma exclusiva

Además, cada tipo de evento tiene un conjunto distintivo de atributos descriptivos que son específicos de ese tipo. Por ejemplo, el `ActivityTaskCompleted` evento tiene atributos que contienen los IDs eventos que corresponden a la hora en que se programó la tarea de la actividad y al momento en que se inició, así como un atributo que contiene los datos de los resultados.

Puede obtener una copia del estado actual del historial de ejecución del flujo de trabajo mediante la [GetWorkflowExecutionHistory](#) acción. Además, como parte de la interacción entre Amazon SWF y el decisor del flujo de trabajo, el decisor recibe periódicamente copias del historial.

A continuación se muestra una sección de un historial de ejecución de un flujo de trabajo de ejemplo en formato JSON.

```
[ {
  "eventId": 11,
  "eventTimestamp": 1326671603.102,
  "eventType": "WorkflowExecutionTimedOut",
  "workflowExecutionTimedOutEventAttributes": {
    "childPolicy": "TERMINATE",
    "timeoutType": "START_TO_CLOSE"
  }
}, {
  "decisionTaskScheduledEventAttributes": {
    "startToCloseTimeout": "600",
    "taskList": {
      "name": "specialTaskList"
    }
  },
  "eventId": 10,
  "eventTimestamp": 1326670566.124,
  "eventType": "DecisionTaskScheduled"
}, {
  "activityTaskTimedOutEventAttributes": {
    "details": "Waiting for confirmation",
    "scheduledEventId": 8,
    "startedEventId": 0,
    "timeoutType": "SCHEDULE_TO_START"
  },
  "eventId": 9,
  "eventTimestamp": 1326670566.124,
  "eventType": "ActivityTaskTimedOut"
}, {
  "activityTaskScheduledEventAttributes": {
    "activityId": "verification-27",
    "activityType": {
      "name": "activityVerify",
      "version": "1.0"
    },
    "control": "digital music",
    "decisionTaskCompletedEventId": 7,
    "heartbeatTimeout": "120",
    "input": "5634-0056-4367-0923,12/12,437",
    "scheduleToCloseTimeout": "900",
    "scheduleToStartTimeout": "300",
    "startToCloseTimeout": "600",
    "taskList": {
```

```

        "name": "specialTaskList"
    }
},
"eventId": 8,
"eventTimestamp": 1326670266.115,
"eventType": "ActivityTaskScheduled"
}, {
"decisionTaskCompletedEventAttributes": {
    "executionContext": "Black Friday",
    "scheduledEventId": 5,
    "startedEventId": 6
},
"eventId": 7,
"eventTimestamp": 1326670266.103,
"eventType": "DecisionTaskCompleted"
}, {
"decisionTaskStartedEventAttributes": {
    "identity": "Decider01",
    "scheduledEventId": 5
},
"eventId": 6,
"eventTimestamp": 1326670161.497,
"eventType": "DecisionTaskStarted"
}, {
"decisionTaskScheduledEventAttributes": {
    "startToCloseTimeout": "600",
    "taskList": {
        "name": "specialTaskList"
    }
},
"eventId": 5,
"eventTimestamp": 1326668752.66,
"eventType": "DecisionTaskScheduled"
}, {
"decisionTaskTimedOutEventAttributes": {
    "scheduledEventId": 2,
    "startedEventId": 3,
    "timeoutType": "START_TO_CLOSE"
},
"eventId": 4,
"eventTimestamp": 1326668752.66,
"eventType": "DecisionTaskTimedOut"
}, {
"decisionTaskStartedEventAttributes": {

```

```
    "identity": "Decider01",
    "scheduledEventId": 2
  },
  "eventId": 3,
  "eventTimestamp": 1326668152.648,
  "eventType": "DecisionTaskStarted"
}, {
  "decisionTaskScheduledEventAttributes": {
    "startToCloseTimeout": "600",
    "taskList": {
      "name": "specialTaskList"
    }
  },
  "eventId": 2,
  "eventTimestamp": 1326668003.094,
  "eventType": "DecisionTaskScheduled"
}
]
```

Para obtener una lista detallada de los distintos tipos de eventos que pueden aparecer en el historial de ejecución del flujo de trabajo, consulte el tipo de [HistoryEvent](#) datos en la referencia de la API de Amazon Simple Workflow Service.

Amazon SWF almacena el historial completo de todas las ejecuciones del flujo de trabajo durante un número de días configurable después del cierre de la ejecución. Este periodo, que se conoce como periodo de retención del historial del flujo de trabajo, se especifica cuando se registra un dominio para el flujo de trabajo. Los dominios se detallan en más profundidad más adelante en esta sección.

Identificadores de objetos en Amazon SWF

En la siguiente lista, se describe cómo se identifican de forma exclusiva los objetos de Amazon SWF, como las ejecuciones de flujo de trabajo.

- Tipo de flujo de trabajo: un tipo de flujo de trabajo registrado se identifica por su dominio, nombre y versión. Los tipos de flujo de trabajo se especifican en la llamada a `RegisterWorkflowType`.
- Tipo de actividad: un tipo de actividad registrada se identifica por su dominio, nombre y versión. Los tipos de actividad se especifican en la llamada a `RegisterActivityType`.
- Tareas de decisión y tareas de actividad: cada tarea de decisión y de actividad se identifica por un token de tarea único. Amazon SWF genera el token de la tarea, que se devuelve con otra información sobre la tarea en la respuesta de `PollForDecisionTask` o

`PollForActivityTask`. Aunque el proceso que recibió la tarea usa el token con más frecuencia, el proceso podría transmitir el token a otro proceso, que podría notificar la finalización o un error de la tarea.

- Ejecución de flujo de trabajo: una sola ejecución de un flujo de trabajo se identifica por el dominio, el ID del flujo de trabajo y el ID de ejecución. Los dos primeros son parámetros a los que se pasa. [StartWorkflowExecution](#) `StartWorkflowExecution` devuelve el ID de ejecución.

Dominios en Amazon SWF

Los flujos de trabajo se ejecutan en un AWS recurso denominado dominio que proporciona una forma de determinar el alcance de los recursos de Amazon SWF de su cuenta. AWS Todos los componentes de un flujo de trabajo, como el tipo de flujo de trabajo y los tipos de actividad, deben especificarse para estar en un dominio.

Una cuenta de AWS puede tener varios dominios, y cada uno de ellos puede contener numerosos flujos de trabajo. No obstante, los flujos de trabajo de distintos dominios no pueden interactuar entre sí.

Cuando configure un nuevo flujo de trabajo, antes de configurar cualquiera de los otros componentes del flujo de trabajo, debe registrar un dominio si aún no lo ha hecho.

Al registrar un dominio, debe especificar un periodo de retención del historial del flujo de trabajo. El período de retención es el tiempo durante el que Amazon SWF seguirá conservando la información sobre la ejecución del flujo de trabajo una vez finalizada la ejecución del flujo de trabajo.

El registro de dominios es la única funcionalidad disponible inicialmente en la consola. Después de registrar al menos un dominio, puede realizar las siguientes acciones para el dominio:

- Registra los tipos de flujo de trabajo y actividad.
- Iniciar ejecuciones de flujos de trabajo.
- Cancelar, terminar y enviar señales para ejecutar ejecuciones de flujos de trabajo.
- Reiniciar ejecuciones de flujos de trabajo cerradas.

También puedes realizar acciones de administración de dominios, como dejar dominios obsoletos o no en desuso.

Después establecer un dominio como obsoleto, no podrá utilizarlo para crear nuevas ejecuciones de flujos de trabajo ni para registrar nuevos flujos de trabajo. Al establecer un dominio como obsoleto,

también se quedarán en desuso todas las actividades y los flujos de trabajo registrados en dicho dominio. Las ejecuciones que se hubieran iniciado antes de que se estableciera el dominio como obsoleto seguirán ejecutándose.

Tras eliminar la obsolescencia de un dominio que anteriormente estaba en desuso, puedes volver a usarlo para registrar los tipos de flujo de trabajo e iniciar nuevas ejecuciones de flujos de trabajo.

Para obtener más información sobre estas acciones de administración de dominios, consulte y [DeprecateDomainUndeprecateDomain](#)

Actores en Amazon SWF

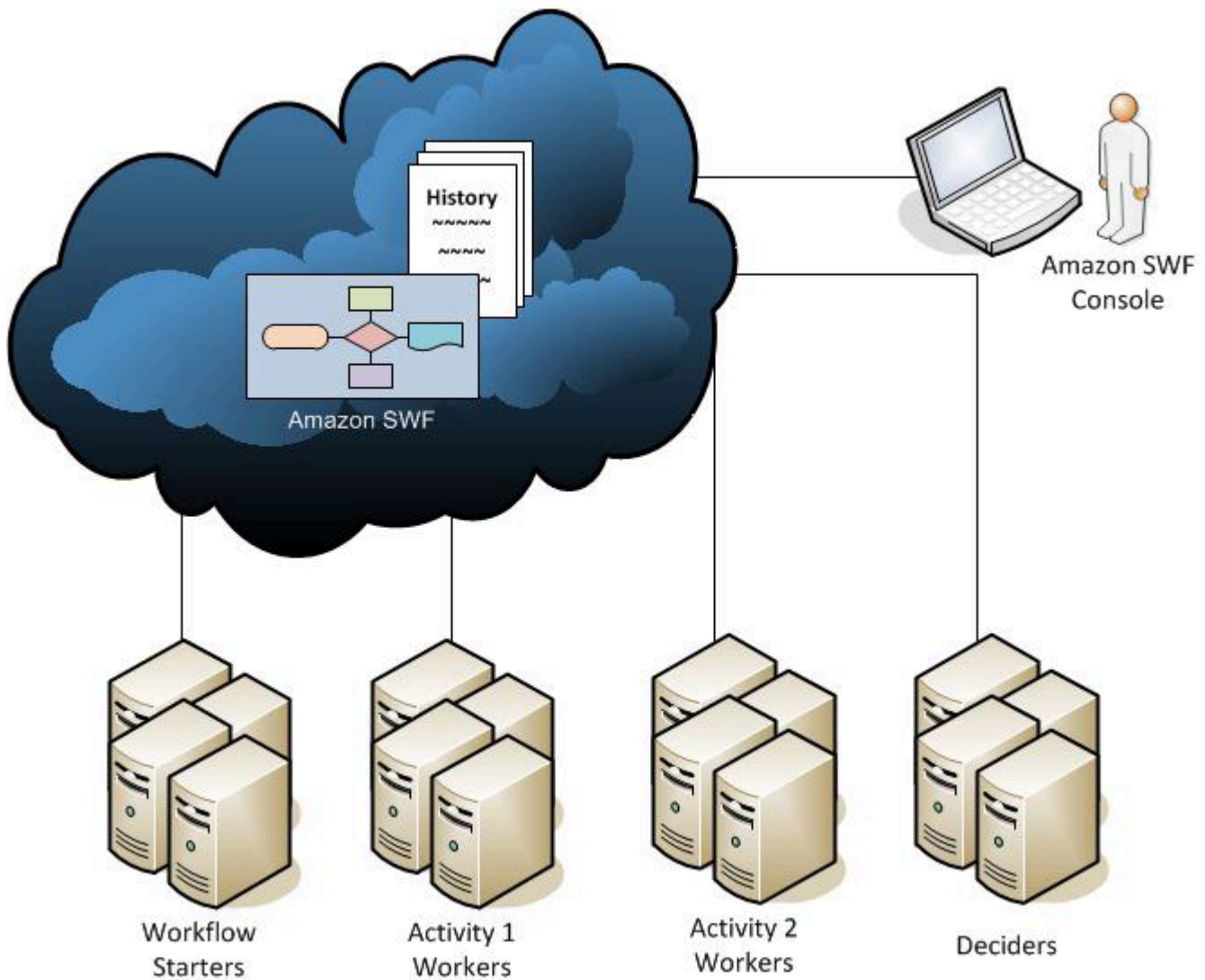
Temas

- [¿Qué es un actor en Amazon SWF?](#)
- [Iniciadores de flujo de trabajo](#)
- [Decisores](#)
- [Procesos de trabajo de actividad](#)
- [Intercambio de datos entre actores](#)

¿Qué es un actor en Amazon SWF?

En el curso de sus operaciones, Amazon SWF interactúa con varios tipos diferentes de actores programáticos. Los actores pueden ser [iniciadores de flujo de trabajo](#), [decisores](#) o [procesos de trabajo de actividad](#). Estos actores se comunican con Amazon SWF por medio de sus API. Puede desarrollar estos actores en cualquier lenguaje de programación.

En el siguiente diagrama, se muestra la arquitectura de Amazon SWF, incluido Amazon SWF y sus actores.



Iniciadores de flujo de trabajo

Un iniciador de flujo de trabajo es cualquier aplicación que permite iniciar ejecuciones de flujos de trabajo. En el ejemplo de e-commerce, un iniciador de flujo de trabajo puede ser el sitio web en el que el cliente realiza un pedido. Otro iniciador de flujo de trabajo puede ser un sistema o aplicación móvil usado por un representante de servicio al cliente para realizar un pedido en nombre del cliente.

Decisores

Un decisor es una implementación de la lógica de coordinación de un flujo de trabajo. Los decisores controlan el flujo de las tareas de actividad en una ejecución de flujo de trabajo. Siempre que se produce un cambio durante una ejecución de flujo de trabajo, como la finalización de una tarea, se transmite a un decisor una tarea de decisión que incluye todo el historial del flujo de trabajo. Cuando

el decisor recibe una tarea de decisión de Amazon SWF, analiza el historial de ejecución del flujo de trabajo para determinar los siguientes pasos apropiados en la ejecución del flujo de trabajo. El decisor comunica estos pasos a Amazon SWF mediante decisiones. Una decisión es un tipo de datos de Amazon SWF que puede representar varias acciones siguientes. Para obtener una lista de las posibles decisiones, consulte [Decisión](#) en la referencia de la API de Amazon Simple Workflow Service.

Este es un ejemplo de una decisión en formato JSON, que es el formato en el que se transmite a Amazon SWF. Esta decisión programa una tarea de actividad nueva.

```
{
  "decisionType" : "ScheduleActivityTask",
  "scheduleActivityTaskDecisionAttributes" : {
    "activityType" : {
      "name" : "activityVerify",
      "version" : "1.0"
    },
    "activityId" : "verification-27",
    "control" : "digital music",
    "input" : "5634-0056-4367-0923,12/12,437",
    "scheduleToCloseTimeout" : "900",
    "taskList" : {
      "name": "specialTaskList"
    },
    "scheduleToStartTimeout" : "300",
    "startToCloseTimeout" : "600",
    "heartbeatTimeout" : "120"
  }
}
```

El decisor recibe una tarea de decisión cuando comienza la ejecución del flujo de trabajo y cada vez que se produce un cambio en la ejecución del flujo de trabajo. Los decisores siguen impulsando la ejecución del flujo de trabajo mediante la recepción de tareas de decisión y el envío de respuestas a Amazon SWF con más decisiones, hasta que el decisor determine que la ejecución de flujo de trabajo se ha completado. Entonces responde con una decisión para cerrar la ejecución del flujo de trabajo. Después del cierre de la ejecución del flujo de trabajo, Amazon SWF no programará tareas adicionales para esa ejecución.

En el ejemplo de e-commerce, el decisor determina si cada paso se ha realizado correctamente, y luego programa el paso siguiente o aborda las condiciones de error si las hubiera.

Un decisor representa un proceso o subproceso informático único. Diversos decisores pueden procesar tareas para el mismo tipo de flujo de trabajo.

Procesos de trabajo de actividad

Un trabajo de actividad es un proceso o subproceso que realiza tareas de actividad que forman parte de su flujo de trabajo. La tarea de actividad representa una de las tareas que usted identificó en su aplicación.

Para utilizar una tarea de actividad en su flujo de trabajo, debe registrarla mediante la consola Amazon SWF o la [RegisterActivityType](#) acción.

Cada proceso de trabajo de actividad sondea Amazon SWF para obtener nuevas tareas que ese trabajo de actividad pueda realizar; ciertas tareas solo las pueden realizar determinados procesos de trabajo de actividad. Tras recibir una tarea, el proceso de trabajo de actividad la procesa hasta su finalización y, a continuación, comunica a Amazon SWF que la tarea se ha completado y proporciona el resultado. A continuación, el trabajo de actividad busca una nueva tarea. Los procesos de trabajo de actividad asociados con una ejecución de flujo de trabajo continúan procesando tareas de esta forma hasta que se complete la ejecución del flujo de trabajo. En el ejemplo de e-commerce, los procesos de trabajo de actividad son procesos y aplicaciones independientes utilizados por las personas, como procesadores de tarjeta de crédito y empleados de almacén, que realizan pasos individuales en el proceso.

Un trabajo de actividad representa un proceso (o subproceso) informático único. Diversos procesos de trabajo de actividad pueden procesar tareas del mismo tipo de actividad.

Intercambio de datos entre actores

Los datos de entradas pueden transmitirse a una ejecución de flujo de trabajo cuando se inicia. Igualmente, los datos de entrada pueden proporcionarse a los procesos de trabajo de actividad cuando estos programan las tareas de actividad. Cuando se completa una tarea de actividad, el proceso de trabajo de actividad puede devolver los resultados a Amazon SWF. Igualmente, el decisor puede comunicar los resultados de la ejecución del flujo de trabajo una vez completada dicha ejecución. Los actores pueden enviar datos a Amazon SWF o recibir datos de este a través de cadenas, cuya forma está definida por el usuario. En función del tamaño y la confidencialidad de los datos, puede transmitir los datos directamente o pasar un puntero a los datos almacenados en otro sistema o servicio (como Amazon S3 o DynamoDB). Tanto los datos transmitidos directamente como los punteros a otros almacenes de datos se registran en el historial de ejecución del flujo de trabajo;

sin embargo, Amazon SWF no copia ni guarda en la caché datos de almacenes externos como parte del historial.

Puesto que Amazon SWF mantiene el estado completo de cada ejecución del flujo de trabajo, incluidas las entradas y los resultados de las tareas, no es necesario que los actores tengan estado. Como resultado, el procesamiento del flujo de trabajo es altamente escalable. A medida que crece la carga en su sistema, simplemente puede añadir más actores para aumentar la capacidad.

Tareas en Amazon SWF

Amazon SWF interactúa con los procesos de trabajo de actividad y con los decisores al proporcionarles asignaciones de trabajo conocidas como tareas. Existen tres tipos diferentes de tareas en Amazon SWF:

- Tarea de actividad: una tarea de actividad pide a un proceso de trabajo de actividad que realice su función, como comprobar el inventario o efectuar un cargo a una tarjeta de crédito. La tarea de actividad contiene toda la información que necesita el trabajo de actividad para cumplir su función.
- Tarea de Lambda: una tarea de Lambda es parecida a una tarea de actividad, pero ejecuta una función de Lambda en lugar de una actividad tradicional de Amazon SWF. Para obtener más información sobre cómo definir una tarea de Lambda, consulte [AWS Lambda tareas en Amazon SWF](#).
- Tarea de decisión: una tarea de decisión indica a un decisor que el estado de una ejecución de flujo de trabajo ha cambiado para que el decisor pueda determinar la siguiente actividad que sea necesario realizar. La tarea de decisión contiene el historial del flujo de trabajo en curso.

Amazon SWF programa una tarea de decisión cuando comienza el flujo de trabajo y siempre que cambia el estado del flujo de trabajo, como cuando se completa una tarea de actividad. Todas las tareas de decisión contienen una vista paginada de todo el historial de ejecución del flujo de trabajo. El decisor analiza el historial de ejecución del flujo de trabajo y responde a Amazon SWF con un conjunto de decisiones que especifican lo que debe ocurrir a continuación en la ejecución del flujo de trabajo. Básicamente, cada tarea de decisión ofrece al decisor la oportunidad de evaluar el flujo de trabajo y de orientar a Amazon SWF en consecuencia.

Para garantizar que no se procesan decisiones contradictorias, Amazon SWF asigna cada tarea de decisión a un solo decisor y solo permite que esté activa una tarea de decisión en cada momento en una ejecución de flujo de trabajo.

La siguiente tabla muestra la relación entre los diferentes componentes relacionados con flujos de trabajo y decisores.

Diseño lógico	Registrado como	Realizado por	Recibe y realiza	Genera
Flujo de trabajo	Tipo de flujo de trabajo	Decisor	Tareas de decisión	Decisiones

Cuando un proceso de trabajo de actividad ha completado la tarea de la actividad, informa a Amazon SWF de que la tarea se ha completado e incluye todos los resultados relevantes que se hayan generado. Amazon SWF actualiza el historial de ejecución del flujo de trabajo con un evento que indica que la tarea se ha completado y, a continuación, programa una tarea de decisión para transmitir el historial actualizado al decisor.

Amazon SWF asigna cada tarea de actividad a un solo proceso de trabajo de actividad. Una vez que se asigna la tarea, ningún otro trabajo de actividad puede reclamar o realizar esa tarea.

La siguiente tabla muestra la relación entre los diferentes componentes relacionados con actividades.

Diseño lógico	Registrado como	Realizado por	Recibe y realiza	Genera
Actividad	Tipo de actividad	Trabajo de actividad	Tareas de actividad	Datos de resultados

Listas de tareas en Amazon SWF

Las listas de tareas proporcionan una forma de organizar las diversas tareas asociadas a un flujo de trabajo. Puede considerar las listas de tareas similares a las colas dinámicas. Al programar una tarea en Amazon SWF, se puede especificar una cola (lista de tareas) donde incluirla. De forma similar, al sondear Amazon SWF para detectar alguna tarea, se puede indicar la cola (lista de tareas) de la que se obtendrá la tarea.

Las listas de tareas proporcionan un mecanismo flexible para dirigir tareas a los procesos de trabajo como los requiera su caso de uso. Las listas de tareas son dinámicas en el sentido de que no es necesario registrar una lista de tareas o crearla de forma explícita a través de una acción: basta con programar una tarea para que se cree la lista de tareas si no existe todavía.

Hay listas separadas de tareas de actividad y tareas de decisión. Una tarea siempre se programa solo en una lista de tareas; las tareas no se comparten entre listas. Además, al igual que las actividades y los flujos de trabajo, las listas de tareas se centran en una AWS región concreta y en un dominio de Amazon SWF.

Temas

- [Listas de tareas de decisión](#)
- [Listas de tareas de actividad](#)
- [Direccionamiento de tareas](#)

Listas de tareas de decisión

Cada ejecución de flujo de trabajo se asocia a una lista de tareas de decisión específica. Cuando se registra un tipo de flujo de trabajo ([RegisterWorkflowType](#) acción), puede especificar una lista de tareas predeterminada para las ejecuciones de ese tipo de flujo de trabajo. Cuando el iniciador del flujo de trabajo inicia la ejecución de flujo de trabajo (acción `StartWorkflowExecution`), tiene la opción de especificar una lista de tareas diferente para esa ejecución de flujo de trabajo.

Cuando un decisor sondea una nueva tarea de decisión (acción `PollForDecisionTask`), el decisor especifica una lista de tareas de decisión en la que basarse. Un solo decisor podría atender varias ejecuciones de flujo de trabajo llamando a `PollForDecisionTask` varias veces, mediante una lista de tareas diferente en cada llamada, donde cada lista de tareas es específica de una ejecución de flujo de trabajo determinada. De forma alternativa, el decisor podría sondear una sola lista de tareas de decisión que proporcione tareas de decisión para varias ejecuciones de flujo de trabajo. También podría disponer de varios decisores que atiendan una sola ejecución de flujo de trabajo sondeando todos ellos la lista de tareas para esa ejecución de flujo de trabajo.

Listas de tareas de actividad

Una sola lista de tareas de actividad puede incluir tareas de diferentes tipos de actividad. Las tareas se programan en el orden en el que aparezcan en la lista de tareas. Amazon SWF devuelve las tareas de la lista ordenadas en la medida de lo posible. En determinadas circunstancias, es posible que las listas no den como resultado la lista en orden.

Cuando se registra un tipo de actividad ([RegisterActivityType](#) acción), puede especificar una lista de tareas predeterminada para ese tipo de actividad. De forma predeterminada, las tareas de actividad de este tipo se programarán en la lista de tareas especificada; sin embargo, cuando quien decide

programar una tarea de actividad ([ScheduleActivityTask](#)decisión), puede especificar opcionalmente una lista de tareas diferente en la que programar la tarea. Si el decisor no especifica una lista de tareas, se usará la lista de tareas predeterminada. Como resultado, puede colocar tareas de actividad en listas de tareas específicas según los atributos de la tarea. Por ejemplo, podría colocar todas las instancias de una tarea de actividad para un tipo de tarjeta de crédito determinado en una lista de tareas en particular.

Direccionamiento de tareas

Cuando un trabajador que realiza una actividad realiza una encuesta sobre una nueva tarea ([PollForActivityTask](#)acción), puede especificar una lista de actividades y tareas de la que extraer información. Si es así, el proceso de trabajo de actividad aceptará tareas solo de esa lista. De esta forma, puede garantizar que se asignen determinadas tareas solo a procesos de trabajo de actividad. Por ejemplo, podría crear una lista de tareas que contenga tareas que requieren el uso de un equipo de alto desempeño. Solo los procesos de trabajo de actividad que se ejecutan en el hardware adecuado sondearían esa lista de tareas. Otro ejemplo sería crear una lista de tareas para una región geográfica determinada. Podría entonces garantizar que solo los procesos de trabajo implementados en esa región incluyeran esas tareas. O bien, podría crear una lista de tareas para pedidos de alta prioridad y comprobar siempre esa lista primero.

La asignación de determinadas tareas a procesos de trabajo de actividad en particular de esta forma se llama direccionamiento de tareas. El direccionamiento de tareas es opcional; si no especifica una lista de tareas al programar una actividad, la tarea se coloca automáticamente en la lista de tareas predeterminada.

Cierre de la ejecución del flujo de trabajo en Amazon SWF

Una vez que se inicia la ejecución de un flujo de trabajo, se abre. Una ejecución de flujo de trabajo abierta podría cerrarse como completada, cancelada, con error o como tiempo de espera agotado. También podría continuarse como nueva ejecución, o bien podría terminarse. El decisor, la persona que administra el flujo de trabajo o Amazon SWF pueden cerrar una ejecución de flujo de trabajo.

Si el decisor determina que las actividades del flujo de trabajo han finalizado, debería cerrar la ejecución del flujo de trabajo como completada mediante la acción [RespondDecisionTaskCompleted](#) y pasar la decisión [CompleteWorkflowExecution](#).

Asimismo, un decisor podría cerrar la ejecución de flujo de trabajo como cancelada o con error. Para cancelar la ejecución, el decisor debería usar la acción [RespondDecisionTaskCompleted](#) y pasar la decisión [CancelWorkflowExecution](#).

El decisor debe determinar como error las ejecuciones de flujo de trabajo cuyo estado quede fuera del ámbito de finalización normal. Para dar como errónea la ejecución, el decisor deberá usar la acción `RespondDecisionTaskCompleted` y pasar la decisión [FailWorkflowExecution](#).

Amazon SWF monitorea las ejecuciones de flujos de trabajo para garantizar que no superen ningún ajuste de tiempo de espera especificado por el usuario. Si se agota el tiempo de espera de una ejecución de flujo de trabajo, Amazon SWF la cierra automáticamente. Para obtener más información acerca de los valores de tiempo de espera, consulte la sección [Tipos de tiempo de espera de Amazon SWF](#).

Asimismo, un decisor puede cerrar la ejecución y, de forma lógica, continuarla como nueva ejecución mediante la acción `RespondDecisionTaskCompleted` y pasando la decisión [ContinueAsNewWorkflowExecution](#). Esta es una estrategia útil para ejecuciones de flujo de trabajo de ejecución prolongada en las que es posible que el historial crezca demasiado.

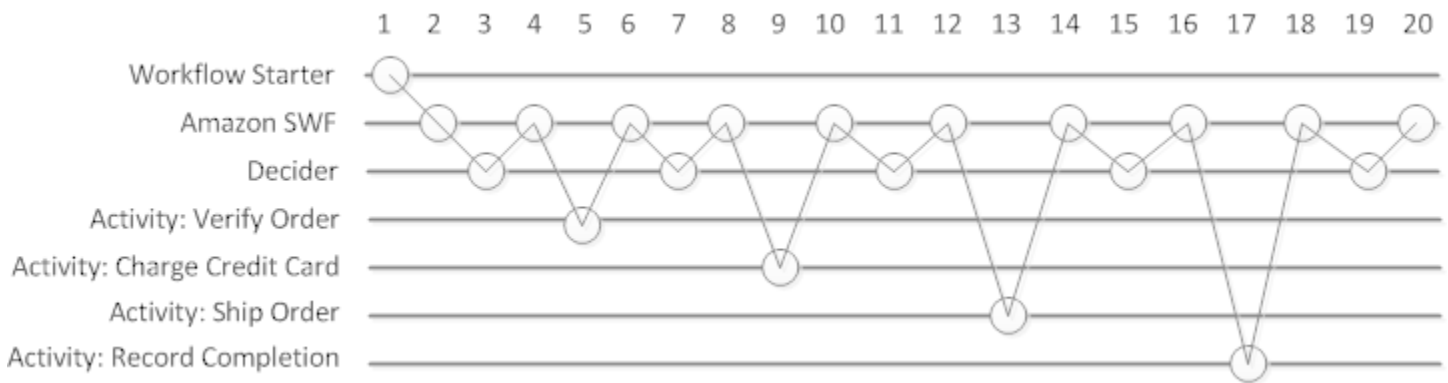
Por último, podría terminar las ejecuciones de flujo de trabajo directamente en la consola de Amazon SWF o mediante programación a través de la API [TerminateWorkflowExecution](#). La terminación fuerza el cierre de la ejecución de flujo de trabajo. Se prefiere la cancelación a la terminación, ya que sus decisores pueden administrar el cierre de la ejecución de flujo de trabajo.

Amazon SWF finaliza la ejecución de un flujo de trabajo si la ejecución supera determinados límites definidos por el servicio. Amazon SWF finaliza un flujo de trabajo secundario si el flujo de trabajo principal ha terminado y si la política secundaria aplicable indica que el flujo de trabajo secundario también debe darse por terminado.

Ciclo de vida de un flujo de trabajo de Amazon SWF

Desde el inicio de una ejecución de flujo de trabajo hasta su finalización, Amazon SWF interactúa con actores al asignarles las tareas adecuadas, ya sean tareas de actividad o de decisión.

En el siguiente diagrama se muestra el ciclo de vida de una ejecución de flujo de trabajo de procesamiento de pedidos desde la perspectiva de los componentes que actúan en ella.



Ciclo de vida de ejecución de flujo de trabajo

En la siguiente tabla se explica cada tarea de la imagen anterior.

Descripción	Acción, decisión o evento
1. El iniciador del flujo de trabajo llama a la acción de Amazon SWF adecuada para iniciar la ejecución del flujo de trabajo para un pedido, y proporciona la información de dicho pedido.	Acción StartWorkflowExecution .
2. Amazon SWF recibe la solicitud de inicio de ejecución de flujo de trabajo y, a continuación, programa la primera tarea de decisión.	Eventos WorkflowExecutionStarted y DecisionTaskScheduled .
3. El decisor recibe la tarea de Amazon SWF, revisa el historial, aplica la	Acción PollForDecisionTask . Acción RespondDecisionTaskCompleted y decisión ScheduleActivityTask .

Descripción	Acción, decisión o evento
<p>lógica de coordinación para determinar que no se ha producido ninguna actividad anterior, toma la decisión de programar la actividad de verificación del pedido con la información necesaria para que el proceso de trabajo de actividad procese la tarea, y devuelve la decisión a Amazon SWF.</p>	
<p>4. Amazon SWF recibe la decisión, programa la tarea de actividad de verificación del pedido y espera a que la tarea de actividad se complete o a que se agote su tiempo de espera.</p>	<p>ActivityTaskScheduled event</p>
<p>5. Un proceso de trabajo de actividad que puede realizar la actividad de verificación del pedido recibe la tarea, la realiza y devuelve los resultados a Amazon SWF.</p>	<p>Acciones PollForActivityTask y RespondActivityTaskCompleted.</p>

Descripción	Acción, decisión o evento
6. Amazon SWF recibe los resultados de la actividad de verificación del pedido, los añade al historial del flujo de trabajo y programa una tarea de decisión.	Eventos ActivityTaskCompleted y DecisionTaskScheduled .
7. El decisor recibe la tarea de Amazon SWF, revisa el historial, aplica la lógica de coordinación, toma la decisión de programar ChargeCreditCard una tarea de actividad con la información que el trabajador de la actividad necesita para procesar la tarea y devuelve la decisión a Amazon SWF.	Acción PollForDecisionTask . Acción RespondDecisionTaskCompleted con la decisión ScheduleActivityTask .
8. Amazon SWF recibe la decisión, programa la tarea de ChargeCreditCard actividad y espera a que se complete o se agote el tiempo de espera.	Eventos DecisionTaskCompleted y ActivityTaskScheduled .

Descripción	Acción, decisión o evento
9. Un trabajador de la actividad que puede realizar la ChargeCreditCard actividad recibe la tarea, la ejecuta y devuelve los resultados a Amazon SWF.	Acciones PollForActivityTask y RespondActivityTaskCompleted .
10. Amazon SWF recibe los resultados de la tarea de ChargeCreditCard actividad, los añade al historial del flujo de trabajo y programa una tarea de decisión.	Eventos ActivityTaskCompleted y DecisionTaskScheduled .
11. El decisor recibe la tarea de Amazon SWF, revisa el historial, aplica la lógica de coordinación, toma la decisión de programar ShipOrder una tarea de actividad con la información que el trabajador de la actividad necesita para realizar la tarea y devuelve la decisión a Amazon SWF.	Acción PollForDecisionTask . RespondDecisionTaskCompleted con la decisión ScheduleActivityTask .

Descripción	Acción, decisión o evento
12. Amazon SWF recibe la decisión, programa una tarea de ShipOrder actividad y espera a que se complete o se agote el tiempo de espera.	Eventos DecisionTaskCompleted y ActivityTaskScheduled .
13. Un trabajador de la actividad que puede realizar la ShipOrder actividad recibe la tarea, la ejecuta y devuelve los resultados a Amazon SWF.	Acciones PollForActivityTask y RespondActivityTaskCompleted .
14. Amazon SWF recibe los resultados de la tarea de ShipOrder actividad, los añade al historial del flujo de trabajo y programa una tarea de decisión.	Eventos ActivityTaskCompleted y DecisionTaskScheduled .

Descripción	Acción, decisión o evento
15. El decisor recibe la tarea de Amazon SWF, revisa el historial, aplica la lógica de coordinación, toma la decisión de programar <code>RecordCompletion</code> una tarea de actividad con la información que el trabajador de la actividad necesita para realizar la tarea y devuelve la decisión a Amazon SWF.	Acción PollForDecisionTask . Acción RespondDecisionTaskCompleted con la decisión ScheduleActivityTask .
16. Amazon SWF recibe la decisión, programa una tarea de <code>RecordCompletion</code> actividad y espera a que se complete o se agote el tiempo de espera.	Eventos DecisionTaskCompleted y ActivityTaskScheduled .
17. Un trabajador de la actividad que puede realizar la <code>RecordCompletion</code> actividad recibe la tarea, la ejecuta y devuelve los resultados a Amazon SWF.	Acciones PollForActivityTask y RespondActivityTaskCompleted .

Descripción	Acción, decisión o evento
18. Amazon SWF recibe los resultados de la tarea de RecordCompletion actividad, los añade al historial del flujo de trabajo y programa una tarea de decisión.	Eventos ActivityTaskCompleted y DecisionTaskScheduled .
19. El decisor recibe la tarea de Amazon SWF, revisa el historial, aplica la lógica de coordinación, toma la decisión de cerrar la ejecución de flujo de trabajo y devuelve la decisión junto con cualquier resultado a Amazon SWF.	Acción PollForDecisionTask . Acción RespondDecisionTaskCompleted con la decisión CompleteWorkflowExecution .
20. Amazon SWF cierra la ejecución de flujo de trabajo y archiva el historial para poder consultarlo en el futuro.	Evento WorkflowExecutionCompleted .

Sondeo de tareas en Amazon SWF

Los decisores y los procesos de trabajo de actividad se comunican con Amazon SWF mediante el sondeo largo. El decisor o el proceso de trabajo de actividad inicia de forma periódica la

comunicación con Amazon SWF, en la que notifica a Amazon SWF su disponibilidad para aceptar una tarea y, a continuación, especifica una lista de tareas de la que obtener tareas.

Si una tarea está disponible en la lista de tareas especificada, Amazon SWF la devuelve inmediatamente en la respuesta. Si no hay ninguna tarea disponible, Amazon SWF mantiene la conexión TCP abierta durante un máximo de 60 segundos, de modo que, si aparece una nueva tarea disponible en ese tiempo, se puede devolver en la misma conexión. Si no hay ninguna tarea disponible en 60 segundos, devuelve una respuesta vacía y cierra la conexión (una respuesta vacía es una estructura de tareas en la que el valor de `taskToken` es una cadena vacía). Si esto ocurre, el decisor o el proceso de trabajo de la actividad debe volver a realizar un sondeo.

El sondeo largo funciona bien para el procesamiento de tareas de gran volumen. Los decisores y los procesos de trabajo de la actividad pueden administrar su propia capacidad, que resulta fácil de usar si estos se encuentran detrás de un firewall.

Para obtener más información, consulte [Sondeo de tareas de decisión](#) y [Sondeo de tareas de actividad](#).

Conceptos de flujo de trabajo avanzados en Amazon SWF

El ejemplo de E-Commerce de la sección [???](#) representa un escenario de flujo de trabajo simplificado. En realidad, es probable que desee que su flujo de trabajo realice tareas simultáneas (enviar un email de confirmación de pedido al mismo tiempo que se autoriza una tarjeta de crédito), registre eventos principales (todos los elementos se empaquetan), actualice el pedido con cambios (añadir o quitar un elemento) y tome otras decisiones más avanzadas como parte de la ejecución de flujo de trabajo. En esta sección se describen los conceptos avanzados de flujo de trabajo que puede utilizar para crear sus flujos de trabajo.

Conceptos avanzados

- [Control de versiones](#)
- [Señales](#)
- [Flujos de trabajo secundarios en Amazon SWF](#)
- [Marcadores en Amazon SWF](#)
- [Etiquetas en Amazon SWF](#)
- [Implementación de opciones exclusivas con Amazon SWF](#)
- [Temporizadores en Amazon SWF](#)
- [Cancelación de tareas de actividad en Amazon SWF](#)

Control de versiones

Las necesidades del negocio suelen exigirle tener diversas implementaciones o variaciones del mismo flujo de trabajo o actividad ejecutándose simultáneamente. Por ejemplo, es posible que desee probar una nueva implementación de un flujo de trabajo mientras otro se encuentra en producción. También podría desear ejecutar dos implementaciones diferentes con dos conjuntos de características distintos, como una implementación básica y premium. El control de versiones le permite ejecutar varias implementaciones de flujos de trabajo y actividades simultáneamente, para cualquier fin que cumpla sus requisitos.

Los tipos de flujo de trabajo y actividad tienen una versión asociada a ellos que se especifica en el momento del registro. La versión es una cadena de formato libre y puede elegir su propio esquema de control de versiones. Para crear una nueva versión de un tipo registrado, debe registrarlo con el mismo nombre y otra versión. Las [Listas de tareas en Amazon SWF](#), que se han

descrito anteriormente, pueden ayudarle a implementar mejor el control de versiones. Imagine una situación en la que tenga ejecuciones de flujo de trabajo de ejecución prolongada de un tipo determinado en curso y las circunstancias requieran que revise el flujo de trabajo, con miras a añadir una nueva característica. Podría implementar la nueva característica creando nuevas versiones de tipos de actividad y procesos de trabajo, además de un nuevo decisor. A continuación, podría lanzar ejecuciones de la nueva versión del flujo de trabajo mediante un conjunto diferente de listas de tareas. De esta forma, podría tener ejecuciones de flujos de trabajo de versiones diferentes ejecutándose de manera simultánea sin afectarse mutuamente.

Señales

Las señales le permiten inyectar información en una ejecución de flujo de trabajo en ejecución. En algunos casos, puede ser conveniente añadir información a una ejecución de flujo de trabajo en ejecución para informar a esta de que algo ha cambiado o de un evento externo. Cualquier proceso puede enviar una señal a una ejecución de flujo de trabajo abierta. Por ejemplo, una ejecución de flujo de trabajo podría señalar otra.

Note

Un intento de envío de una señal a una ejecución de flujo de trabajo que no está abierta hace que `SignalWorkflowExecution` experimente errores con `UnknownResourceFault`.

Para utilizar señales, defina el nombre de la señal y los datos que se van a transferir a esta, si los hay. A continuación, programe el decisor para que reconozca el evento de señal ([WorkflowExecutionSignaled](#)) en el historial y lo procese adecuadamente. Cuando un proceso quiere señalar la ejecución de un flujo de trabajo, realiza una llamada a Amazon SWF (mediante la [SignalWorkflowExecution](#) acción o, en el caso de un decisor, mediante la [SignalExternalWorkflowExecution](#) decisión) que especifica el identificador de la ejecución del flujo de trabajo de destino, el nombre de la señal y los datos de la señal. A continuación, Amazon SWF recibe la señal, la registra en el historial de ejecución del flujo de trabajo de destino y programa una tarea de decisión para ella. Si el decisor recibe la tarea de decisión, también recibirá la señal dentro del historial de ejecución del flujo de trabajo. Es entonces cuando el decisor puede tomar las acciones adecuadas según la señal y sus datos.

En ocasiones, es posible que quiera esperar a recibir una señal. Por ejemplo, para cancelar un pedido, un usuario puede enviar una señal, pero solo en el plazo de una hora desde el momento en que realizó el pedido. Amazon SWF no tiene un tipo primitivo que permita a un decisor esperar a

recibir una señal del servicio. La funcionalidad de pausa debe implementarse en el propio decisor. Para hacer una pausa, el decisor debería iniciar un temporizador mediante la decisión `StartTimer`, que especifica la duración de la espera de la señal por parte del decisor mientras se sigue realizando el sondeo de tareas de decisión. Al recibir el decisor una tarea de decisión, debería comprobar el historial para ver si se ha recibido la señal o si se ha iniciado el temporizador. Si se ha recibido la señal, el decisor debería cancelar el temporizador. Sin embargo, si por el contrario, se ha iniciado el temporizador, significa que la señal no llegó en el periodo de tiempo especificado. Para resumir, a fin de esperar a recibir una señal específica, haga lo siguiente.

1. Cree un temporizador durante el tiempo que debería esperar el decisor.
2. Al recibirse una tarea de decisión, compruebe el historial para ver si ha llegado la señal o si se ha iniciado el temporizador.
3. Si ha llegado una señal, cancele el temporizador mediante una decisión `CancelTimer` y procese la señal. Dependiendo de los tiempos, el historial puede contener los eventos `TimerFired` y `WorkflowExecutionSignaled`. En estos casos, puede confiar en el orden relativo de los eventos del historial para determinar qué ocurrió primero.
4. Si se ha iniciado el temporizador, antes de recibirse una señal, el decisor ha agotado el tiempo de espera de la señal. Puede producir un error en la ejecución o llevar a cabo cualquier otra lógica que sea adecuada para su caso de uso.

Para los casos en los que deba cancelarse un flujo de trabajo, por ejemplo, si el cliente canceló el pedido, debe utilizarse la acción `RequestCancelWorkflowExecution` en lugar de enviar una señal al flujo de trabajo.

Algunas aplicaciones para señales incluyen las siguientes acciones:

- Detener el progreso de ejecuciones de flujo de trabajo hasta que se reciba una señal (p. ej., esperar un envío del inventario).
- Proporcionar información a una ejecución de flujo de trabajo que podría afectar a la lógica de la toma de decisiones por parte de los decisores. Esto resulta útil para los flujos de trabajo afectados por eventos externos (p. ej., intentar finalizar la venta de una acción tras cerrar el mercado).
- Actualizar una ejecución de flujo de trabajo si prevé que pueden producirse cambios (p. ej., cambiar los volúmenes de pedidos después de realizarse un pedido y antes de enviarse).

En el siguiente ejemplo, se envía una señal a la ejecución de flujo de trabajo para cancelar un pedido.

```
https://swf.us-east-1.amazonaws.com
SignalWorkflowExecution
{"domain": "867530901",
 "workflowId": "20110927-T-1",
 "runId": "f5ebbac6-941c-4342-ad69-dfd2f8be6689",
 "signalName": "CancelOrder",
 "input": "order 3553"}
```

Si la ejecución del flujo de trabajo recibe la señal, Amazon SWF devolverá una respuesta HTTP correcta similar a la siguiente: Amazon SWF generará una tarea de decisión para informar al decisor de que procese la señal.

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/json
x-amzn-RequestId: bf78ae15-3f0c-11e1-9914-a356b6ea8bdf
```

Flujos de trabajo secundarios en Amazon SWF

Los flujos de trabajo complejos pueden descomponerse en componentes más pequeños, más gestionables y potencialmente reutilizables mediante flujos de trabajo secundarios. Un flujo de trabajo secundario es una ejecución de flujo de trabajo iniciada por otra ejecución de flujo de trabajo (principal). Para iniciar un flujo de trabajo secundario, el decisor del flujo de trabajo principal usa la decisión `StartChildWorkflowExecution`. Los datos de entrada especificados con esta decisión están disponibles para el flujo de trabajo secundario a través de su historial.

Los atributos para la decisión `StartChildWorkflowExecution` también especifican la política secundaria, es decir, de qué forma debería Amazon SWF controlar la situación en la que la ejecución del flujo de trabajo principal termina antes de la ejecución del flujo de trabajo secundario. Hay tres valores posibles:

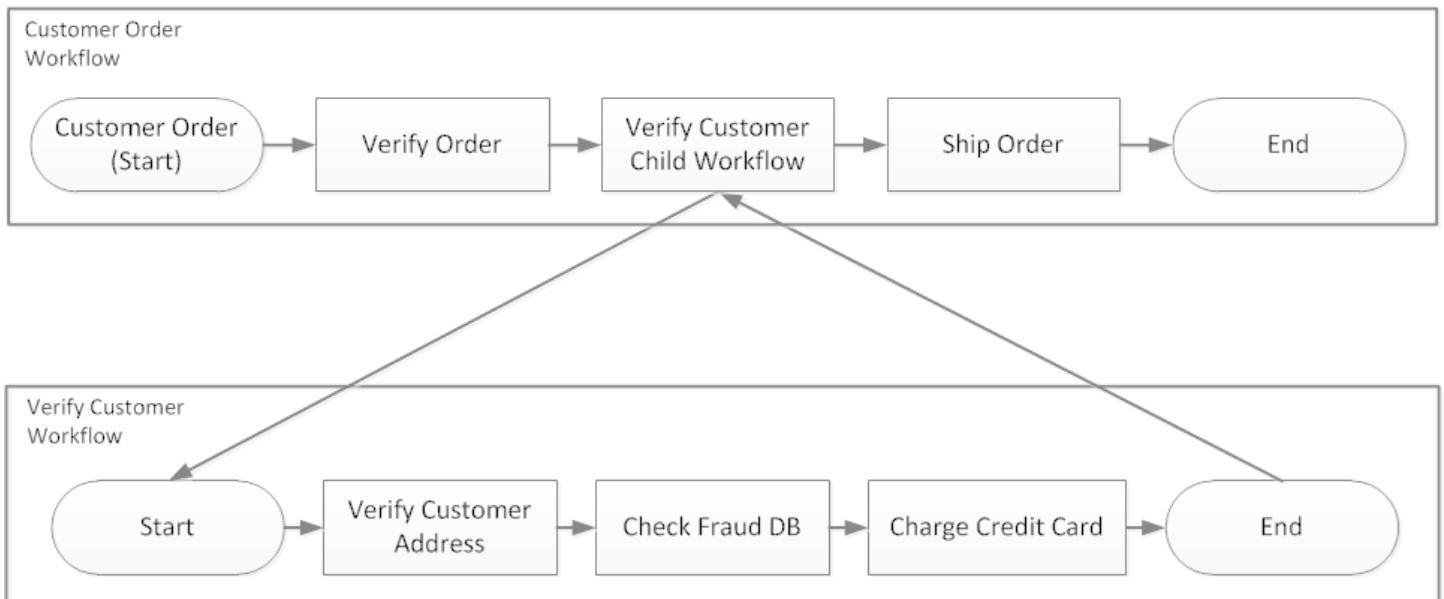
- **TERMINATE:** Amazon SWF terminará las ejecuciones secundarias.
- **REQUEST_CANCEL:** Amazon SWF intentará cancelar la ejecución secundaria situando un evento `WorkflowExecutionCancelRequested` en el historial de ejecución del flujo de trabajo de la ejecución secundaria.
- **ABANDON:** Amazon SWF no realizará ninguna acción; las ejecuciones secundarias continuarán ejecutándose.

Tras iniciarse la ejecución de flujo de trabajo secundario, se ejecuta con normalidad. Al completarse, Amazon SWF registra la finalización, junto con sus resultados, en el historial del flujo de trabajo de la ejecución de flujo de trabajo principal. Entre los ejemplos de flujos de trabajo secundarios, se incluyen los siguientes:

- Flujo de trabajo secundario de procesamiento de tarjeta de crédito usado por flujos de trabajo en diversos sitios web
- Flujo de trabajo secundario de correo electrónico que verifica la dirección de correo electrónico del cliente, comprueba la lista de cancelación, envía el correo electrónico y verifica que no rebotó ni produjo ningún error.
- Flujo de trabajo de recuperación y almacenamiento de base de datos que combina conexión, configuración, transacción y verificación.
- Flujo de trabajo secundario que combina compilación, embalaje y verificación.

En el ejemplo de E-Commerce, es posible que desee hacer de la actividad de cobro a la tarjeta de crédito un flujo de trabajo secundario. Para ello, podría registrar un nuevo flujo de trabajo de verificación del cliente, registrar la dirección de verificación del cliente y comprobar actividades de base de datos de fraude, y definir la lógica de coordinación para las tareas. A continuación, un decisor del flujo de trabajo del pedido de un cliente puede iniciar un flujo de trabajo secundario de verificación del cliente programando la decisión `StartChildWorkflowExecution` que especifica este tipo de flujo de trabajo.

En el siguiente gráfico se muestra un flujo de trabajo del pedido de un cliente que incluye un nuevo flujo de trabajo secundario de verificación del cliente, que comprueba la dirección del cliente y la base de datos de fraudes, y cobra de la tarjeta de crédito.



Varios flujos de trabajo podrían crear ejecuciones de flujo de trabajo mediante el mismo tipo de flujo de trabajo. Por ejemplo, el flujo de trabajo secundario de verificación del cliente también podría usarse en otras partes de una organización. Los eventos de un flujo de trabajo secundario se incluyen en su propio historial del flujo de trabajo y no se incluyen en el historial del flujo de trabajo del elemento principal.

Como los flujos de trabajo secundarios no son más que ejecuciones de flujo de trabajo iniciadas por un decisor, también podrían iniciarse como ejecuciones de flujos de trabajo independientes normales.

Marcadores en Amazon SWF

A veces, es posible que desee registrar información en el historial del flujo de trabajo de una ejecución de flujo de trabajo que es específica de su caso de uso. Los marcadores le permiten registrar información en el historial de ejecución del flujo de trabajo que puede usar para cualquier fin personalizado o específico de la situación.

Para utilizar marcadores, un decisor utiliza la `RecordMarker` decisión, asigna un nombre al marcador, adjunta los datos deseados a la decisión y notifica a Amazon SWF mediante la acción `RespondDecisionTaskCompleted`. Amazon SWF recibe la solicitud, registra el marcador en el historial del flujo de trabajo y aplica cualquier otra decisión de la solicitud. A partir de entonces, los decisores pueden ver el marcador en el historial del flujo de trabajo y usarlo de la forma que programe.

El registro de un marcador no inicia una tarea de decisión por sí mismo. Para evitar que la ejecución de flujo de trabajo se estanque, debe ocurrir algo que la haga continuar. Por ejemplo, esto podría incluir la programación de otra tarea de actividad por parte del decisor, la recepción de una señal por parte de la ejecución de flujo de trabajo o la finalización de una tarea de actividad programada con anterioridad.

Entre los ejemplos de marcadores, se incluyen los siguientes:

- Un contador que cuenta el número de bucles de un flujo de trabajo recursivo.
- Progreso de la ejecución de flujo de trabajo según los resultados de las actividades.
- Información resumida de eventos del historial del flujo de trabajo anteriores.

En el ejemplo de E-Commerce, es posible que desee añadir una actividad que compruebe el inventario cada día e incremente el recuento en un marcador cada vez. A continuación, podría añadir una lógica de decisión que envíe un correo electrónico al cliente o notifique a un administrador cuando el recuento supere cinco, sin tener que revisar todo el historial.

En el siguiente ejemplo, el decisor completa una tarea de decisión y responde con una acción `RespondDecisionTaskCompleted` que contiene una decisión `RecordMarker`.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions": [{
    "decisionType": "RecordMarker",
    "recordMarkerDecisionAttributes": {
      "markerName": "customer elected special shipping offer"
    }
  },
]
}
```

Si Amazon SWF registra correctamente el marcador, devolverá una respuesta HTTP correcta similar a la siguiente.

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/json
```

```
x-amzn-RequestId: 6c0373ce-074c-11e1-9083-8318c48dee96
```

Etiquetas en Amazon SWF

Amazon SWF permite etiquetar una ejecución de flujo de trabajo. Esto es útil especialmente cuando dispone de muchos recursos.

Amazon SWF admite el etiquetado de una ejecución de flujo de trabajo con hasta cinco etiquetas. Cada etiqueta es una cadena de formato libre y puede tener una longitud de hasta 256 caracteres. Si desea usar etiquetas, debe asignarlas al iniciar una ejecución de flujo de trabajo. No puede añadir etiquetas a una ejecución de flujo de trabajo una vez que se haya iniciado ni tampoco editar o quitar etiquetas que se hayan asignado a una ejecución de flujo de trabajo.

IAM permite controlar el acceso a los dominios de Amazon SWF en función de las etiquetas. Para controlar el acceso según las etiquetas, proporcione información sobre las etiquetas en el elemento de condición de una política de IAM.

Administración de etiquetas

Gestione las etiquetas de Amazon Simple Workflow Service mediante la API de Amazon SWF AWS SDKs o interactuando directamente con ella. Si utiliza la API, puede añadir etiquetas cuando registre un dominio, muestre las etiquetas de un dominio actual y las añada o elimine.

Note

Cada recurso puede tener un máximo de 50 etiquetas. Consulte [Cuotas de cuentas generales para Amazon SWF](#)

- [RegisterDomain](#)
- [ListTagsForResource](#)
- [TagResource](#)
- [UntagResource](#)

Para obtener más información, consulte [Uso de Amazon SWF APIs](#) y la [referencia de la API de Amazon Simple Email Service](#).

Etiqueta las ejecuciones del flujo de trabajo

Con Amazon SWF, puede asociar etiquetas a las ejecuciones del flujo de trabajo y, a continuación, consultar las ejecuciones del flujo de trabajo en función de estas etiquetas. Puede filtrar las listas cuando utilice las operaciones de visibilidad. Al seleccionar cuidadosamente las etiquetas que asigne a una ejecución, podrá utilizarlas para proporcionar listados significativos.

Suponga, por ejemplo, que ejecuta varios centros logísticos. Con las etiquetas, puedes enumerar los procesos que se llevan a cabo en un centro logístico específico. O bien, si un cliente está convirtiendo diferentes tipos de archivos multimedia, las etiquetas podrían indicar distintos procesos al convertir archivos de vídeo, audio e imagen.

Puede asociar hasta cinco etiquetas a una ejecución de flujo de trabajo cuando comienza la ejecución con la acción `StartWorkflowExecution`, la decisión `StartChildWorkflowExecution` o la decisión `ContinueAsNewWorkflowExecution`. Cuando utilizas acciones de visibilidad para enumerar o contar las ejecuciones del flujo de trabajo, puedes filtrar los resultados en función de tus etiquetas.

Para utilizar el etiquetado

1. Idee una estrategia de etiquetado. Piense en los requisitos de su compañía y cree una lista de etiquetas que tengan sentido para usted. Determine las ejecuciones que irán con las etiquetas. Aunque es posible asignar un máximo de cinco etiquetas a una ejecución, su biblioteca de etiquetas puede incluir el número de etiquetas que desee. Puesto que cada etiqueta puede incluir cualquier valor de cadena hasta 256 caracteres, una etiqueta puede describir casi cualquier concepto de negocio.
2. Etiquete la ejecución con un máximo de cinco etiquetas cuando la cree.
3. Para enumerar o contar las ejecuciones que están etiquetadas con una etiqueta en concreto, especifique el parámetro `tagFilter` con las acciones `ListOpenWorkflowExecutions`, `ListClosedWorkflowExecutions`, `CountOpenWorkflowExecutions` y `CountClosedWorkflowExecutions`. La acción filtrará las ejecuciones en función de las etiquetas especificadas.

Al asociar una etiqueta con una ejecución de flujo de trabajo, queda permanentemente asociada a esa ejecución y no se puede quitar.

Puede especificar una sola etiqueta en el parámetro `tagFilter` con `ListWorkflowExecutions`. Además, el establecimiento de correspondencia de etiquetas no distingue entre minúsculas y mayúsculas y solo muestra las coincidencias exactas.

Suponga que ya ha configurado dos ejecuciones con las siguientes etiquetas.

Nombre de la ejecución	Etiquetas asignadas
Ejecución uno	Consumidor, febrero de 2011
Ejecución dos	Venta al por mayor, marzo de 2011

Puede filtrar la lista de ejecuciones devueltas por `ListOpenWorkflowExecutions` en la etiqueta `Consumidor`. Los valores `oldestDate` y `latestDate` están especificados como valores de [tiempo Unix](#).

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "domain":"867530901",
  "startTimeFilter":{
    "oldestDate":1262332800,
    "latestDate":1325348400
  },
  "tagFilter":{
    "tag":"Consumer"
  }
}
```

Controle el acceso a los dominios con etiquetas

Para controlar el acceso a los dominios de Amazon Simple Workflow Service, puede hacer referencia a las etiquetas asociadas a dichos dominios de Amazon SWF en IAM.

Por ejemplo, puede restringir los dominios de Amazon SWF que incluyan una etiqueta con la clave `environment` y el valor `production` con la siguiente condición:

```
"Condition": {
  "StringEquals": {"aws:ResourceTag/environment": "production"}
```

```
}
```

Para obtener más información, consulte:

- [Control del acceso mediante etiquetas de IAM](#)
- [Políticas basadas en etiquetas](#)

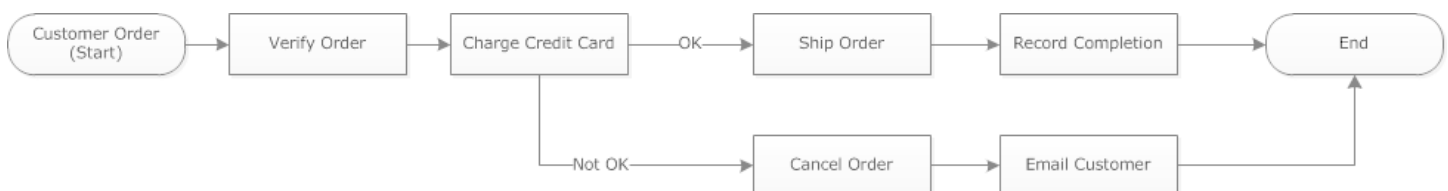
Implementación de opciones exclusivas con Amazon SWF

En algunos casos, puede ser conveniente programar un conjunto diferente de actividades en función del resultado de una actividad anterior. Con el patrón exclusivo Choice, puede crear flujos de trabajo flexibles que cumplan con los complejos requisitos de su aplicación.

Amazon SWF no tiene una acción de elección exclusiva específica. Para implementar la opción exclusiva, debe escribir su lógica de decisión para tomar decisiones en función de los resultados de una actividad anterior. Algunas aplicaciones para la opción exclusiva incluyen las siguientes acciones:

- Realización de actividades de limpieza si los resultados de una actividad anterior fueron infructuosos
- Programación de diversas actividades en función de si el cliente adquirió un plan básico o avanzado
- Realización de diversas actividades de autenticación del cliente en función del historial de pedidos del cliente

En el ejemplo de E-Commerce, podría usar la opción exclusiva para enviar o cancelar un pedido en función del resultado de carga de la tarjeta de crédito. En el siguiente gráfico, el decisor programa las tareas de actividad de envío del pedido y registro de conclusión del pedido si la tarjeta de crédito se carga correctamente. De lo contrario, programa las tareas de actividad de cancelación del pedido y envío de correo electrónico al cliente.



El decisor programa la actividad ShipOrder si la tarjeta de crédito se carga correctamente. De lo contrario, el decisor programa la actividad CancelOrder.

En este caso, programe el decisor para interpretar el historial y determinar si la tarjeta de crédito se cargó correctamente. Para ello, su lógica podría ser similar a la siguiente

```
IF lastEvent = "WorkflowExecutionStarted"
  addToDecisions ScheduleActivityTask(ActivityType = "VerifyOrderActivity")

ELSIF lastEvent = "ActivityTaskCompleted"
  AND ActivityType = "VerifyOrderActivity"
  addToDecisions ScheduleActivityTask(ActivityType = "ChargeCreditCardActivity")

#Successful Credit Card Charge Activities
ELSIF lastEvent = "ActivityTaskCompleted"
  AND ActivityType = "ChargeCreditCardActivity"
  addToDecisions ScheduleActivityTask(ActivityType = "ShipOrderActivity")

ELSIF lastEvent = "ActivityTaskCompleted"
  AND ActivityType = "ShipOrderActivity"
  addToDecisions ScheduleActivityTask(ActivityType = "RecordOrderCompletionActivity")

ELSIF lastEvent = "ActivityTaskCompleted"
  AND ActivityType = "RecordOrderCompletionActivity"
  addToDecisions CompleteWorkflowExecution

#Unsuccessful Credit Card Charge Activities
ELSIF lastEvent = "ActivityTaskFailed"
  AND ActivityType = "ChargeCreditCardActivity"
  addToDecisions ScheduleActivityTask(ActivityType = "CancelOrderActivity")

ELSIF lastEvent = "ActivityTaskCompleted"
  AND ActivityType = "CancelOrderActivity"
  addToDecisions ScheduleActivityTask(ActivityType = "EmailCustomerActivity")

ELSIF lastEvent = "ActivityTaskCompleted"
  AND ActivityType = "EmailCustomerActivity"
  addToDecisions CompleteWorkflowExecution

ENDIF
```

Si la tarjeta de crédito se cargó correctamente, el decisor debería responder con `RespondDecisionTaskCompleted` para programar la actividad `ShipOrder`.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
```

```

{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions": [
    {
      "decisionType": "ScheduleActivityTask",
      "scheduleActivityTaskDecisionAttributes": {
        "control": "OPTIONAL_DATA_FOR_DECIDER",
        "activityType": {
          "name": "ShipOrder",
          "version": "2.4"
        },
        "activityId": "3e2e6e55-e7c4-fee-deed-aa815722b7be",
        "scheduleToCloseTimeout": "3600",
        "taskList": {
          "name": "SHIPPING"
        },
        "scheduleToStartTimeout": "600",
        "startToCloseTimeout": "3600",
        "heartbeatTimeout": "300",
        "input": "123 Main Street, Anytown, United States"
      }
    }
  ]
}

```

Si la tarjeta de crédito no se cargó correctamente, el decisor debería responder con `RespondDecisionTaskCompleted` para programar la actividad `CancelOrder`.

```

https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions": [
    {
      "decisionType": "ScheduleActivityTask",
      "scheduleActivityTaskDecisionAttributes": {
        "control": "OPTIONAL_DATA_FOR_DECIDER",
        "activityType": {
          "name": "CancelOrder",
          "version": "2.4"
        },
        "activityId": "3e2e6e55-e7c4-fee-deed-aa815722b7be",
        "scheduleToCloseTimeout": "3600",

```

```
        "taskList":{
            "name":"CANCELLATIONS"
        },
        "scheduleToStartTimeout":"600",
        "startToCloseTimeout":"3600",
        "heartbeatTimeout":"300",
        "input": "Out of Stock"
    }
}
]
```

Si Amazon SWF puede validar los datos en la acción `RespondDecisionTaskCompleted`, Amazon SWF devolverá una respuesta HTTP correcta similar a la siguiente.

```
HTTP/1.1 200 OK
Content-Length: 11
Content-Type: application/json
x-amzn-RequestId: 93cec6f7-0747-11e1-b533-79b402604df1
```

Temporizadores en Amazon SWF

Con un temporizador, puede notificar a su decisor cuando haya transcurrido un período de tiempo determinado.

Al responder a una tarea de decisión, el decisor tiene la opción de responder con una decisión `StartTimer`. Esta decisión especifica una cantidad de tiempo tras la cual el temporizador debería dispararse. Una vez que haya transcurrido el tiempo especificado, Amazon SWF añadirá un evento `TimerFired` al historial de ejecución del flujo de trabajo y programará una tarea de decisión. El decisor puede entonces usar esta información para comunicar más decisiones. Una aplicación común para un temporizador es retrasar la ejecución de una tarea de actividad. Por ejemplo, es posible que un cliente desee recibir un elemento con retraso.

Cancelación de tareas de actividad en Amazon SWF

La cancelación de las tareas de actividad informa a la persona que decide finalizar las actividades que ya no es necesario realizar. Amazon SWF utiliza un mecanismo de cancelación cooperativa y no interrumpe de manera forzada las tareas de actividad en ejecución. Debe programar sus procesos de trabajo de actividad para controlar las solicitudes de cancelación.

El decisor puede decidir cancelar una tarea de actividad mientras procesa una tarea de decisión. Para cancelar una tarea de actividad, el decisor usa la acción `RespondDecisionTaskCompleted` con la decisión `RequestCancelActivityTask`.

Si un proceso de trabajo de actividad no ha adquirido aún la tarea de actividad, el servicio cancelará la tarea. Tenga en cuenta que existe una posible condición de carrera en la que un proceso de trabajo de actividad podría adquirir la tarea en cualquier momento. Si la tarea ya se ha asignado a un proceso de trabajo de actividad, se solicitará a este que cancele la tarea.

En este ejemplo, se envía una señal a la ejecución de flujo de trabajo para cancelar el pedido.

```
https://swf.us-east-1.amazonaws.com
SignalWorkflowExecution
{"domain": "867530901",
 "workflowId": "20110927-T-1",
 "runId": "9ba33198-4b18-4792-9c15-7181fb3a8852",
 "signalName": "CancelOrder",
 "input": "order 3553"}
```

Si la ejecución del flujo de trabajo recibe la señal, Amazon SWF devolverá una respuesta HTTP correcta similar a la siguiente: Amazon SWF generará una tarea de decisión para informar al decisor de que procese la señal.

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/json
x-amzn-RequestId: 6c0373ce-074c-11e1-9083-8318c48dee96
```

Cuando el decisor procese la tarea de decisión y vea la señal en el historial, intentará cancelar la actividad pendiente con el ID de actividad `ShipOrderActivity0001`. El evento de programación de tareas de actividad proporciona el ID de actividad en el historial del flujo de trabajo.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions": [{
    "decisionType": "RequestCancelActivityTask",
    "RequestCancelActivityTaskDecisionAttributes": {
      "ActivityID": "ShipOrderActivity0001"
    }
  ]
}
```

```
    }  
  }  
]  
}
```

Si Amazon SWF recibe correctamente la solicitud de cancelación, devuelve una respuesta HTTP correcta similar a la siguiente:

```
HTTP/1.1 200 OK  
Content-Length: 0  
Content-Type: application/json  
x-amzn-RequestId: 6c0373ce-074c-11e1-9083-8318c48dee96
```

El intento de cancelación se registra en el historial como el evento `ActivityTaskCancelRequested`.

Si la tarea se cancela correctamente (como lo indicaría un evento `ActivityTaskCanceled`), programe al decisor para que tome los pasos adecuados que deberían llevarse a cabo tras cancelar dicha tarea, como cerrar la ejecución del flujo de trabajo.

Si la tarea de actividad no se pudo cancelar (por ejemplo, debido a que la tarea se completó, experimentó algún error o se agotó el tiempo de espera en lugar de cancelarse), el decisor debería aceptar los resultados de la actividad o realizar cualquier limpieza o mitigación que necesite su caso de uso.

Si un proceso de trabajo de actividad ya ha adquirido la tarea de actividad, la solicitud de cancelación se transmite a través del mecanismo de latidos de tareas. Los procesos de trabajo de actividad pueden utilizar `RecordActivityTaskHeartbeat` periódicamente para informar a Amazon SWF de que la tarea sigue en curso.

Tenga en cuenta que no se requiere que los procesos de trabajo de actividad laten, aunque es recomendable para las tareas de ejecución prolongada. La cancelación de tareas requiere que se registre un latido periódico; si el proceso de trabajo no late, no se podrá cancelar la tarea.

Si el decisor solicita una cancelación de la tarea, Amazon SWF establecerá el valor del objeto `cancelRequest` en "true". El objeto `cancelRequest` forma parte del objeto `ActivityTaskStatus` devuelto por el servicio en respuesta a `RecordActivityTaskHeartbeat`.

Amazon SWF no impide la realización correcta de una tarea de actividad cuya cancelación se hubiera solicitado; dependerá de la actividad determinar cómo controlar la solicitud de cancelación.

Dependiendo de sus requisitos, programe el proceso de trabajo de actividad para cancelar la tarea de actividad o hacer caso omiso de la solicitud de cancelación.

Si desea que el proceso de trabajo de actividad indique la cancelación del trabajo para la tarea de actividad, prográmelo para que responda con `RespondActivityTaskCanceled`. Si desea que el proceso de trabajo de actividad complete la tarea, prográmelo para que responda con `RespondActivityTaskCompleted` estándar.

Cuando Amazon SWF reciba la solicitud `RespondActivityTaskCompleted` o `RespondActivityTaskCanceled`, actualizará el historial de ejecución del flujo de trabajo y programará una tarea de decisión para informar al decisor.

Programe el decisor para procesar la tarea de decisión y devolver cualquier decisión adicional. Si la tarea de actividad se cancela correctamente, programe el decisor para realizar las tareas necesarias para continuar o cerrar la ejecución de flujo de trabajo. Si la tarea de actividad no se ha cancelado correctamente, programe el decisor para aceptar los resultados, hacer caso omiso de los resultados o programar cualquier tarea de limpieza necesaria.

Seguridad en Amazon Simple Workflow Service

En esta sección se proporciona información sobre la seguridad y la autenticación en Amazon Simple Workflow Service (Amazon SWF).

Temas

- [Protección de datos en Amazon Simple Workflow Service](#)
- [Identity and Access Management en Amazon Simple Workflow Service](#)
- [Registro y supervisión](#)
- [Validación de conformidad para Amazon Simple Workflow Service](#)
- [Resiliencia en Amazon Simple Workflow Service](#)
- [Seguridad de la infraestructura en Amazon Simple Workflow Service](#)
- [Configuración y análisis de vulnerabilidades en Amazon Simple Workflow Service](#)

Amazon SWF utiliza IAM para controlar el acceso a otros AWS servicios y recursos. Para obtener información general sobre cómo funciona IAM, consulte [Información general sobre la administración del acceso](#) en la Guía del usuario de IAM. Para obtener información general de credenciales de seguridad, consulte [Credenciales de seguridad de AWS](#) en la Referencia general de Amazon Web Services.

Protección de datos en Amazon Simple Workflow Service

El [modelo de](#) se aplica a protección de datos en Amazon Simple Workflow Service. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulta las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para

cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Cómo trabajar con CloudTrail senderos](#) en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados que contienen Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta [Estándar de procesamiento de la información federal \(FIPS\) 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con Amazon SWF u otro tipo Servicios de AWS mediante la consola, la API o. AWS CLI AWS SDKs Cualquier dato que ingrese en etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Cifrado en Amazon Simple Workflow Service

Cifrado en reposo

Amazon SWF siempre cifra sus datos en reposo. Los datos de Amazon Simple Workflow Service se cifran en reposo mediante cifrado del servidor transparente. Esto ayuda a reducir la carga y la complejidad operativas que conlleva la protección de información confidencial. Con el cifrado en reposo, puede crear aplicaciones sensibles a la seguridad que cumplen los requisitos de cifrado y normativos.

Cifrado en tránsito

Todos los datos que pasan entre Amazon SWF y otros servicios se cifran mediante Seguridad de la capa de transporte (TLS).

Identity and Access Management en Amazon Simple Workflow Service

El acceso a Amazon SWF requiere credenciales que AWS pueda utilizar para autenticar sus solicitudes. Estas credenciales deben tener permisos para acceder a los AWS recursos, por ejemplo, para recuperar datos de eventos de otros recursos. AWS En las siguientes secciones, se presenta información detallada acerca de cómo puede utilizar [AWS Identity and Access Management \(IAM\)](#) y Amazon SWF para proteger sus recursos al controlar quién puede obtener acceso a ellos.

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon SWF. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración del acceso con políticas](#)
- [Control de acceso](#)
- [Acciones de política para Amazon SWF](#)
- [Recursos de políticas para Amazon SWF](#)
- [Claves de condición de políticas para Amazon SWF](#)
- [ACLs en Amazon SWF](#)
- [ABAC con Amazon SWF](#)
- [Uso de credenciales temporales con Amazon SWF](#)
- [Permisos de entidades principales entre servicios para Amazon SWF](#)
- [Roles de servicio para Amazon SWF](#)
- [Roles vinculados a servicios para Amazon SWF](#)
- [Políticas de Amazon SWF basadas en identidades](#)

- [Políticas basadas en recursos de Amazon SWF](#)
- [Cómo funciona Amazon Simple Workflow Service con IAM](#)
- [Ejemplos de políticas basadas en identidades de Amazon Simple Workflow Service](#)
- [Principios básicos](#)
- [Políticas de IAM para Amazon SWF](#)
- [Resumen de las API](#)
- [Políticas basadas en etiquetas](#)
- [Puntos de conexión de VPC de Amazon para Amazon SWF](#)
- [Solución de problemas de identidad y acceso de Amazon Simple Workflow Service](#)

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte [Solución de problemas de identidad y acceso de Amazon Simple Workflow Service](#)).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte [Cómo funciona Amazon Simple Workflow Service con IAM](#)).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte [Ejemplos de políticas basadas en identidades de Amazon Simple Workflow Service](#)).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte [Cómo iniciar sesión en la Cuenta de AWS](#) en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte [AWS Signature Version 4 para solicitudes de API](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Se recomienda encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos 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 empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte [Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte [Casos de uso para usuarios de IAM](#) en la Guía del usuario de IAM.

Roles de IAM

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Administración del acceso con políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte [Información general de políticas de JSON](#) en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de política de permisos JSON que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte [Selección entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe [especificar una entidad principal](#) en una política basada en recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte [Políticas de control de servicios](#) en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte [Políticas de control de recursos \(RCPs\)](#) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

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

Control de acceso

Aunque disponga de credenciales válidas para autenticar las solicitudes, si no tiene permisos, no podrá crear recursos de Amazon SWF ni obtener acceso a ellos. Por ejemplo, debe tener permisos para AWS Lambda invocar los destinos de Amazon Simple Notification Service (Amazon SNS) y Amazon Simple Queue Service (Amazon SQS) asociados a sus reglas de Amazon SWF.

En las secciones siguientes, se describe cómo administrar los permisos de Amazon SWF. Recomendamos que lea primero la información general.

- [Principios básicos](#)
- [Políticas de IAM para Amazon SWF](#)
- [Redacción de políticas para Amazon SWF](#)

Acciones de política para Amazon SWF

Compatibilidad con las acciones de políticas: sí

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

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Amazon SWF, consulte [Recursos definidos por Amazon Simple Workflow Service](#) en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Amazon SWF utilizan el siguiente prefijo antes de la acción:

```
swf
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [  
  "swf:action1",  
  "swf:action2"  
]
```

Para ver ejemplos de políticas basadas en identidad de Amazon SWF, consulte [Ejemplos de políticas basadas en identidades de Amazon Simple Workflow Service](#).

Recursos de políticas para Amazon SWF

Compatibilidad con los recursos de políticas: sí

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

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). En el caso de las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de recursos de Amazon SWF y sus tipos ARNs, consulte [Acciones definidas por Amazon Simple Workflow Service](#) en la Referencia de autorización del servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte [Recursos definidos por Amazon Simple Workflow Service](#).

Para ver ejemplos de políticas basadas en identidad de Amazon SWF, consulte [Ejemplos de políticas basadas en identidades de Amazon Simple Workflow Service](#).

Claves de condición de políticas para Amazon SWF

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

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

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del usuario de IAM.

Para obtener una lista de las claves de condición de Amazon SWF, consulte [Claves de condición de Amazon Simple Workflow Service](#) en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte [Recursos definidos por Amazon Simple Workflow Service](#).

Para ver ejemplos de políticas basadas en identidad de Amazon SWF, consulte [Ejemplos de políticas basadas en identidades de Amazon Simple Workflow Service](#).

ACLs en Amazon SWF

Soporta ACLs: No

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

ABAC con Amazon SWF

Compatibilidad con ABAC (etiquetas en las políticas): parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos denominados etiquetas. Puede adjuntar etiquetas a las entidades y AWS los recursos de IAM y, a continuación, diseñar políticas de ABAC para permitir las operaciones cuando la etiqueta del director coincida con la etiqueta del recurso.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte [Definición de permisos con la autorización de ABAC](#) en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte [Uso del control de acceso basado en atributos \(ABAC\)](#) en la Guía del usuario de IAM.

Uso de credenciales temporales con Amazon SWF

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a AWS los recursos a corto plazo y se crean automáticamente cuando se utiliza la federación o se cambia de rol. 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](#) y [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Permisos de entidades principales entre servicios para Amazon SWF

Admite sesiones de acceso directo (FAS): sí

Las sesiones de acceso directo (FAS) utilizan los permisos del principal que llama y los que solicitan Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Para

obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Sesiones de acceso directo](#).

Roles de servicio para Amazon SWF

Compatible con roles de servicio: sí

Un rol de servicio es un [rol de IAM](#) que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Crear un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

Warning

Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de Amazon SWF. Edite los roles de servicio solo cuando Amazon SWF proporcione orientación para hacerlo.

Roles vinculados a servicios para Amazon SWF

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta [Servicios de AWS que funcionan con IAM](#). Busque un servicio en la tabla que incluya Yes en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Políticas de Amazon SWF basadas en identidades

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué

condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de la política de JSON de IAM](#) en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades para Amazon SWF

Para ver ejemplos de políticas basadas en identidad de Amazon SWF, consulte [Ejemplos de políticas basadas en identidades de Amazon Simple Workflow Service](#).

Políticas basadas en recursos de Amazon SWF

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política basada en recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Cómo funciona Amazon Simple Workflow Service con IAM

Antes de utilizar IAM para administrar el acceso a Amazon SWF, obtenga información sobre qué características de IAM se encuentran disponibles en Amazon SWF.

Características de IAM que puede utilizar con Amazon Simple Workflow Service

Característica de IAM	Compatibilidad de Amazon SWF
Políticas basadas en identidades	Sí

Característica de IAM	Compatibilidad de Amazon SWF
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACLs	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan Amazon SWF y otros AWS servicios con la mayoría de las funciones de IAM, consulte los [AWS servicios que funcionan con IAM en la Guía del usuario de IAM](#).

Ejemplos de políticas basadas en identidades de Amazon Simple Workflow Service

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon SWF. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte [Creación de políticas de IAM \(consola\)](#) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por Amazon SWF, incluido el formato de cada uno de los tipos de recursos, consulte [Acciones, recursos y claves de condición de Amazon Simple Workflow Service](#) en la Referencia de autorización del servicio. ARNs

Temas

- [Prácticas recomendadas sobre las políticas](#)
- [Uso de la consola de Amazon SWF](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon SWF de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de tarea](#) en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.

- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte [Validación de políticas con el Analizador de acceso de IAM](#) en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte [Acceso seguro a la API con MFA](#) en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Uso de la consola de Amazon SWF

Para acceder a la consola de Amazon Simple Workflow Service, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de Amazon SWF que tiene en 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 necesita conceder permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de Amazon SWF, adjunte también la política gestionada *ReadOnly* AWS o Amazon *ConsoleAccess* SWF a las entidades. Para obtener más información, consulte [Adición de permisos a un usuario](#) en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. AWS CLI AWS

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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": "*"
    }
  ]
}

```

Principios básicos

El control de acceso de Amazon SWF se basa principalmente en dos tipos de permisos:

- Permisos en el nivel de recursos: los recursos de Amazon SWF a los que puede obtener acceso un usuario.

Solo puede expresar los permisos a nivel de recursos para los dominios.

- Permisos en el nivel de API: las acciones de Amazon SWF a las que puede llamar un usuario.

El enfoque más sencillo consiste en conceder acceso total a la cuenta (llamar a cualquier acción de Amazon SWF en cualquier dominio) o denegar el acceso por completo. Sin embargo, IAM admite un método de control de acceso más preciso que suele ser más útil. Por ejemplo, puede:

- Permitir que el usuario llame a cualquier acción de Amazon SWF sin restricciones, pero solo en un dominio determinado. Puede utilizar una política de este tipo para permitir que aplicaciones de flujo de trabajo que están en desarrollo utilicen cualquier acción, pero solo un dominio de "entorno de prueba".
- Permitir que el usuario acceda a cualquier dominio, pero limitar la forma en que usa la API. Puede utilizar una política de este tipo para permitir que una aplicación de "auditoría" llame a la API en cualquier dominio, pero acceso de solo lectura.
- Permitir que el usuario llame solo a un conjunto limitado de acciones en ciertos dominios. Puede utilizar una política de este tipo para permitir que un iniciador de flujo de trabajo solo llame a la acción `StartWorkflowExecution` en un determinado dominio.

El control de acceso de Amazon SWF se basa en los siguientes principios:

- Las decisiones de control de acceso se basan solo en políticas de IAM; todas las operaciones de manipulación y auditoría de la política se realizan por medio de IAM.
- El modelo de control de acceso utiliza una deny-by-default política; se deniega cualquier acceso que no esté permitido de forma explícita.
- Para controlar el acceso a los recursos de Amazon SWF, asocie políticas de IAM adecuadas a los actores del flujo de trabajo.
- Solo se pueden expresar permisos a nivel de recursos para los dominios.
- Puede limitar aún más el uso de ciertas acciones aplicando condiciones a uno o más parámetros.
- Si concedes permiso de uso [RespondDecisionTaskCompleted](#), puedes expresar los permisos para la lista de decisiones incluidas en esa acción.

Cada decisión tiene uno o más parámetros, como una llamada a la API normal. Para permitir que las políticas sean lo más fáciles de leer posible, puede expresar permisos a nivel de decisiones como si fueran llamadas a la API reales, lo que incluye la aplicación de condiciones a ciertos parámetros. Estos tipos de permisos se llaman permisos pseudo API.

Para obtener un resumen de qué parámetros de la API normal y la pseudo API se pueden limitar con estas condiciones, consulte [Resumen de las API](#).

Políticas de IAM para Amazon SWF

Una política de IAM comprende uno o varios elementos `Statement`, cada uno de los cuales consta de un conjunto de elementos que define la política. Para obtener una lista completa de los elementos y un análisis general acerca de cómo crear políticas, consulte [El lenguaje de la política de acceso](#). El control de acceso a Amazon SWF se basa en los siguientes elementos:

Efecto

(Obligatorio) El efecto de la instrucción: `deny` o `allow`.

Note

Debe permitir el acceso de forma explícita. IAM deniega el acceso de forma predeterminada.

Recurso

(Obligatorio) El recurso (una entidad de un AWS servicio con la que el usuario puede interactuar) al que se aplica la declaración.

Solo puede expresar los permisos a nivel de recursos para los dominios. Por ejemplo, una política puede permitir acceder solo a determinados dominios en su cuenta. Para expresar los permisos de un dominio, `Resource` configúrelo en el nombre de recurso de Amazon (ARN) del dominio, que tiene el formato «`arn:aws:swf:: /domain/`». *Region AccountID DomainName* *Regiones* la AWS región, *AccountID* es el ID de la cuenta sin guiones y es el nombre de dominio. *DomainName*

Action

(Obligatorio) La acción a la que se aplica la declaración, a la que puede hacer referencia utilizando el siguiente formato: *serviceId:action*. Para Amazon SWF, *serviceID* establézcalo en. `swf` Por ejemplo, `swf:StartWorkflowExecution` hace referencia a la [StartWorkflowExecution](#) acción y se utiliza para controlar qué usuarios pueden iniciar flujos de trabajo.

Si concedes permiso de uso [RespondDecisionTaskCompleted](#), también puedes controlar el acceso a la lista de decisiones incluida `Action` para expresar los permisos de la pseudoAPI. Dado que IAM deniega el acceso de forma predeterminada, la decisión del decisor debe indicarse

explícitamente o, de lo contrario, no se aceptará. Puede utilizar un valor * para permitir todas las decisiones.

Condición

(Opcional) Expresa una restricción en uno o más parámetros de una acción, que limitan los valores permitidos.

Las acciones de Amazon SWF tienen a menudo un gran alcance, que se puede reducir con la ayuda de condiciones de IAM. Por ejemplo, para limitar las listas de tareas a las que puede acceder la [PollForActivityTask](#) acción, debes incluir una `Condition` y utilizar la `swf:taskList.name` clave para especificar las listas permitidas.

Puede expresar restricciones para las siguientes entidades:

- El tipo de flujo de trabajo. El nombre y la versión tienen claves independientes.
- El tipo de actividad. El nombre y la versión tienen claves independientes.
- Las listas de tareas.
- Tags. Puede especificar varias etiquetas para algunas acciones. En ese caso, cada etiqueta tiene una clave distinta.

Note

Para Amazon SWF, todos los valores son cadenas, de modo que puede limitar un parámetro por medio de un operador de cadena como `StringEquals`, que limita el parámetro a una cadena especificada. Sin embargo, los operadores de comparación de la cadena regulares como `StringEquals` requieren que todas las solicitudes incluyan el parámetro. Si no incluye el parámetro de forma explícita, y no hay un valor predeterminado como la lista de tareas predeterminada provista durante el registro del tipo, se denegará el acceso.

A menudo es útil tratar las condiciones como si fueran opcionales. De este modo, puede llamar a una acción sin incluir necesariamente el parámetro asociado. Por ejemplo, es posible que desee permitir que un decisor especifique un conjunto de [RespondDecisionTaskCompleted](#) decisiones, pero también permitirle especificar solo una de ellas para una llamada determinada. En ese caso, puede limitar los parámetros pertinentes con un operador `StringEqualsIfExists`, a fin de permitir el acceso si el parámetro satisface la condición, pero sin denegar el acceso si el parámetro está ausente.

Para obtener una lista completa de los parámetros que puede limitar y las claves asociadas, consulte [Resumen de las API](#).

La siguiente sección proporciona ejemplos de cómo crear políticas de Amazon SWF. Para obtener más información, consulte [Condiciones de la cadena](#).

Redacción de políticas para Amazon SWF

Un flujo de trabajo consta de varios actores: actividades, decisores, etc. Para controlar el acceso de cada actor, asocie una política de IAM adecuada.

Con la siguiente acción, el actor tendrá acceso completo a la cuenta en todas las regiones:

- Acción: `swf:*`
- Recurso: `arn:aws:swf:*:123456789012:/domain/*`

Puede utilizar comodines para que un solo valor represente varios recursos, acciones o regiones.

- El primer comodín (*) del `Resource` valor indica que los permisos de recursos se aplican a todas las regiones.

Para limitar los permisos a una sola región, sustituya el comodín por la cadena de región apropiada, como `us-east-1`.

- El segundo comodín (*) en el valor `Resource` permite al actor acceder a todos los dominios en las regiones especificadas.
- El comodín (*) en el valor `Action` permite al actor llamar a cualquier acción de Amazon SWF.

Para obtener más información sobre el uso de los comodines, consulte [Descripciones de elementos](#)

Permisos de dominio

Para restringir los flujos de trabajo de un departamento a un dominio concreto, puedes conceder un permiso que permita a un actor realizar cualquier acción, pero solo para un departamento específico.

Para conceder a un actor acceso a más de un dominio, expresa el permiso para cada dominio en forma de lista de declaraciones:

- Acción: `swf:*`
- Recurso: `arn:aws:swf:*:123456789012:/domain/department1`

- Recurso: `arn:aws:swf:*:123456789012:/domain/department2`

Puede permitir que un actor utilice cualquier acción de Amazon SWF en los dominios `department1` y `department2`. También puede utilizar comodines en ocasiones para representar varios dominios.

Restricciones y permisos de la API

Usted controla qué acciones puede utilizar un actor especificando la acción en el `Action` elemento.

Con la siguiente acción, un actor solo puede llamar `StartWorkflowExecution` para iniciar flujos de trabajo. No puede usar ninguna otra acción.

- Acción: `swf:StartWorkflowExecution`

Condiciones

Si lo desea, puede restringir los valores de los parámetros permitidos de la acción mediante un `Condition` elemento.

Para restringir los flujos de trabajo que puede iniciar un actor, restrinja uno o más de los valores de los `StartWorkflowExecution` parámetros, de la siguiente manera:

```
"Condition" : {
  "StringEquals" : {
    "swf:workflowType.name" : "workflow1",
    "swf:workflowType.version" : "version2"
  }
}
```

Un actor con las restricciones anteriores solo `version2` puede ejecutar `workflow1` y ambos parámetros deben incluirse en la solicitud.

Para limitar un parámetro sin tener que incluirlo en una solicitud, utilice un operador `StringEqualsIfExists`, como sigue:

```
"Condition" : {
  "StringEqualsIfExists" : { "swf:taskList.name" : "task_list_name" }
}
```

Un actor con la política anterior puede especificar opcionalmente una lista de tareas al iniciar la ejecución de un flujo de trabajo.

Puede limitar la lista de etiquetas para algunas acciones. Cada etiqueta tiene una clave independiente, por lo que puede `swf:tagList.member.0` restringir la primera etiqueta de la lista, `swf:tagList.member.1` restringir la segunda etiqueta de la lista, etc., hasta un máximo de 5.

Debe tener cuidado al restringir las listas de etiquetas. Por ejemplo, no se recomienda la siguiente condición.

No se recomienda la siguiente condición porque permite especificar opcionalmente una `some_ok_tag` u `otraanother_ok_tag`. Sin embargo, la condición restringe solo el primer elemento de la lista de etiquetas. La lista podría tener elementos adicionales con valores arbitrarios, todos los cuales estarían permitidos porque la condición no aplica ninguna condición `swf:tagList.member.1swf:tagList.member.2`, etc.

```
// Example to illustrate an insecure Condition
"Condition" : {
  "StringEqualsIfExists" : {
    "swf:tagList.member.0" : "some_ok_tag", "another_ok_tag"
  }
}
```

Una forma de solucionar el problema anterior es impedir el uso de listas de etiquetas.

La siguiente política asegura que solo se permitan `some_ok_tag` o `another_ok_tag` al exigir que la lista tenga solo un elemento.

```
"Condition" : {
  "StringEqualsIfExists" : {
    "swf:tagList.member.0" : "some_ok_tag", "another_ok_tag"
  },
  "Null" : { "swf:tagList.member.1" : "true" }
}
```

Restricciones y permisos de pseudo API

Para restringir las decisiones disponibles `RespondDecisionTaskCompleted`, primero debes permitir que el actor llame `RespondDecisionTaskCompleted`. A continuación, debes expresar los permisos para los miembros de la pseudoAPI correspondientes utilizando la misma sintaxis que para la API normal, de la siguiente manera:

- Declaración 1

Recurso: `arn:aws:swf:*:123456789012:/domain/*`

Acción: `swf:RespondDecisionTaskCompleted`

- Declaración 2

Recurso: `*`

Acción: `swf:ScheduleActivityTask`

Condición: `"StringEquals" : { "swf:activityType.name" :
"SomeActivityType" }`

La primera Statement permite al actor llamar `RespondDecisionTaskCompleted`. La segunda afirmación permite al actor utilizar la `ScheduleActivityTask` decisión de ordenar a Amazon SWF que programe una tarea de actividad. Para permitir todas las decisiones, sustituya «`swf:ScheduleActivityTask`» por «`swf:*`».

Puede utilizar operadores de condición para limitar los parámetros al igual que con la API normal. El `StringEquals` operador del ejemplo anterior `Condition` permite `RespondDecisionTaskCompleted` programar una tarea de actividad para la `SomeActivityType` actividad y debe programarla. Si quiere permitir que `RespondDecisionTaskCompleted` utilice un valor de parámetro pero sin que sea obligatorio, utilice el operador `StringEqualsIfExists`.

AWS política gestionada: `SimpleWorkflowFullAccess`

Puede asociar la política `SimpleWorkflowFullAccess` a las identidades de IAM.

Esta política proporciona acceso completo al servicio de configuración de Amazon SWF.

Limitaciones del modelo de servicio de las políticas de IAM

Debe tener presentes las restricciones del modelo de servicio al crear políticas de IAM. Es posible crear una política de IAM válida desde el punto de vista sintáctico que represente una solicitud de Amazon SWF no válida; una solicitud permitida en términos de control de acceso puede producir un error, ya que se trata de una solicitud no válida.

Por ejemplo, el modelo de servicio de Amazon SWF no permite utilizar `tagFilter` los parámetros `typeFilter` and en la misma [ListOpenWorkflowExecutions](#) solicitud. La siguiente condición

permitiría que las llamadas que el servicio rechazara (mediante el lanzamiento) por considerarlas solicitudes no válidas `ValidationException`:

```
"Condition" : {
  "StringEquals" : {
    "swf:typeFilter.name" : "workflow_name",
    "swf:typeFilter.version" : "workflow_version",
    "swf:tagFilter.tag" : "some_tag"
  }
}
```

Resumen de las API

En esta sección, se describe brevemente cómo utilizar las políticas de IAM para controlar la forma en que un actor puede usar cada API y pseudo API para obtener acceso a los recursos de Amazon SWF.

- Para todas las acciones excepto `RegisterDomain` y `ListDomains`, puede expresar permisos para el recurso del dominio a fin de autorizar o denegar el acceso a todos o parte de los dominios de la cuenta.
- Puede permitir o denegar el permiso a cualquier miembro de la API normal y, si concede permiso para llamar a [RespondDecisionTaskCompleted](#), para cualquier miembro de la pseudo API.
- Puede utilizar una condición para restringir los valores permitidos de algunos parámetros.

Las siguientes secciones indican los parámetros que es posible limitar para cada miembro de la API normal y la pseudo API; proporcionan la clave asociada y señalan las limitaciones sobre cómo puede controlar el acceso al dominio.


API normal

Esta sección enumera los miembros de la API normal, y describe brevemente los parámetros que se pueden limitar y las claves asociadas. También señala las limitaciones sobre cómo puede controlar el acceso al dominio.

[CountClosedWorkflowExecutions](#)

- `tagFilter.tag`: restricción de cadena. La clave es `swf:tagFilter.tag`
- `typeFilter.name`: restricción de cadena. La clave es `swf:typeFilter.name`.


- `typeFilter.version`: restricción de cadena. La clave es `swf:typeFilter.version`.

 Note

`CountClosedWorkflowExecutions` exige que `typeFilter` y `tagFilter` sean mutuamente exclusivos.

[CountOpenWorkflowExecutions](#)

- `tagFilter.tag`: restricción de cadena. La clave es `swf:tagFilter.tag`
- `typeFilter.name`: restricción de cadena. La clave es `swf:typeFilter.name`.
- `typeFilter.version`: restricción de cadena. La clave es `swf:typeFilter.version`.

 Note

`CountOpenWorkflowExecutions` exige que `typeFilter` y `tagFilter` sean mutuamente exclusivos.

[CountPendingActivityTasks](#)

- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.

[CountPendingDecisionTasks](#)

- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.

[DeleteActivityType](#)

- `activityType.name`: restricción de cadena. La clave es `swf:activityType.name`.
- `activityType.version`: restricción de cadena. La clave es `swf:activityType.version`.

[DeprecateActivityType](#)

- `activityType.name`: restricción de cadena. La clave es `swf:activityType.name`.

- `activityType.version`: restricción de cadena. La clave es `swf:activityType.version`.

[DeprecateDomain](#)

- No puede limitar los parámetros de esta acción.

[DeleteWorkflowType](#)

- `workflowType.name`: restricción de cadena. La clave es `swf:workflowType.name`.
- `workflowType.version`: restricción de cadena. La clave es `swf:workflowType.version`.

[DeprecateWorkflowType](#)

- `workflowType.name`: restricción de cadena. La clave es `swf:workflowType.name`.
- `workflowType.version`: restricción de cadena. La clave es `swf:workflowType.version`.

[DescribeActivityType](#)

- `activityType.name`: restricción de cadena. La clave es `swf:activityType.name`.
- `activityType.version`: restricción de cadena. La clave es `swf:activityType.version`.

[DescribeDomain](#)

- No puede limitar los parámetros de esta acción.

[DescribeWorkflowExecution](#)

- No puede limitar los parámetros de esta acción.

[DescribeWorkflowType](#)

- `workflowType.name`: restricción de cadena. La clave es `swf:workflowType.name`.
- `workflowType.version`: restricción de cadena. La clave es `swf:workflowType.version`.

[GetWorkflowExecutionHistory](#)

- No puede limitar los parámetros de esta acción.

[ListActivityTypes](#)

- No puede limitar los parámetros de esta acción.

[ListClosedWorkflowExecutions](#)

- `tagFilter.tag`: restricción de cadena. La clave es `swf:tagFilter.tag`
- `typeFilter.name`: restricción de cadena. La clave es `swf:typeFilter.name`.
- `typeFilter.version`: restricción de cadena. La clave es `swf:typeFilter.version`.

Note

`ListClosedWorkflowExecutions` exige que `typeFilter` y `tagFilter` sean mutuamente exclusivos.

[ListDomains](#)

- No puede limitar los parámetros de esta acción.

[ListOpenWorkflowExecutions](#)

- `tagFilter.tag`: restricción de cadena. La clave es `swf:tagFilter.tag`
- `typeFilter.name`: restricción de cadena. La clave es `swf:typeFilter.name`.
- `typeFilter.version`: restricción de cadena. La clave es `swf:typeFilter.version`.

Note

`ListOpenWorkflowExecutions` exige que `typeFilter` y `tagFilter` sean mutuamente exclusivos.

[ListWorkflowTypes](#)

- No puede limitar los parámetros de esta acción.

[PollForActivityTask](#)

- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.

[PollForDecisionTask](#)

- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.

[RecordActivityTaskHeartbeat](#)

- No puede limitar los parámetros de esta acción.

[RegisterActivityType](#)

- `defaultTaskList.name`: restricción de cadena. La clave es `swf:defaultTaskList.name`.
- `name`: restricción de cadena. La clave es `swf:name`.
- `version`: restricción de cadena. La clave es `swf:version`.

[RegisterDomain](#)

- `name`: el nombre del dominio que se está registrando está disponible como recurso de esta acción.

[RegisterWorkflowType](#)

- `defaultTaskList.name`: restricción de cadena. La clave es `swf:defaultTaskList.name`.
- `name`: restricción de cadena. La clave es `swf:name`.
- `version`: restricción de cadena. La clave es `swf:version`.

[RequestCancelWorkflowExecution](#)

- No puede limitar los parámetros de esta acción.

[RespondActivityTaskCanceled](#)

- No puede limitar los parámetros de esta acción.

[RespondActivityTaskCompleted](#)

- No puede limitar los parámetros de esta acción.

[RespondActivityTaskFailed](#)

- No puede limitar los parámetros de esta acción.

[RespondDecisionTaskCompleted](#)

- `decisions.member.N`: restringido indirectamente mediante pseudopermisos de API. Para obtener más información, consulte [Pseudo API](#).

[SignalWorkflowExecution](#)

- No puede limitar los parámetros de esta acción.

[StartWorkflowExecution](#)

- `tagList.member.0`: restricción de cadena. La clave es `swf:tagList.member.0`
- `tagList.member.1`: restricción de cadena. La clave es `swf:tagList.member.1`
- `tagList.member.2`: restricción de cadena. La clave es `swf:tagList.member.2`
- `tagList.member.3`: restricción de cadena. La clave es `swf:tagList.member.3`
- `tagList.member.4`: restricción de cadena. La clave es `swf:tagList.member.4`
- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.
- `workflowType.name`: restricción de cadena. La clave es `swf:workflowType.name`.
- `workflowType.version`: restricción de cadena. La clave es `swf:workflowType.version`.

Note

No puede limitar más de cinco etiquetas.

TerminateWorkflowExecution

- No puede limitar los parámetros de esta acción.

Pseudo API

Esta sección indica los miembros de la pseudo API, que representan las decisiones incluidas en [RespondDecisionTaskCompleted](#). Si otorga permiso para utilizar `RespondDecisionTaskCompleted`, su política puede expresar permisos para los miembros de esta API de la misma forma que para la API normal. Para limitar aún más algunos miembros de la pseudo API, puede establecer condiciones en uno o más parámetros. Esta sección enumera los miembros de la pseudo API, y describe brevemente los parámetros que se pueden limitar y las claves asociadas.

Note

Las claves `aws:SourceIP`, `aws:UserAgent` y `aws:SecureTransport` no están disponibles para la pseudo API. Si la política de seguridad prevista exige que estas claves controlen el acceso a la pseudo API, puede utilizarlas con la acción `RespondDecisionTaskCompleted`.

CancelTimer

- No puede limitar los parámetros de esta acción.

CancelWorkflowExecution

- No puede limitar los parámetros de esta acción.


CompleteWorkflowExecution

- No puede limitar los parámetros de esta acción.

ContinueAsNewWorkflowExecution

- `tagList.member.0`: restricción de cadena. La clave es `swf:tagList.member.0`

- `tagList.member.1`: restricción de cadena. La clave es `swf:tagList.member.1`
- `tagList.member.2`: restricción de cadena. La clave es `swf:tagList.member.2`
- `tagList.member.3`: restricción de cadena. La clave es `swf:tagList.member.3`
- `tagList.member.4`: restricción de cadena. La clave es `swf:tagList.member.4`
- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.
- `workflowTypeVersion`: restricción de cadena. La clave es `swf:workflowTypeVersion`.

 Note

No puede limitar más de cinco etiquetas.

FailWorkflowExecution

- No puede limitar los parámetros de esta acción.

RecordMarker

- No puede limitar los parámetros de esta acción.

RequestCancelActivityTask

- No puede limitar los parámetros de esta acción.

RequestCancelExternalWorkflowExecution

- No puede limitar los parámetros de esta acción.

ScheduleActivityTask

- `activityType.name`: restricción de cadena. La clave es `swf:activityType.name`.
- `activityType.version`: restricción de cadena. La clave es `swf:activityType.version`.
- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.

SignalExternalWorkflowExecution

- No puede limitar los parámetros de esta acción.

StartChildWorkflowExecution

- `tagList.member.0`: restricción de cadena. La clave es `swf:tagList.member.0`
- `tagList.member.1`: restricción de cadena. La clave es `swf:tagList.member.1`
- `tagList.member.2`: restricción de cadena. La clave es `swf:tagList.member.2`
- `tagList.member.3`: restricción de cadena. La clave es `swf:tagList.member.3`
- `tagList.member.4`: restricción de cadena. La clave es `swf:tagList.member.4`
- `taskList.name`: restricción de cadena. La clave es `swf:taskList.name`.
- `workflowType.name`: restricción de cadena. La clave es `swf:workflowType.name`.
- `workflowType.version`: restricción de cadena. La clave es `swf:workflowType.version`.

Note

No puede limitar más de cinco etiquetas.

StartTimer

- No puede limitar los parámetros de esta acción.

Políticas basadas en etiquetas

Amazon SWF admite políticas basadas en etiquetas. Por ejemplo, puede restringir los dominios de Amazon SWF que incluyan una etiqueta con la clave `environment` y el valor `production` con la siguiente condición:

```
"Condition": {
  "StringEquals": {"aws:ResourceTag/environment": "production"}
}
```

Para obtener más información acerca del etiquetado, consulte:

- [Etiquetas en Amazon SWF](#)
- [Control del acceso mediante etiquetas de IAM](#)

Puntos de conexión de VPC de Amazon para Amazon SWF

Note

AWS PrivateLink Actualmente, el soporte solo está disponible en las regiones AWS Top Secret: Este, AWS Secret Region y China.

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus AWS recursos, puede establecer una conexión entre sus flujos de trabajo de Amazon VPC y Amazon Simple Workflow Service. Puede utilizar esta conexión con sus flujos de trabajo de Amazon SWF sin cruzar la red de internet pública.

Amazon VPC le permite lanzar AWS recursos en una red virtual personalizada. Puede utilizar una VPC para controlar la configuración de red, como el intervalo de direcciones IP, las subredes, las tablas de enrutamiento y las puertas de enlace de red. Para obtener más información VPCs, consulte la Guía del [usuario de Amazon VPC](#).

Para conectar Amazon VPC a Amazon SWF, primero debe definir un punto de conexión de VPC de interfaz, lo que le permitirá conectar la VPC a otros Servicios de AWS. El punto de conexión ofrece conectividad escalable de confianza sin necesidad de utilizar una gateway de Internet, una instancia de conversión de las direcciones de red (NAT) o una conexión de VPN. Para obtener más información, consulte [Puntos de conexión de VPC de la interfaz \(AWS PrivateLink\)](#) en la Guía del usuario de Amazon VPC.

Creación del punto de conexión

Puede crear un punto de conexión de Amazon SWF en su VPC mediante Consola de administración de AWS, the AWS Command Line Interface (AWS CLI), un AWS SDK, la API de Amazon SWF o CloudFormation

Para obtener información sobre la creación y configuración de un punto de conexión mediante la consola de Amazon VPC o la AWS CLI, consulte [Creación de un punto de conexión de interfaz](#) en la Guía del usuario de Amazon VPC.

Note

Al crear un punto de conexión, especifique que Amazon SWF es el servicio al que desea conectar la VPC. En la consola de Amazon VPC, los nombres de los servicios varían en

función de la región de AWS . Por ejemplo, en la región AWS Top Secret (Este), el nombre del servicio de Amazon SWF es `com.amazonaws.us-iso-east-1.swf`.

Para obtener información sobre cómo crear y configurar un punto final mediante CloudFormation, consulte el VPC Endpoint recurso [AWS:EC2:::](#) en la Guía del CloudFormation usuario.

Políticas de punto de conexión de VPC de Amazon

Para controlar el acceso de conectividad a Amazon SWF, puede adjuntar una política de punto final AWS Identity and Access Management (IAM) mientras crea un punto de enlace de Amazon VPC. Puede crear reglas de IAM complejas al asociar varias políticas de punto de conexión. Para obtener más información, consulte:

- [Políticas de puntos de conexión de Amazon Virtual Private Cloud para Amazon SWF](#)
- [Control del acceso a los servicios con puntos de conexión de VPC](#)

Políticas de puntos de conexión de Amazon Virtual Private Cloud para Amazon SWF

Puede crear una política para los puntos de conexión de VPC de Amazon para Amazon SWF donde especifique lo siguiente:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

En el siguiente ejemplo, se añade una función de IAM específica a una política:

```
"Principal": {  
  "AWS": "arn:aws:iam::123456789012:role/MyRole"  
}
```

- Para obtener más información sobre cómo crear políticas de puntos de conexión, consulte [Controlar el acceso a servicios con puntos de conexión de VPC](#).
- Para obtener información sobre cómo puede utilizar IAM para controlar el acceso a sus recursos AWS y a los de Amazon SWF, consulte [Identity and Access Management en Amazon Simple Workflow Service](#)

Solución de problemas de identidad y acceso de Amazon Simple Workflow Service

Utilice la siguiente información para diagnosticar y solucionar los problemas habituales que puedan surgir al trabajar con Amazon SWF e IAM.

Temas

- [No tengo autorización para realizar una acción en Amazon SWF](#)
- [No estoy autorizado a realizar tareas como: PassRole](#)
- [Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon SWF](#)

No tengo autorización para realizar una acción en Amazon SWF

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio `my-example-widget`, pero no tiene los permisos ficticios `swf:GetWidget`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
swf:GetWidget on resource: my-example-widget
```

En este caso, la política de Mateo se debe actualizar para permitirle acceder al recurso `my-example-widget` mediante la acción `swf:GetWidget`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para llevar a cabo la acción `iam:PassRole`, las políticas se deben actualizar para permitirle pasar un rol a Amazon SWF.

Algunos 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 la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon SWF. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon SWF

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se 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), puede utilizar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si Amazon SWF admite estas características, consulte [Cómo funciona Amazon Simple Workflow Service con IAM](#).
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte [Proporcionar acceso a un usuario de IAM en otro de su propiedad en la Cuenta de AWS Guía del usuario](#) de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta [Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Registro y supervisión

En esta sección, se proporciona información sobre el registro y la monitorización en Amazon SWF.

Temas

- [Métricas de Amazon SWF para CloudWatch](#)
- [Visualización de las métricas de Amazon SWF para usar el CloudWatch Consola de administración de AWS](#)
- [Grabación de llamadas a la API con AWS CloudTrail](#)
- [EventBridge para cambios en el estado de ejecución de Amazon SWF](#)
- [Uso AWS User Notifications con Amazon Simple Workflow Service](#)

Métricas de Amazon SWF para CloudWatch

Amazon SWF ahora proporciona métricas CloudWatch que puede utilizar para realizar un seguimiento de sus flujos de trabajo y actividades y configurar alarmas en los valores límite que elija. Puede ver las métricas mediante. Consola de administración de AWS Para obtener más información, consulte [Visualización de las métricas de Amazon SWF para usar el CloudWatch Consola de administración de AWS](#).

Temas

- [Unidades de informes para métricas de Amazon SWF](#)
- [Métricas de eventos de decisión y API](#)
- [Métricas de Amazon SWF](#)
- [Nombres y dimensiones de los recursos de Amazon SWF que no son ASCII CloudWatch](#)

Unidades de informes para métricas de Amazon SWF

Métricas que registran un intervalo de tiempo

Algunas de las métricas de Amazon SWF CloudWatch son intervalos de tiempo, siempre medidos en milisegundos. La CloudWatch unidad se indica como Time. Estas métricas, por lo general, corresponden a las fases de la ejecución del flujo de trabajo para los que se puede definir los tiempos de espera de flujo de trabajo y actividad y tienen nombres similares.

Por ejemplo, la métrica `DecisionTaskStartToCloseTime` mide el tiempo necesario para completar la tarea de decisión después de empezar a ejecutarse, que es el mismo periodo de tiempo para el que puede establecer un valor `DecisionTaskStartToCloseTimeout`.

Para ver un diagrama de cada una de estas fases del flujo de trabajo y saber cuándo se producen a lo largo de los ciclos de vida del flujo de trabajo y la actividad, consulte [Tipos de tiempo de espera de Amazon SWF](#).

Métricas que registran un recuento

Algunas de las métricas de Amazon SWF para registrar los CloudWatch resultados como un recuento. Por ejemplo, `WorkflowsCanceled`, registra un resultado como uno o cero, indicando si el flujo de trabajo se ha cancelado o no. Un valor de cero no indica que la métrica no se ha registrado, solo que la condición descrita por la métrica no se produjo.

Algunas de las métricas de Amazon SWF para CloudWatch ese informe CloudWatch son un Count recuento por segundo. Por ejemplo `ProvisionedRefillRate`, si se indica como entrada Count CloudWatch, representa una tasa Count de solicitudes por segundo.

Para contar las métricas, el mínimo y máximo siempre serán cero o uno, pero el valor promedio será un valor de cero a uno.

Métricas de eventos de decisión y API

Puede supervisar tanto los eventos de la API como los de Decision CloudWatch para obtener información sobre su uso y capacidad. Consulte los [decisores](#) en la sección [Conceptos básicos de flujo de trabajo en Amazon SWF](#) y el tema [Decisión](#) en la [referencia de la API de Amazon Simple Workflow Service](#).

También se pueden monitorear estos límites para activar una alarma cuando el usuario se aproxime a sus límites de limitación controlada de Amazon SWF. Consulte [Cuotas de limitación de Amazon SWF](#) para obtener una descripción de estos límites y su configuración predeterminada. Estos límites se han diseñado para impedir que flujos de trabajo incorrectos consuman recursos excesivos del sistema. Para solicitar un aumento de sus límites, consulte: [???](#).

Como práctica recomendada, debes configurar CloudWatch las alarmas en torno al 60% de la capacidad de tu API o de los eventos de decisión. Esto le permitirá ajustar el flujo de trabajo o solicitar un aumento del límite del servicio antes de que se active la limitación controlada de Amazon SWF. Dependiendo de la [explosividad](#) de sus llamadas, puede configurar diversas alarmas para notificar cuando se aproxime a sus límites del servicio:

- Si su tráfico tiene picos significativos, establezca una alarma en el 60 % de sus límites `ProvisionedBucketSize`.
- Si sus llamadas tienen una tasa relativamente estable, establezca una alarma en el 60 % de su límite `ProvisionedRefillRate` para sus eventos de decisión y API relacionados.

Métricas de Amazon SWF

Las siguientes métricas están disponibles para Amazon SWF:

Métrica	Description (Descripción)
<code>DecisionTaskScheduleToStartTime</code>	<p>El intervalo de tiempo, en milisegundos, entre el momento en que se programó la tarea de decisión y el momento en que la seleccionó un proceso de trabajo y la inició.</p> <p>CloudWatch Unidades: Time</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Average, Minimum, Maximum</p>
<code>DecisionTaskStartToCloseTime</code>	<p>El intervalo de tiempo, en milisegundos, entre el momento en que se inició la decisión y el momento en que se cerró.</p> <p>CloudWatch Unidades: Time</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Average, Minimum, Maximum</p>
<code>DecisionTasksCompleted</code>	<p>El número de tareas de decisión que se han completado.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>

Métrica	Description (Descripción)
PendingTasks	<p>Número de tareas pendientes en un intervalo de 1 minuto para una lista de tareas específica.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, TaskListName</p> <p>Estadísticas válidas: Sum</p>
StartedDecisionTasksTimedOutOnClose	<p>El número de tareas de decisión que se iniciaron pero cuyo tiempo de espera se agotó durante el cierre.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>
WorkflowStartToCloseTime	<p>El tiempo, en milisegundos, entre el momento en que se inició el flujo de trabajo y el momento en que se cerró.</p> <p>CloudWatch Unidades: Time</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Average, Minimum, Maximum</p>
WorkflowsCanceled	<p>El número de flujos de trabajo que se cancelaron.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>

Métrica	Description (Descripción)
WorkflowsCompleted	<p>El número de flujos de trabajo que se completaron.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>
WorkflowsContinuedAsNew	<p>El número de flujos de trabajo que continuaron como nuevos.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>
WorkflowsFailed	<p>El número de flujos de trabajo que produjeron un error.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>
WorkflowsTerminated	<p>El número de flujos de trabajo que se terminaron.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Cause, Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>

Métrica	Description (Descripción)
<code>WorkflowsTimedOut</code>	<p>El número de flujos de trabajo para los que se agotó el tiempo de espera, por cualquier motivo.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Estadísticas válidas: Sum</p>
<code>ActivityTaskScheduleToCloseTime</code>	<p>El intervalo de tiempo, en milisegundos, entre el momento en que se programó la actividad y el momento en que se cerró.</p> <p>CloudWatch Unidades: Time</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Average, Minimum, Maximum</p>
<code>ActivityTaskScheduleStartTime</code>	<p>El intervalo de tiempo, en milisegundos, entre el momento en que se programó la tarea de actividad y el momento en que se inició.</p> <p>CloudWatch Unidades: Time</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Average, Minimum, Maximum</p>

Métrica	Description (Descripción)
ActivityTaskStartToCloseTime	<p>El intervalo de tiempo, en milisegundos, entre el momento en que se inició la tarea de actividad y el momento en que se cerró.</p> <p>CloudWatch Unidades: Time</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Average, Minimum, Maximum</p>
ActivityTasksCancelled	<p>El número de tareas de actividad que se cancelaron.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Sum</p>
ActivityTasksCompleted	<p>El número de tareas de actividad que se completaron.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Sum</p>
ActivityTasksFailed	<p>El número de tareas de actividad que produjeron un error.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Sum</p>

Métrica	Description (Descripción)
<code>ScheduledActivityTasksTimedOutOnClose</code>	<p>El número de tareas de actividad que se programaron pero cuyo tiempo de espera se agotó durante el cierre.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Sum</p>
<code>ScheduledActivityTasksTimedOutOnStart</code>	<p>El número de tareas de actividad que se programaron pero cuyo tiempo de espera se agotó durante el inicio.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Sum</p>
<code>StartedActivityTasksTimedOutOnClose</code>	<p>El número de tareas de actividad que se iniciaron pero cuyo tiempo de espera se agotó durante el cierre.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Sum</p>

Métrica	Description (Descripción)
StartedActivityTasksTimedOutOnHeartbeat	<p>El número de tareas de actividad que se iniciaron pero cuyo tiempo de espera se agotó debido a que se agotó el tiempo de espera del latido.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Estadísticas válidas: Sum</p>
ThrottledEvents	<p>El número de solicitudes que han sido objeto de una limitación controlada.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: APIName, DecisionName, ThrottlingScope</p> <p>Estadísticas válidas: Sum</p>
ProvisionedBucketSize	<p>El número de solicitudes disponibles por segundo.</p> <p>Dimensiones: APIName, DecisionName</p> <p>Estadísticas válidas: Minimum</p>
ConsumedCapacity	<p>El número de solicitudes por segundo.</p> <p>CloudWatch Unidades: Count</p> <p>Dimensiones: APIName, DecisionName</p> <p>Estadísticas válidas: Sum</p>
ConsumedLimit	<p>La cantidad del límite general que se ha consumido.</p> <p>Dimensiones: GeneralLimitType</p>

Métrica	Description (Descripción)
ProvisionedRefillRate	<p>El número de solicitudes por segundo que se permiten en el bucket.</p> <p>Dimensiones: APIName, DecisionName</p> <p>Estadísticas válidas: Minimum</p>
ProvisionedLimit	<p>La cantidad del límite general que se aprovisiona a la cuenta.</p> <p>Dimensiones: GeneralLimitType</p>

Dimensión	Description (Descripción)
Domain	Filtra datos al dominio de Amazon SWF en el que se ejecuta el flujo de trabajo o la actividad.
ActivityTypeName	Filtra datos al nombre del tipo de actividad.
ActivityTypeVersion	Filtra datos a la versión del tipo de actividad.
WorkflowTypeName	Filtra datos al nombre del tipo de flujo de trabajo para esta ejecución de flujo de trabajo.
WorkflowTypeVersion	Filtra datos a la versión del tipo de flujo de trabajo para esta ejecución de flujo de trabajo.
APIName	Filtra datos a una API con el nombre de API especificado.
DecisionName	Filtra datos al nombre de decisión especificado.
TaskListName	Filtra datos al nombre de Lista de tareas especificado.
TaskListClassification	Filtra datos a la clasificación de la lista de tareas. El valor es "D" para las listas de tareas de decisión y "A" para las listas de tareas de actividad.

Dimensión	Description (Descripción)
ThrottlingScope	Filtra los datos según el alcance de regulación especificado. El valor es «Cuenta» cuando se supera la cuota a nivel de cuenta, o «Flujo de trabajo» cuando se supera la cuota a nivel de flujo de trabajo.

Nombres y dimensiones de los recursos de Amazon SWF que no son ASCII CloudWatch

Amazon SWF permite caracteres no ASCII en nombres de recursos como y. TaskList DomainName Sin embargo, los valores de dimensión de CloudWatch las métricas solo pueden contener caracteres ASCII imprimibles. Para garantizar que Amazon SWF utiliza valores de dimensión compatibles con los [CloudWatch requisitos](#), los nombres de los recursos de Amazon SWF que no cumplen estos requisitos se convierten y se les añade una suma de control de la siguiente manera:

- Cualquier carácter que no sea ASCII se sustituye por ?.
- La cadena de entrada o la cadena convertida se truncará, si es necesario. Esto garantiza que, cuando se añada la suma de comprobación, la nueva longitud de la cadena no supere el máximo. CloudWatch
- Como todos los caracteres que no son ASCII se convierten en caracteres?, algunos valores de dimensiones CloudWatch métricas que eran diferentes antes de la conversión pueden parecer los mismos después de la conversión. Para ayudar a diferenciarlos, se añade al nombre del recurso un guión bajo (_) seguido de los primeros 16 caracteres de la SHA256 suma de verificación del nombre del recurso original.

Ejemplos de conversión:

- test àpple se convertiría en test ?pple_82cc5b8e3a771d12.
- à` se convertiría en ???_2fec5edbb2c05c22.
- Ambos TaskList nombres àpplé se âpplè convertirían en nombres idénticos y serían idénticos. ?pp1? Al agregar la suma de comprobación, se obtienen valores distintos: ?pp1?_f39a36df9d85a69d y ?pp1?_da3efb4f11dd0f7f.

i Tip

Puede generar su propia SHA256 suma de verificación. Por ejemplo, para utilizar la herramienta de línea de comandos shasum:

```
echo -n "<the original resource name>" | shasum -a 256 | cut -c1-16
```

Visualización de las métricas de Amazon SWF para usar el CloudWatch Consola de administración de AWS

Amazon CloudWatch proporciona una serie de métricas visibles para los flujos de trabajo y las actividades de Amazon SWF. Puede ver las métricas y establecer alarmas para las ejecuciones de flujos de trabajo de Amazon SWF con la [Consola de administración de AWS](#). Debe iniciar sesión en la consola para continuar.

Para obtener una descripción de cada una de las métricas disponibles, consulte [Métricas de Amazon SWF para CloudWatch](#).

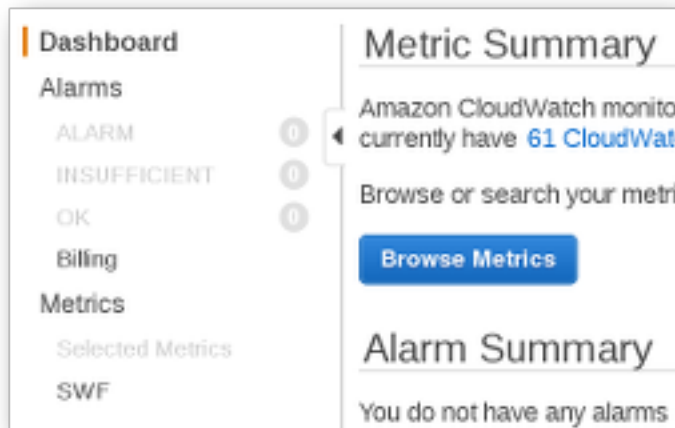
Temas

- [Visualización de métricas](#)
- [Configuración de alarmas](#)

Visualización de métricas

Cómo ver las métricas de Amazon SWF

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en. <https://console.aws.amazon.com/cloudwatch/>
2. En el panel de navegación izquierdo, en Metrics, elija SWF.



Si ha ejecutado alguna ejecución de flujo de trabajo recientemente, verá dos listas de métricas presentadas: las métricas acerca de los tipos de flujo de trabajo y las métricas acerca de los tipos de actividad.

Showing all results (61) for SWF Metrics.

Select All | Clear

SWF > Workflow Type Metrics

Domain	WorkflowTypeName	WorkflowTypeVersion	Metric Name
HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowStartToCloseTime
HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowsCompleted

SWF > Activity Type Metrics

Domain	ActivityTypeName	ActivityTypeVersion	Metric Name
Booking	BookingActivity.reserve_airline	1.0	ActivityTaskScheduleToStartTime
Booking	BookingActivity.reserve_airline	1.0	ActivityTaskStartToCloseTime

Note

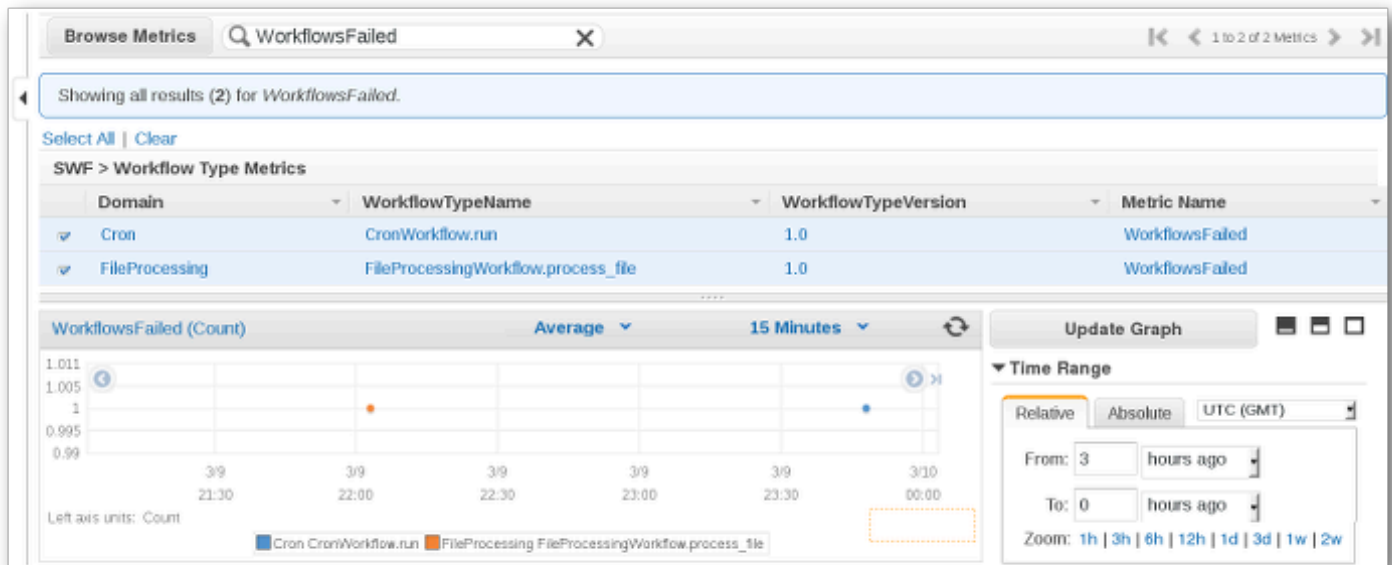
Puede que inicialmente solo vea las métricas acerca de los tipos de flujo de trabajo; las métricas acerca de los tipos de actividad se presentan en la misma vista, pero es posible que tenga que desplazarse hacia abajo para verlas.

Hasta 50 de las métricas más recientes se mostrarán a la vez, divididas entre métricas de actividad y de flujo de trabajo.

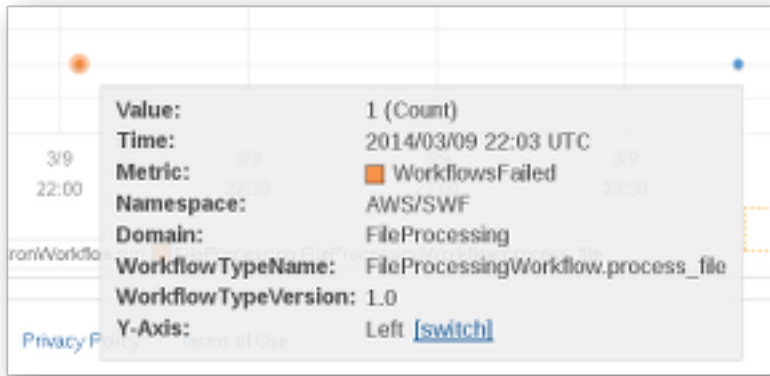
Puede usar los encabezados interactivos que aparecen sobre cada columna de la lista para ordenar sus métricas mediante cualquiera de las dimensiones proporcionadas. En el caso de los flujos de trabajo, las dimensiones son el dominio `WorkflowTypeNameWorkflowTypeVersion`, y el nombre de la métrica. En el caso de las actividades, las dimensiones son el dominio `ActivityTypeNameActivityTypeVersion`, y el nombre de la métrica.

Los diversos tipos de métricas se describen en [Métricas de Amazon SWF para CloudWatch](#).

Puede ver gráficos para métricas eligiendo los cuadros situados junto a la fila de métricas en la lista y cambiar los parámetros del gráfico mediante los controles Time Range ubicados a la derecha de la vista de gráficos.



Para obtener información acerca de cualquier punto del gráfico, sitúe el cursor sobre el punto del gráfico. Se mostrará un detalle de las dimensiones del punto.



Para obtener más información sobre cómo trabajar con CloudWatch métricas, consulte [Visualización, representación gráfica y publicación de métricas](#) en la Guía del CloudWatch usuario de Amazon.

Configuración de alarmas

Puede utilizar CloudWatch las alarmas para realizar acciones como avisarle cuando se alcance un umbral de alarma. Por ejemplo, puede establecer una alarma para enviar una notificación a un tema de SNS o para enviar un correo electrónico cuando la métrica `WorkflowsFailed` supere un umbral determinado.

Para establecer una alarma en cualquiera de sus métricas

1. Seleccione una sola métrica eligiendo su cuadro.
2. A la derecha del gráfico, en los controles Tools, elija Create Alarm.
3. En la pantalla Define Alarm, escriba el valor de umbral de la alarma, los parámetros de periodo y las medidas que se van a tomar.

The screenshot shows the 'Define Alarm' configuration page in the Amazon CloudWatch console. It is divided into two main sections: 'Alarm Threshold' and 'Actions'.

1. Select Metric
2. Define Alarm

Navigation buttons: Back, Next, Cancel.

Alarm Threshold
Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: [text input]
Description: [text input]

Whenever: WorkflowsFailed
is: >= [dropdown] 1 [input]
for: 2 [input] consecutive period(s)

Actions
Define what actions are taken when your alarm changes state.

Notification [Delete]

Whenever this alarm: State is ALARM [dropdown]
Send notification to: SWF_Sample_Topic [dropdown] [New list]
Email list: me@example.com [input]

Buttons: + Notification, + AutoScaling Action, + EC2 Action

Para obtener más información sobre la configuración y el uso de CloudWatch alarmas, consulte [Creación de CloudWatch alarmas de Amazon](#) en la Guía del CloudWatch usuario de Amazon.

Grabación de llamadas a la API con AWS CloudTrail

Amazon Simple Workflow Service está integrado con [AWS CloudTrail](#) un servicio que proporciona un registro de las acciones realizadas por un usuario, rol o un Servicio de AWS. CloudTrail captura todas las llamadas a la API de Amazon SWF como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de Amazon SWF y las llamadas desde el código a las operaciones de la API de Amazon SWF. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon SWF, la dirección IP desde la que se realizó la solicitud, cuándo se realizó y detalles adicionales.

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

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario.

- Si la solicitud se realizó en nombre de un usuario de IAM Identity Center.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro Servicio de AWS.

CloudTrail está activa en tu cuenta Cuenta de AWS cuando creas la cuenta y tienes acceso automáticamente al historial de CloudTrail eventos. El historial de CloudTrail eventos proporciona un registro visible, consultable, descargable e inmutable de los últimos 90 días de eventos de gestión registrados en un. Región de AWS Para obtener más información, consulte [Uso del historial de CloudTrail eventos en la Guía del usuario](#). AWS CloudTrail La visualización del historial de eventos no conlleva ningún CloudTrail cargo.

Para tener un registro continuo de los eventos de Cuenta de AWS los últimos 90 días, crea un almacén de datos de eventos de senderos o [CloudTrail lagos](#).

CloudTrail senderos

Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. Todos los senderos creados con él Consola de administración de AWS son multirregionales. Puede crear un registro de seguimiento de una sola región o multirregionales mediante la AWS CLI. Se recomienda crear un sendero multirregional, ya que puedes capturar toda la actividad de tu Regiones de AWS cuenta. Si crea un registro de seguimiento de una sola región, solo podrá ver los eventos registrados en la Región de AWS del registro de seguimiento. Para obtener más información acerca de los registros de seguimiento, consulte [Creación de un registro de seguimiento para su Cuenta de AWS](#) y [Creación de un registro de seguimiento para una organización](#) en la Guía del usuario de AWS CloudTrail .

Puede enviar una copia de sus eventos de administración en curso a su bucket de Amazon S3 sin coste alguno CloudTrail mediante la creación de una ruta; sin embargo, hay cargos por almacenamiento en Amazon S3. Para obtener más información sobre CloudTrail los precios, consulte [AWS CloudTrail Precios](#). Para obtener información acerca de los precios de Amazon S3, consulte [Precios de Amazon S3](#).

CloudTrail Almacenes de datos de eventos en Lake

CloudTrail Lake le permite ejecutar consultas basadas en SQL en sus eventos. CloudTrail Lake convierte los eventos existentes en formato JSON basado en filas al formato [Apache](#) ORC. ORC es un formato de almacenamiento en columnas optimizado para una recuperación rápida

de datos. Los eventos se agregan en almacenes de datos de eventos, que son recopilaciones inmutables de eventos en función de criterios que se seleccionan aplicando [selectores de eventos avanzados](#). Los selectores que se aplican a un almacén de datos de eventos controlan los eventos que perduran y están disponibles para la consulta. Para obtener más información sobre CloudTrail Lake, consulte [Cómo trabajar con AWS CloudTrail Lake](#) en la Guía del AWS CloudTrail usuario.

CloudTrail Los almacenes de datos y las consultas sobre eventos de Lake conllevan costes. Cuando crea un almacén de datos de eventos, debe elegir la [opción de precios](#) que desee utilizar para él. La opción de precios determina el costo de la incorporación y el almacenamiento de los eventos, así como el período de retención predeterminado y máximo del almacén de datos de eventos. Para obtener más información sobre CloudTrail los precios, consulte [AWS CloudTrail Precios](#).

Eventos de datos en CloudTrail

Los [eventos de datos](#) proporcionan información sobre las operaciones de recursos realizadas en o dentro de un recurso (por ejemplo, leer o escribir en un objeto de Amazon S3). Se denominan también operaciones del plano de datos. Los eventos de datos suelen ser actividades de gran volumen. De forma predeterminada, CloudTrail no registra los eventos de datos. El historial de CloudTrail eventos no registra los eventos de datos.

Se aplican cargos adicionales a los eventos de datos. Para obtener más información sobre CloudTrail los precios, consulta [AWS CloudTrail Precios](#).

Puede registrar eventos de datos para los tipos de recursos de Amazon SWF mediante la CloudTrail consola o las operaciones AWS CLI de la CloudTrail API. Para obtener más información sobre cómo registrar los eventos de datos, consulte [Registro de eventos de datos con la Consola de administración de AWS](#) y [Registro de eventos de datos con la AWS Command Line Interface](#) en la Guía del usuario de AWS CloudTrail .

En la siguiente tabla se enumeran los tipos de recursos de Amazon SWF para los que puede registrar eventos de datos. La columna de tipos de eventos de datos muestra el valor que se puede elegir en la lista de tipos de eventos de datos de la CloudTrail consola. La columna de valores `resources.type` muestra el `resources.type` valor que se debe especificar al configurar los selectores de eventos avanzados mediante la tecla o. AWS CLI CloudTrail APIs La CloudTrail columna Datos APIs registrados muestra las llamadas a la API registradas CloudTrail para el tipo de recurso.

Puede configurar selectores de eventos avanzados para filtrar según los campos `eventName`, `readOnly` y `resources.ARN` y así registrar solo los eventos que son importantes para usted. Para obtener más información acerca de estos campos, consulte [AdvancedFieldSelector](#) en la Referencia de la API de AWS CloudTrail .

Tipos de eventos de datos	resources.type value	Datos APIs registrados en CloudTrail
Dominio SWF	AWS::SWF::Domain	<p>Eventos de flujo de trabajo</p> <ul style="list-style-type: none"> • CountClosedWorkflowExecutions • CountOpenWorkflowExecutions • DescribeWorkflowExecution • ListClosedWorkflowExecutions • ListOpenWorkflowExecutions • GetWorkflowExecutionHistory • RequestCancelWorkflowExecution • SignalWorkflowExecution • StartWorkflowExecution • TerminateWorkflowExecution <p>Eventos de tareas</p> <ul style="list-style-type: none"> • CountPendingActivityTasks • PollForDecisionTask • PollForActivityTask • RecordActivityTaskHeartbeat

Tipos de eventos de datos	resources.type value	Datos APIs registrados en CloudTrail
		<ul style="list-style-type: none"> • RespondActivityTaskCanceled • RespondActivityTaskCompleted • RespondActivityTaskFailed • RespondDecisionTaskCompleted <p>Eventos de decisión</p> <ul style="list-style-type: none"> • CancelTimer • CancelWorkflowExecution • CompleteWorkflowExecution • ContinueAsNewWorkflowExecution • FailWorkflowExecution • RecordMarker • RequestCancelActivityTask • RequestCancelExternalWorkflowExecution • ScheduleActivityTask • ScheduleLambdaFunction • SignalExternalWorkflowExecution • StartChildWorkflowExecution • StartTimer

CloudTrail eventos y RespondDecisionTaskCompleted

La [RespondDecisionTaskCompleted](#) acción toma una lista de decisiones en la carga útil de la solicitud. Una llamada completada emitirá N+1 eventos de CloudTrail datos, uno para cada decisión y otro para la propia llamada a la API. Los eventos de datos y el evento de API tendrán todos el mismo identificador de solicitud.

Eventos de gestión en CloudTrail

[Los eventos de administración](#) proporcionan información sobre las operaciones de administración que se llevan a cabo en los recursos de su empresa Cuenta de AWS. Se denominan también operaciones del plano de control. De forma predeterminada, CloudTrail registra los eventos de administración.

Amazon Simple Workflow Service registra las siguientes operaciones del plano de control CloudTrail como eventos de administración.

Eventos de dominio

- [RegisterDomain](#)
- [DescribeDomain](#)
- [ListDomains](#)
- [DeprecateDomain](#)
- [UndeprecateDomain](#)

Eventos de actividad

- [RegisterActivityType](#)
- [DescribeActivityType](#)
- [ListActivityTypes](#)
- [DeprecateActivityType](#)
- [UndeprecateActivityType](#)
- [DeleteActivityType](#)

WorkflowType Eventos

- [RegisterWorkflowType](#)
- [DescribeWorkflowType](#)
- [ListWorkflowTypes](#)
- [DeprecateWorkflowType](#)
- [UndeprecateWorkflowType](#)
- [DeleteWorkflowType](#)

Etiqueta eventos

- [TagResource](#)
- [UntagResource](#)
- [ListTagsforResource](#)

Evento de ejemplo de

Un evento representa una solicitud única de cualquier fuente e incluye información sobre la operación de API solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que los eventos no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra un CloudTrail evento que demuestra la CountClosedWorkflowExecutions operación.

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "1234567890abcdef02345:admin",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/admin",
    "accountId": "111122223333",
    "accessKeyId": "abcdef01234567890abc",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "1234567890abcdef02345",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      }
    }
  }
}
```

```
    },
    "attributes": {
      "creationDate": "2023-11-23T16:37:38Z",
      "mfaAuthenticated": "false"
    }
  },
  "eventTime": "2023-11-23T17:52:46Z",
  "eventSource": "swf.amazonaws.com",
  "eventName": "CountClosedWorkflowExecutions",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "198.51.100.42",
  "userAgent": "aws-internal/3 aws-sdk-java/1.11.42",
  "requestParameters": {
    "domain": "nsg-domain",
    "closeTimeFilter": {
      "oldestDate": "Nov 23, 2023 5:52:46 PM",
      "latestDate": "Nov 23, 2023 5:52:46 PM"
    }
  },
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::SWF::Domain",
      "ARN": "arn:aws:swf:us-east-1:111122223333:/domain/nsg-domain"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "111122223333",
  "eventCategory": "Data",
  "tlsDetails": {
    "clientProvidedHostHeader": "swf.example.amazondomains.com"
  }
}
```

Para obtener información sobre el contenido de los CloudTrail registros, consulte el [contenido de los CloudTrail registros](#) en la Guía del AWS CloudTrail usuario.

EventBridge para cambios en el estado de ejecución de Amazon SWF

Utilizas Amazon EventBridge para responder a cambios de estado o eventos en un AWS recurso. Cuando Amazon SWF emite un evento, siempre va al bus de EventBridge eventos predeterminado de su cuenta. Puede crear una regla para los eventos, asociarla al bus de eventos predeterminado y especificar la acción objetivo que se llevará a cabo cuando EventBridge reciba un evento que coincida con la regla. De esa forma, se pueden monitorizar los flujos de trabajo sin tener que sondear constantemente mediante la API [GetWorkflowExecutionHistory](#). En función de los cambios en las ejecuciones del flujo de trabajo, puede utilizar un EventBridge objetivo para llamar a AWS Lambda funciones, publicar mensajes en temas del Amazon Simple Notification Service (Amazon SNS) y mucho más.

Puede ver el contenido completo de un evento de cambio de estado de ejecución mediante [DescribeWorkflowExecution](#).

Para obtener más información, consulta la [Guía del EventBridge usuario de Amazon](#).

EventBridge eventos

Los tipos de eventos del historial contienen los cambios en el estado de ejecución. La sección `detail` de cada evento contiene al menos los siguientes parámetros:

- `eventId`: el identificador del evento mostrado por `GetWorkflowExecutionHistory`.
- `workflowExecutionDetail`: el estado del flujo de trabajo en el momento en que se emitió el evento.
- `eventType`: el tipo de evento del historial, que puede ser alguno de los que se muestran a continuación:
 - `ActivityTaskCanceled`
 - `ActivityTaskFailed`
 - `ActivityTaskTimedOut`
 - `WorkflowExecutionCanceled`
 - `WorkflowExecutionCompleted`
 - `WorkflowExecutionFailed`
 - `WorkflowExecutionStarted`
 - `WorkflowExecutionTerminated`
 - `WorkflowExecutionTimedOut`

- `WorkflowExecutionContinuedAsNew`
- `CancelTimerFailed`
- `CancelWorkflowExecutionFailed`
- `ChildWorkflowExecutionFailed`
- `ChildWorkflowExecutionTimedOut`
- `CompleteWorkflowExecutionFailed`
- `ContinueAsNewWorkflowExecutionFailed`
- `DecisionTaskTimedOut`
- `FailWorkflowExecutionFailed`
- `RecordMarkerFailed`
- `RequestCancelActivityTaskFailed`
- `RequestCancelExternalWorkflowExecutionFailed`
- `ScheduleActivityTaskFailed`
- `SignalExternalWorkflowExecutionFailed`
- `StartActivityTaskFailed`
- `StartChildWorkflowExecutionFailed`
- `StartTimerFailed`
- `TimerCanceled`
- `LambdaFunctionFailed`
- `LambdaFunctionTimedOut`
- `StartLambdaFunctionFailed`
- `ScheduleLambdaFunctionFailed`

Ejemplos de eventos de Amazon SWF

Los siguientes son ejemplos de cómo Amazon SWF envía eventos a: EventBridge

Temas

- [Ejecución iniciada](#)
- [Ejecución completada](#)
- [Ejecución errónea](#)

- [Ejecución con tiempo de espera agotado](#)
- [Ejecución finalizada](#)

En todos los casos, la sección `detail` de los datos del evento proporciona la misma información que la API [DescribeWorkflowExecution](#). El campo `executionStatus` indica el estado de la ejecución en el momento en el que se envió el evento, que puede ser `OPEN` o `CLOSED`.

Ejecución iniciada

```
{
  "version": "0",
  "id": "44444444444444",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "44444444444444",
  "time": "2020-05-08T15:57:38Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:44444444444444:/domain/SimpleWorkflowUserSimulator"
  ],
  "detail": {
    "eventId": 1,
    "eventType": "WorkflowExecutionStarted",
    "workflowExecutionDetail": {
      "executionInfo": {
        "execution": {
          "workflowId": "123456789012",
          "runId": "AKIAIOSFODNN7EXAMPLE"
        },
        "workflowType": {
          "name": "SimpleWorkflowUserSimulator",
          "version": "myWorkflow"
        },
        "startTimestamp": 1588953458484,
        "closeTimestamp": null,
        "executionStatus": "OPEN",
        "closeStatus": null,
        "parent": null,
        "parentExecutionArn": null,
        "tagList": null,
        "cancelRequested": false
      },
    },
  },
}
```

```

    "executionConfiguration": {
      "taskStartToCloseTimeout": "60",
      "executionStartToCloseTimeout": "1000",
      "taskList": {
        "name": "44444444444444"
      },
      "taskPriority": null,
      "childPolicy": "ABANDON",
      "lambdaRole": "arn:aws:iam::44444444444444:role/BasicSWFLambdaExecution"
    },
    "openCounts": {
      "openActivityTasks": 0,
      "openDecisionTasks": 1,
      "openTimers": 0,
      "openChildWorkflowExecutions": 0,
      "openLambdaFunctions": 0
    },
    "latestActivityTaskTimestamp": null,
  }
}
}

```

Ejecución completada

```

{
  "version": "0",
  "id": "1111-2222-3333",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "444455556666",
  "time": "2020-05-08T15:57:39Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:444455556666:/domain/SimpleWorkflowUserSimulator"
  ],
  "detail": {
    "eventId": 35,
    "eventType": "WorkflowExecutionCompleted",
    "workflowExecutionDetail": {
      "executionInfo": {
        "execution": {
          "workflowId": "1234-5678-9012",
          "runId": "777788889999"
        }
      }
    }
  }
}

```

```

    },
    "workflowType": {
      "name": "SimpleWorkflowUserSimulator",
      "version": "myWorkflow"
    },
    },
    "startTimestamp": 1588953458820,
    "closeTimestamp": 1588953459448,
    "executionStatus": "CLOSED",
    "closeStatus": "COMPLETED",
    "parent": null,
    "parentExecutionArn": null,
    "tagList": null,
    "cancelRequested": false
  },
  "executionConfiguration": {
    "taskStartToCloseTimeout": "60",
    "executionStartToCloseTimeout": "1000",
    "taskList": {
      "name": "1111-1111-1111"
    },
    },
    "taskPriority": null,
    "childPolicy": "ABANDON",
    "lambdaRole": "arn:aws:iam::444455556666:role/BasicSWFLambdaExecution"
  },
  "openCounts": {
    "openActivityTasks": 0,
    "openDecisionTasks": 0,
    "openTimers": 0,
    "openChildWorkflowExecutions": 0,
    "openLambdaFunctions": 0
  },
  "latestActivityTaskTimestamp": 1588953459402,
}
}
}

```

Ejecución errónea

```

{
  "version": "0",
  "id": "1111-2222-3333",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",

```

```
"account": "444455556666",
"time": "2020-05-08T15:57:38Z",
"region": "us-east-1",
"resources": [
  "arn:aws:swf:us-east-1:444455556666:/domain/SimpleWorkflowUserSimulator"
],
"detail": {
  "eventId": 11,
  "eventType": "WorkflowExecutionFailed",
  "workflowExecutionDetail": {
    "executionInfo": {
      "execution": {
        "workflowId": "1234-5678-9012",
        "runId": "777788889999"
      },
      "workflowType": {
        "name": "SimpleWorkflowUserSimulator",
        "version": "myWorkflow"
      },
      "startTimestamp": 1588953158481,
      "closeTimestamp": 1588953458560,
      "executionStatus": "CLOSED",
      "closeStatus": "FAILED",
      "parent": null,
      "parentExecutionArn": null,
      "tagList": null,
      "cancelRequested": false
    },
    "executionConfiguration": {
      "taskStartToCloseTimeout": "60",
      "executionStartToCloseTimeout": "1000",
      "taskList": {
        "name": "1111-1111-1111"
      },
      "taskPriority": null,
      "childPolicy": "ABANDON",
      "lambdaRole": "arn:aws:iam::444455556666:role/BasicSWFLambdaExecution"
    },
    "openCounts": {
      "openActivityTasks": 0,
      "openDecisionTasks": 0,
      "openTimers": 0,
      "openChildWorkflowExecutions": 0,
      "openLambdaFunctions": 0
    }
  }
}
```

```
    },
    "latestActivityTaskTimestamp": null,
  }
}
```

Ejecución con tiempo de espera agotado

```
{
  "version": "0",
  "id": "1111-2222-3333",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "444455556666",
  "time": "2020-05-05T17:26:30Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:444455556666:/domain/SimpleWorkflowUserSimulator"
  ],
  "detail": {
    "eventId": 6,
    "eventType": "WorkflowExecutionTimedOut",
    "workflowExecutionDetail": {
      "executionInfo": {
        "execution": {
          "workflowId": "1234-5678-9012",
          "runId": "777788889999"
        },
        "workflowType": {
          "name": "SimpleWorkflowUserSimulator",
          "version": "myWorkflow"
        },
        "startTimestamp": 1588698073748,
        "closeTimestamp": 1588699590745,
        "executionStatus": "CLOSED",
        "closeStatus": "TIMED_OUT",
        "parent": null,
        "parentExecutionArn": null,
        "tagList": null,
        "cancelRequested": false
      },
      "executionConfiguration": {
        "taskStartToCloseTimeout": "60",

```

```

    "executionStartToCloseTimeout": "1000",
    "taskList": {
      "name": "1111-1111-1111"
    },
    "taskPriority": null,
    "childPolicy": "ABANDON",
    "lambdaRole": "arn:aws:iam::444455556666:role/BasicSWFLambdaExecution"
  },
  "openCounts": {
    "openActivityTasks": 1,
    "openDecisionTasks": 0,
    "openTimers": 0,
    "openChildWorkflowExecutions": 0,
    "openLambdaFunctions": 0
  },
  "latestActivityTaskTimestamp": 1588699585802,
}
}
}

```

Ejecución finalizada

```

{
  "version": "0",
  "id": "1111-2222-3333",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "444455556666",
  "time": "2020-05-08T22:37:26Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:444455556666:/domain/canary"
  ],
  "detail": {
    "eventId": 48,
    "eventType": "WorkflowExecutionTerminated",
    "workflowExecutionDetail": {
      "executionInfo": {
        "execution": {
          "workflowId": "1234-5678-9012",
          "runId": "777788889999"
        },
        "workflowType": {

```

```
    "name": "1111-1111-1111",
    "version": "1.3"
  },
  "startTimestamp": 1588977445279,
  "closeTimestamp": 1588977446062,
  "executionStatus": "CLOSED",
  "closeStatus": "TERMINATED",
  "parent": null,
  "parentExecutionArn": null,
  "tagList": null,
  "cancelRequested": false
},
"executionConfiguration": {
  "taskStartToCloseTimeout": "60",
  "executionStartToCloseTimeout": "120",
  "taskList": {
    "name": "1111-1111-1111-2222-2222-2222"
  },
  "taskPriority": null,
  "childPolicy": "TERMINATE",
  "lambdaRole": null
},
"openCounts": {
  "openActivityTasks": 0,
  "openDecisionTasks": 1,
  "openTimers": 0,
  "openChildWorkflowExecutions": 0,
  "openLambdaFunctions": 0
},
"latestActivityTaskTimestamp": 1588977445882,
}
}
```

Uso AWS User Notifications con Amazon Simple Workflow Service

Puede utilizar [AWS User Notifications](#) para configurar los canales de entrega a fin de recibir notificaciones sobre los eventos de Amazon Simple Workflow Service. 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 [Amazon Q Developer en aplicaciones de chat](#) o notificaciones push 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.

Validación de conformidad para Amazon Simple Workflow Service

Varios auditores externos evalúan la seguridad y la conformidad de Amazon Simple Workflow Service como parte de varios programas de conformidad de AWS . Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para obtener una lista de AWS los servicios incluidos en el ámbito de los programas de cumplimiento específicos, consulte los [AWS servicios incluidos en el ámbito de aplicación por programa de cumplimiento](#) y . 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 conformidad al utilizar Amazon SWF se determina en función de la sensibilidad de los datos, los objetivos de cumplimiento de su empresa y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- [Guías de inicio rápido](#) de de seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en la seguridad y el cumplimiento. AWS
- Documento técnico sobre [cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico](#) se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- [AWS Recursos de cumplimiento Recursos](#) de de trabajo y guías puede aplicarse a su sector y ubicación.
- [Evaluación de los recursos con las reglas](#) de 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 CSPM](#)— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia en Amazon Simple Workflow Service

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

[Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura global.AWS](#)

Además de la infraestructura AWS global, Amazon SWF ofrece varias funciones para respaldar sus necesidades de respaldo y resiliencia de datos.

Seguridad de la infraestructura en Amazon Simple Workflow Service

Como servicio gestionado, está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se protege la infraestructura, consulte [Seguridad AWS en la nube](#). Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte [Protección de infraestructuras en un marco](#) de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Puede llamar a estas operaciones de la API desde cualquier ubicación de red, pero Amazon SWF admite políticas de acceso basadas en recursos, que pueden incluir restricciones en función de la dirección IP de origen. También puede utilizar las políticas de Amazon SWF para controlar el acceso desde puntos de enlace específicos o específicos de Amazon Virtual Private Cloud (Amazon VPC).

VPCs En efecto, esto aísla el acceso a la red a un determinado recurso de Amazon SWF únicamente de la VPC específica de la red. AWS

Configuración y análisis de vulnerabilidades en Amazon Simple Workflow Service

La configuración y los controles de TI son una responsabilidad compartida entre usted AWS y usted, nuestro cliente. Para obtener más información, consulte el [modelo de responsabilidad AWS compartida](#).

Uso del AWS CLI servicio Amazon Simple Workflow

Se puede obtener acceso a muchas de las características de Amazon Simple Workflow Service (Amazon SWF) desde la AWS CLI. AWS CLI Ofrece una alternativa al uso de Amazon SWF con Consola de administración de AWS o, en algunos casos, a la programación con la API de Amazon SWF y la. AWS Flow Framework

Por ejemplo, puede utilizarla AWS CLI para registrar un nuevo tipo de flujo de trabajo:

```
aws swf register-workflow-type --domain MyDomain --name "MySimpleWorkflow" --workflow-version "v1"
```

También puede listar sus tipos de flujo de trabajo registrados:

```
aws swf list-workflow-types --domain MyDomain --registration-status REGISTERED
```

A continuación se muestra un ejemplo de la salida predeterminada de JSON:

```
{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1377471607.752,
      "workflowType": {
        "version": "v1",
        "name": "MySimpleWorkflow"
      }
    },
    {
      "status": "REGISTERED",
      "creationDate": 1371454149.598,
      "description": "MyDomain subscribe workflow",
      "workflowType": {
        "version": "v3",
        "name": "subscribe"
      }
    }
  ]
}
```

Los comandos de Amazon SWF AWS CLI permiten iniciar y gestionar las ejecuciones de flujos de trabajo, realizar sondeos para detectar tareas de actividad, registrar los latidos de las tareas, ¡y mucho más! Para obtener una lista completa de los comandos de Amazon SWF (junto con las descripciones de los argumentos disponibles y ejemplos sobre su uso), consulte los comandos de [Amazon SWF](#) en la Referencia de comandos de la AWS CLI .

Los AWS CLI comandos siguen de cerca la API de Amazon SWF, por lo que puede utilizarlos AWS CLI para obtener información sobre la API de Amazon SWF subyacente. También puede servirse de los conocimientos que haya adquirido acerca de las API para crear un prototipo de código o realizar acciones de Amazon SWF en la línea de comandos.

Para obtener más información sobre la AWS CLI, consulte la Guía del [AWS Command Line Interface usuario](#).

Uso de Amazon SWF APIs

Además de utilizar las AWS SDKs que se describen en [Desarrolle con AWS SDKs](#), puede utilizar la API HTTP directamente.

Para utilizar la API, envíe solicitudes HTTP al [punto de conexión de SWF](#) correspondiente a la región que quiere utilizar para sus dominios, flujos de trabajo y actividades. Para obtener más información sobre cómo realizar solicitudes HTTP para Amazon SWF, consulte [Realizar solicitudes HTTP a Amazon SWF](#).

En esta sección, se proporciona información general sobre la utilización de la API de HTTP para desarrollar flujos de trabajo con Amazon SWF. En la sección encontrará funciones más avanzadas, como el uso de temporizadores, el registro CloudTrail y el etiquetado de los flujos de trabajo.

[Conceptos básicos de flujo de trabajo en Amazon SWF](#)

Temas

- [Realizar solicitudes HTTP a Amazon SWF](#)
- [Lista de acciones de Amazon SWF por categoría](#)
- [Cómo registrar un dominio con Amazon SWF](#)
- [Configuración de valores de tiempo de espera en Amazon SWF](#)
- [Registro de un tipo de flujo de trabajo con Amazon SWF](#)
- [Registro de un tipo de actividad con Amazon SWF](#)
- [AWS Lambda tareas en Amazon SWF](#)
- [Desarrollo de un proceso de trabajo de actividad en Amazon SWF](#)
- [Desarrollo de decisores en Amazon SWF](#)
- [Inicio de flujos de trabajo en Amazon SWF](#)
- [Configuración de la prioridad de las tareas en Amazon SWF](#)
- [Gestión de errores en Amazon SWF](#)

Realizar solicitudes HTTP a Amazon SWF

Si no utiliza ninguno de los AWS SDKs, puede realizar operaciones de Amazon Simple Workflow Service (Amazon SWF) a través de HTTP mediante el método de solicitud POST. El método POST

requiere que especifique la operación en el encabezado de la solicitud y proporcione los datos para la operación en formato JSON en el cuerpo de la solicitud.

Contenido de los encabezados HTTP

Amazon SWF requiere que figure la siguiente información en el encabezado de una solicitud HTTP:

- `host`: el punto de conexión de Amazon SWF.
- `x-amz-date`: Debe proporcionar la marca de tiempo en el `Date` encabezado HTTP o en el `AWS x-amz-date header` (algunas bibliotecas de clientes HTTP no le permiten configurar el `Date` encabezado). Cuando hay un encabezado `x-amz-date` presente, el sistema hace caso omiso de cualquier encabezado `Date` al autenticar la solicitud.

La fecha debe especificarse en uno de los tres formatos siguientes, como se especifica en HTTP/1.1 RFC:

- Sun, 06 Nov 1994 08:49:37 GMT (RFC 822, actualizado por RFC 1123)
- Sunday, 06-Nov-94 08:49:37 GMT (RFC 850, obsoleto en RFC 1036)
- Sun Nov 6 08:49:37 1994 (formato `asctime()` de ANSI C)
- `x-amzn-authorization`: los parámetros de solicitud firmados en el formato:

```
AWS3 AWSAccessKeyId=####,Algorithm=HmacSHA256, [,SignedHeaders=Header1;Header2;...]
Signature=S(StringToSign)
```

AWS3— Se trata de una etiqueta AWS específica de la implementación que indica la versión de autenticación utilizada para firmar la solicitud (actualmente, en Amazon SWF, este valor es siempre). **AWS3**

AWSAccessKeyId— Su AWS ID de clave de acceso.

Algorithm— El algoritmo utilizado para crear el valor HMAC-SHA del string-to-sign, como o. **HmacSHA256 HmacSHA1**

Signature— Base64 (algoritmo (,)) `StringToSign`. `SigningKey` Para obtener más información, consulte [Cálculo de la firma HMAC-SHA para Amazon SWF](#).

SignedHeaders— (Opcional) Si está presente, debe contener una lista de todos los encabezados HTTP utilizados en el cálculo `HttpHeaders` canonicalizado. Debe utilizar un solo punto y coma (;) (carácter ASCII 59) para delimitar los valores de la lista.

- `x-amz-target`: el servicio de destino de la solicitud y la operación de los datos, en el formato:

```
com.amazonaws.swf.service.model.SimpleWorkflowService. + <action>
```

Por ejemplo,

```
com.amazonaws.swf.service.model.SimpleWorkflowService.RegisterDomain
```

- `content-type`: el tipo debe especificar JSON y el conjunto de caracteres, como `application/json; charset=UTF-8`

A continuación se muestra un ejemplo de un encabezado en una solicitud HTTP para crear un dominio.

```
POST http://swf.us-east-1.amazonaws.com/ HTTP/1.1
Host: swf.us-east-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.25) Gecko/20111212
  Firefox/3.6.25 ( .NET CLR 3.5.30729; .NET4.0E)
Accept: application/json, text/javascript, */*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Content-Type: application/json; charset=UTF-8
X-Requested-With: XMLHttpRequest
X-Amz-Date: Fri, 13 Jan 2012 18:42:12 GMT
X-Amz-Target: com.amazonaws.swf.service.model.SimpleWorkflowService.RegisterDomain
Content-Encoding: amz-1.0
X-Amzn-Authorization: AWS3
  AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Algorithm=HmacSHA256,SignedHeaders=Host;X-Amz-
  Date;X-Amz-Target;Content-Encoding,Signature=tzjkF551xAxPhzp/BRGFYQRQrQ6CqRM254dTDE/
  EncI=
Referer: http://swf.us-east-1.amazonaws.com/explorer/index.html
Content-Length: 91
Pragma: no-cache
Cache-Control: no-cache

{"name": "867530902",
 "description": "music",
 "workflowExecutionRetentionPeriodInDays": "60"}
```

Este es un ejemplo de la respuesta HTTP correspondiente.

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/json
x-amzn-RequestId: 4ec4ac3f-3e16-11e1-9b11-7182192d0b57
```

Contenido de cuerpo HTTP

El cuerpo de una solicitud HTTP contiene los datos de la operación especificada en el encabezado de la solicitud HTTP. Use el formato de datos JSON para transmitir los valores de los datos y la estructura de datos, de forma simultánea. Los elementos se pueden anidar en otros elementos mediante la notación de corchete. Por ejemplo, a continuación se muestra una solicitud para enumerar todas las ejecuciones de flujos de trabajo que se iniciaron entre dos puntos específicos en el tiempo (mediante la anotación horaria de Unix).

```
{
  "domain": "867530901",
  "startTimeFilter":
  {
    "oldestDate": 1325376070,
    "latestDate": 1356998399
  },
  "tagFilter":
  {
    "tag": "music purchase"
  }
}
```

Ejemplo de solicitud y respuesta JSON de Amazon SWF

El siguiente ejemplo muestra una solicitud a Amazon SWF para una descripción del dominio que hemos creado previamente. A continuación se muestra la respuesta de Amazon SWF.

Solicitud HTTP POST

```
POST http://swf.us-east-1.amazonaws.com/ HTTP/1.1
Host: swf.us-east-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.25) Gecko/20111212
  Firefox/3.6.25 ( .NET CLR 3.5.30729; .NET4.0E)
Accept: application/json, text/javascript, */*
Accept-Language: en-us,en;q=0.5
```

```
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Content-Type: application/json; charset=UTF-8
X-Requested-With: XMLHttpRequest
X-Amz-Date: Sun, 15 Jan 2012 03:13:33 GMT
X-Amz-Target: com.amazonaws.swf.service.model.SimpleWorkflowService.DescribeDomain
Content-Encoding: amz-1.0
X-Amzn-Authorization: AWS3
  AWSSignature=AKIAIOSFODNN7EXAMPLE,Algorithm=HmacSHA256,SignedHeaders=Host;X-Amz-Date;X-Amz-Target;Content-Encoding,Signature=IFJtq3M366CHqM1TpyqYqd9z0ChCoKDC5SCJBsLifu4=
Referer: http://swf.us-east-1.amazonaws.com/explorer/index.html
Content-Length: 21
Pragma: no-cache
Cache-Control: no-cache

{"name": "867530901"}
```

Respuesta de Amazon SWF

```
HTTP/1.1 200 OK
Content-Length: 137
Content-Type: application/json
x-amzn-RequestId: e86a6779-3f26-11e1-9a27-0760db01a4a8

{"configuration":
  {"workflowExecutionRetentionPeriodInDays": "60"},
 "domainInfo":
  {"description": "music",
   "name": "867530901",
   "status": "REGISTERED"}
}
```

Observe que el protocolo (HTTP/1.1) va seguido de un código de estado (200). Un valor de código de 200 indica el éxito de la operación.

Amazon SWF no serializa valores “null”. Si su analizador de JSON está configurado para serializar valores “null” para las solicitudes, Amazon SWF no los tendrá en cuenta.

Cálculo de la firma HMAC-SHA para Amazon SWF

Cada solicitud enviada a Amazon SWF se debe autenticar. Firman AWS SDKs automáticamente sus solicitudes y administran su autenticación basada en tokens. Sin embargo, si desea escribir sus propias solicitudes HTTP POST, debe crear un valor `x-amzn-authorization` para el contenido HTTP POST Header como para de la autenticación de su solicitud.

Para obtener más información sobre cómo formatear los encabezados, consulte [Contenido de los encabezados HTTP](#). Para ver la AWS SDK para Java implementación de la firma de la AWS versión 3, consulta la [AWSSignerclase.java](#).

Creación de una firma de solicitud

Antes de crear una firma de solicitud HMAC-SHA, debe obtener sus credenciales de AWS (el ID de clave de acceso y la clave secreta).

Important

Puede usar cualquiera de las dos SHA1 opciones SHA256 para firmar sus solicitudes. Sin embargo, asegúrese de utilizar el mismo método para todo el proceso de firma. El método que elija tiene que corresponder al valor del nombre de `Algorithm` en el encabezado HTTP.

Para crear la firma de solicitud

1. Cree una forma canónica de los encabezados de solicitud HTTP. La forma canónica del encabezado de HTTP incluye lo siguiente:
 - host
 - Cualquier elemento del encabezado que empiece por `x-amz-`

Para obtener más información sobre los encabezados incluidos, consulte [Contenido de los encabezados HTTP](#).

- a. Para cada par de nombre-valor del encabezado, convierta el nombre del encabezado (pero no el valor del encabezado) en minúsculas.
- b. Cree un mapa que exprese la relación entre el nombre del encabezado y los valores del encabezado separados por comas.


```
x-amz-example: value1
x-amz-example: value2 => x-amz-example:value1,value2
```

Para obtener más información, consulte la [sección 4.2 de RFC 2616](#).

- c. Para cada par de nombre-valor del encabezado, convierta el par de nombre-valor en una cadena en el formato `headerName:headerValue`. Suprima los espacios en blanco del principio y del fin de `headerName` y `headerValue`, sin espacios a cada lado de los dos puntos.

```
x-amz-example1:value1,value2
x-amz-example2:value3
```

- d. Inserte una nueva línea (U+000A) después de cada cadena convertida, incluida la última cadena.
 - e. Ordene la colección de cadenas convertidas alfabéticamente, por nombre de encabezado.
2. Cree un string-to-sign valor que incluya los siguientes elementos:
 - Línea 1: el método HTTP (POST), seguido de una nueva línea.
 - Línea 2: el URI de la solicitud (/), seguido de una nueva línea.
 - Línea 3: una cadena vacía seguida de una nueva línea.

 Note

Normalmente, la cadena de consulta aparece aquí, pero Amazon SWF no utiliza ninguna cadena de consulta.

- Líneas 4–n: la cadena que representa los encabezados de la solicitud en formato canónico calculados en el Paso 1, seguida de una nueva línea. Esta nueva línea crea una línea en blanco entre los encabezados y el cuerpo de la solicitud HTTP. Para obtener más información, consulte [RFC 2616](#).
 - El cuerpo de la solicitud, no seguido de una nueva línea.
3. Calcule el SHA1 resumen SHA256 o el resumen del string-to-sign valor. Utilice el mismo método SHA en todo el proceso.
 4. Calcule y codifique en Base64 el HMAC-SHA utilizando un resumen SHA256 o un SHA1 resumen (según el método que haya utilizado) del valor resultante del paso anterior y la clave de

acceso secreta temporal del servicio de token de AWS seguridad mediante la acción de la API.

[GetSessionToken](#)

Note

Amazon SWF requiere un signo igual (=) al final del valor HMAC-SHA codificado en Base64. Si su rutina de codificación Base64 no incluye el signo de igual al final, inserte uno al final del valor.

Para obtener más información sobre el uso de credenciales de seguridad temporales con Amazon SWF y otros AWS servicios, consulte [AWS Servicios que funcionan con IAM en la Guía del usuario de IAM](#).

5. Coloque el valor resultante como el valor para el nombre Signature en el encabezado x-amzn-authorization de la solicitud HTTP enviada a Amazon SWF.
6. Amazon SWF comprueba la solicitud y realiza la operación especificada.

Lista de acciones de Amazon SWF por categoría

Esta sección muestra los temas de referencia de las acciones de Amazon SWF en la interfaz de programación de aplicaciones (API) de Amazon SWF. Se muestran por categoría funcional.

Para obtener una lista alfabética de las acciones, consulte la [referencia de la API de Amazon Simple Workflow Service](#).

Temas

- [Acciones relacionadas con actividades](#)
- [Acciones relacionadas con decisores](#)
- [Acciones relacionadas con ejecuciones de flujos de trabajo](#)
- [Acciones relacionadas con administración](#)
- [Acciones de visibilidad](#)

Acciones relacionadas con actividades

Los trabajadores de actividades utilizan `PollForActivityTask` para obtener nuevas tareas de actividad. En cuanto un proceso de trabajo recibe una tarea de actividad de Amazon SWF, realiza

la tarea y responde mediante `RespondActivityTaskCompleted` si ha podido llevarla la cabo o `RespondActivityTaskFailed` si no ha podido.

Estas son las acciones que realizan los procesos de trabajo de actividad.

- [PollForActivityTask](#)
- [RespondActivityTaskCompleted](#)
- [RespondActivityTaskFailed](#)
- [RespondActivityTaskCanceled](#)
- [RecordActivityTaskHeartbeat](#)

Acciones relacionadas con decisores

Los decisores utilizan `PollForDecisionTask` para obtener tareas de decisión. En cuanto un encargado de decisiones recibe una tarea de decisión de Amazon SWF, examina su historial de ejecución del flujo de trabajo y decide qué se debe hacer a continuación. Llama a `RespondDecisionTaskCompleted` para completar la tarea de decisión y proporciona cero o más decisiones siguientes.

Estas son las acciones que realizan los decisores.

- [PollForDecisionTask](#)
- [RespondDecisionTaskCompleted](#)

Acciones relacionadas con ejecuciones de flujos de trabajo

Estas son las acciones que se utilizan en una ejecución de flujo de trabajo.

- [RequestCancelWorkflowExecution](#)
- [StartWorkflowExecution](#)
- [SignalWorkflowExecution](#)
- [TerminateWorkflowExecution](#)

Acciones relacionadas con administración

La consola de Amazon SWF permite realizar tareas administrativas; sin embargo, puede utilizar las acciones de esta sección para automatizar funciones o crear sus propias herramientas administrativas.

Administración de actividades

- [RegisterActivityType](#)
- [DeprecateActivityType](#)
- [UndeprecateActivityType](#)
- [DeleteActivityType](#)

Administración de flujos de trabajo

- [RegisterWorkflowType](#)
- [DeprecateWorkflowType](#)
- [UndeprecateWorkflowType](#)
- [DeleteWorkflowType](#)

Administración de dominios

Estas acciones permiten registrar y descartar dominios de Amazon SWF.

- [RegisterDomain](#)
- [DeprecateDomain](#)
- [UndeprecateDomain](#)

Para obtener más información y ejemplos de estas acciones de administración de dominios, consulte [Cómo registrar un dominio con Amazon SWF](#).

Administración de ejecución de flujos de trabajo

- [RequestCancelWorkflowExecution](#)
- [TerminateWorkflowExecution](#)

Acciones de visibilidad

La consola de Amazon SWF permite realizar acciones de visibilidad; sin embargo, puede utilizar las acciones de esta sección para automatizar funciones o crear su propia consola o sus propias herramientas de administración.

Visibilidad de las actividades

- [ListActivityTypes](#)
- [DescribeActivityType](#)

Visibilidad de los flujos de trabajo

- [ListWorkflowTypes](#)
- [DescribeWorkflowType](#)

Visibilidad de la ejecución de flujos de trabajo

- [DescribeWorkflowExecution](#)
- [ListOpenWorkflowExecutions](#)
- [ListClosedWorkflowExecutions](#)
- [CountOpenWorkflowExecutions](#)
- [CountClosedWorkflowExecutions](#)
- [GetWorkflowExecutionHistory](#)

Visibilidad de dominios

- [ListDomains](#)
- [DescribeDomain](#)

Visibilidad de listas de tareas

- [CountPendingActivityTasks](#)
- [CountPendingDecisionTasks](#)

Cómo registrar un dominio con Amazon SWF

Sus tipos de actividad y flujo de trabajo y la ejecución del flujo de trabajo en sí se limitan a un dominio. Los dominios aíslan un conjunto de tipos, ejecuciones y listas de tareas de otros en la misma cuenta.

Puede registrar un dominio mediante la Consola de administración de AWS o mediante la `RegisterDomain` acción de la API de Amazon SWF. El siguiente ejemplo utiliza la API.

```
https://swf.us-east-1.amazonaws.com
RegisterDomain
{
  "name" : "867530901",
  "description" : "music",
  "workflowExecutionRetentionPeriodInDays" : "60"
}
```

Los parámetros se especifican en formato de notación de JavaScript objetos (JSON). Aquí, el periodo de retención se ha establecido en 60 días. Durante el período de retención, toda la información sobre la ejecución del flujo de trabajo está disponible mediante operaciones de visibilidad que utilizan la API Amazon SWF Consola de administración de AWS o la API de Amazon SWF.

Después de registrar el dominio, debe registrar el tipo de flujo de trabajo y los tipos de actividad utilizados por el flujo de trabajo. Tiene que registrar primero el dominio porque un nombre de dominio registrado forma parte de la información necesaria para registrar tipos de actividad y flujo de trabajo.

Véase también

[RegisterDomain](#) en la referencia de la API de Amazon Simple Workflow Service

Configuración de valores de tiempo de espera en Amazon SWF

Temas

- [Cuotas de los valores de tiempo de espera](#)
- [Tiempos de espera de las ejecuciones de flujos de trabajo y las tareas de decisión](#)
- [Tiempos de espera de las tareas de actividad](#)

- [Véase también](#)

Cuotas de los valores de tiempo de espera

Los valores de tiempo de espera siempre se declaran en segundos, y pueden abarcar cualquier número de segundos hasta un año (31 536 000 segundos), que es el límite máximo de ejecución para cualquier flujo de trabajo o actividad. El valor especial NONE se usa para configurar un parámetro de tiempo de espera "sin tiempo de espera", o infinito, pero sigue aplicándose la duración máxima de un año.

Tiempos de espera de las ejecuciones de flujos de trabajo y las tareas de decisión

Puede configurar valores de tiempos de espera para el flujo de trabajo y las tareas de decisión al registrar el tipo de flujo de trabajo. Por ejemplo:

```
https://swf.us-east-1.amazonaws.com
RegisterWorkflowType
{
  "domain": "867530901",
  "name": "customerOrderWorkflow",
  "version": "1.0",
  "description": "Handle customer orders",
  "defaultTaskStartToCloseTimeout": "600",
  "defaultExecutionStartToCloseTimeout": "3600",
  "defaultTaskList": { "name": "mainTaskList" },
  "defaultChildPolicy": "TERMINATE"
}
```

El registro del tipo de flujo de trabajo configura el [defaultTaskStartToCloseTimeout](#) a 600 segundos (10 minutos) y [defaultExecutionStartToCloseTimeout](#) a 3 600 segundos (1 hora).

Para obtener más información sobre el registro de los tipos de flujo de trabajo, consulte [Registro de un tipo de flujo de trabajo con Amazon SWF](#) y [RegisterWorkflowType](#) en la referencia de la API de Amazon Simple Workflow Service.

Para anular el valor configurado para `defaultExecutionStartToCloseTimeout`, especifique [executionStartToCloseTimeout](#) .

Tiempos de espera de las tareas de actividad

Puede configurar tiempos de espera para las tareas de actividad al registrar el tipo de actividad. Por ejemplo:

```
https://swf.us-east-1.amazonaws.com
RegisterActivityType
{
  "domain": "867530901",
  "name": "activityVerify",
  "version": "1.0",
  "description": "Verify the customer credit",
  "defaultTaskStartToCloseTimeout": "600",
  "defaultTaskHeartbeatTimeout": "120",
  "defaultTaskList": { "name": "mainTaskList" },
  "defaultTaskScheduleToStartTimeout": "1800",
  "defaultTaskScheduleToCloseTimeout": "5400"
}
```

Este registro de tipo de actividad configura el [defaultTaskStartToCloseTimeout](#) a 600 segundos (10 minutos), el [defaultTaskHeartbeatTimeout](#) a 120 segundos (2 minutos), el [defaultTaskScheduleToStartTimeout](#) a 1 800 segundos (30 minutos) y el [defaultTaskScheduleToCloseTimeout](#) a 5 400 segundos (1,5 horas).

Para obtener más información sobre el registro de los tipos de actividad, consulte [Registro de un tipo de actividad con Amazon SWF](#) y [RegisterActivityType](#) en la referencia de la API de Amazon Simple Workflow Service.

Para anular el valor configurado para defaultTaskStartToCloseTimeout, especifique [taskStartToCloseTimeout](#) al programar la tarea de actividad.

Véase también

[Tipos de tiempo de espera de Amazon SWF](#)

Registro de un tipo de flujo de trabajo con Amazon SWF

En el ejemplo de esta sección se registra un tipo de flujo de trabajo con la API de Amazon SWF. El nombre de la versión que especifique durante el registro forma un identificador único para el tipo de flujo de trabajo. El dominio especificado tiene que haber sido registrado con la acción de la API [RegisterDomain](#).

Los parámetros de tiempo de espera del siguiente ejemplo son los valores de duración especificados en segundos. Para el parámetro `defaultTaskStartToCloseTimeout`, puede usar el especificador de duración `NONE` para indicar que no hay tiempo de espera. Sin embargo, no puede especificar un valor de `NONE` para `defaultExecutionStartToCloseTimeout`; hay un límite máximo de un año en el que puede ejecutarse una ejecución de flujo de trabajo. Superar este límite siempre provoca que se agote el tiempo de espera de la ejecución del flujo de trabajo. Si especifica un valor superior a un año para `defaultExecutionStartToCloseTimeout`, el registro producirá un error.

```
https://swf.us-east-1.amazonaws.com
RegisterWorkflowType
{
  "domain" : "867530901",
  "name" : "customerOrderWorkflow",
  "version" : "1.0",
  "description" : "Handle customer orders",
  "defaultTaskStartToCloseTimeout" : "600",
  "defaultExecutionStartToCloseTimeout" : "3600",
  "defaultTaskList" : { "name": "mainTaskList" },
  "defaultChildPolicy" : "TERMINATE"
}
```

Véase también

[RegisterWorkflowType](#) en la referencia de la API de Amazon Simple Workflow Service

Registro de un tipo de actividad con Amazon SWF

En el siguiente ejemplo, se registra un tipo de actividad mediante la API de Amazon SWF. El nombre de la versión que especifique durante el registro forma un identificador único para el tipo de actividad dentro del dominio. El dominio especificado tiene que haberse registrado ya mediante la acción `RegisterDomain`.

Los parámetros de tiempo de espera de este ejemplo son los valores de duración especificados en segundos. Puede usar el especificador de duración `NONE` para indicar que no hay tiempo de espera.

```
https://swf.us-east-1.amazonaws.com
RegisterActivityType
{
```

```
"domain" : "867530901",
"name" : "activityVerify",
"version" : "1.0",
"description" : "Verify the customer credit",
"defaultTaskStartToCloseTimeout" : "600",
"defaultTaskHeartbeatTimeout" : "120",
"defaultTaskList" : { "name" : "mainTaskList" },
"defaultTaskScheduleToStartTimeout" : "1800",
"defaultTaskScheduleToCloseTimeout" : "5400"
}
```

Véase también

[RegisterActivityType](#) en la referencia de la API de Amazon Simple Workflow Service

AWS Lambda tareas en Amazon SWF

Temas

- [Acerca AWS Lambda](#)
- [Beneficios y limitaciones de la utilización de tareas de Lambda](#)
- [Uso de tareas de Lambda en los flujos de trabajo](#)

Acerca AWS Lambda

AWS Lambda es un servicio informático totalmente gestionado que ejecuta el código en respuesta a eventos generados por un código personalizado o desde varios AWS servicios, como Amazon S3, DynamoDB, Amazon Kinesis, Amazon SNS y Amazon Cognito. Para obtener más información acerca de Lambda, consulte la [Guía para desarrolladores de AWS Lambda](#).

Amazon Simple Workflow Service proporciona una tarea de Lambda que permite ejecutar funciones de Lambda en lugar de actividades de Amazon SWF tradicionales, o a la vez que ellas.

Important

Se cobrarán a su AWS cuenta las ejecuciones (solicitudes) de Lambda ejecutadas por Amazon SWF en su nombre. [Para obtener más información sobre los precios de Lambda, consulte <https://aws.amazon.com/lambda/pricing/>.](#)

Beneficios y limitaciones de la utilización de tareas de Lambda

Hay una serie de beneficios derivados de la utilización de tareas de Lambda en lugar de una actividad de Amazon SWF tradicional:

- No hay necesidad de registrar ni crear versiones de tareas de Lambda a diferencia de los tipos de actividad de Amazon SWF.
- Puede utilizar cualquier función de Lambda que ya haya definido en sus flujos de trabajo.
- Amazon SWF llama directamente a las funciones de Lambda; no hay necesidad de implementar ningún proceso de trabajo para ejecutarlas, como ocurre con las actividades tradicionales.
- Lambda proporciona métricas y registros para el seguimiento y el análisis de las ejecuciones de las funciones.

También hay una serie de limitaciones relativas a las tareas Lambda que debe tener presente:

- Las tareas de Lambda solo se pueden ejecutar en AWS regiones que admiten Lambda. Para obtener más información sobre las regiones compatibles actualmente con Lambda, consulte [Regiones y puntos de conexión de Lambda](#) en la Referencia general de Amazon Web Services.
- Actualmente, las tareas Lambda solo se admiten en la API HTTP básica de SWF y en la AWS Flow Framework API para Java. Actualmente, no hay soporte para tareas Lambda en Ruby AWS Flow Framework .

Uso de tareas de Lambda en los flujos de trabajo

Para utilizar tareas de Lambda en los flujos de trabajo de Amazon SWF, tendrá que:

1. Configurar roles de IAM para proporcionar a Amazon SWF permiso para invocar funciones de Lambda.
2. Asociar roles de IAM a los flujos de trabajo.
3. Llamar a la función de Lambda durante la ejecución de un flujo de trabajo.

Configuración de un rol de IAM

Antes de poder invocar funciones de Lambda desde Amazon SWF, debe proporcionar un rol de IAM que permita tener acceso a Lambda desde Amazon SWF. Puede:

- elija un rol predefinido, Rol, AWSLambda para dar permiso a sus flujos de trabajo para invocar cualquier función de Lambda asociada a su cuenta.
- defina su propia política y el rol asociado para conceder a los flujos de trabajo permiso para invocar determinadas funciones de Lambda, especificadas por sus nombres ARNs de recursos de Amazon ().

Limitar permisos de un rol de IAM

Puede limitar los permisos de un rol de IAM que proporcione a Amazon SWF mediante las claves de contexto `SourceArn` y `SourceAccount` en su política de confianza para los recursos. Estas claves limitan el uso de una política de IAM para que solo se utilice en las ejecuciones de Amazon Simple Workflow Service que pertenezcan al ARN del dominio especificado. Si se utilizan ambas claves de contexto de condición global, tanto el valor `aws:SourceAccount` como la cuenta a la que se hace referencia en el valor `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utilicen en la misma instrucción de política.

En el siguiente ejemplo, la clave de `SourceArn` contexto restringe el uso de la función de servicio de IAM únicamente en las ejecuciones de Amazon Simple Workflow Service que pertenezcan a `someDomain` la cuenta, `123456789012`

- Declaración 1

Director: "Service": "swf.amazonaws.com"

Acción: sts:AssumeRole

```
"Condition": {  
  "ArnLike": {  
    "aws:SourceArn": "arn:aws:swf:*:123456789012:/domain/someDomain"  
  }  
}
```

En el siguiente ejemplo, la clave de `SourceAccount` contexto restringe el uso de la función de servicio de IAM únicamente en las ejecuciones de Amazon Simple Workflow Service en la cuenta, `123456789012`

```
"Condition": {
```

```
"StringLike": {  
  "aws:SourceAccount": "123456789012"  
}  
}
```

Proporcionar acceso a Amazon SWF para invocar cualquier rol de Lambda

Puede usar el rol predefinido, Role, para dar a sus flujos de trabajo de Amazon SWF la posibilidad de invocar cualquier AWSLambda función de Lambda asociada a su cuenta.

Para usar AWSLambda Role para dar acceso a Amazon SWF a la hora de invocar funciones de Lambda

1. Abra la [consola de Amazon IAM](#).
2. Elija Roles, y a continuación Create New Role.
3. Dé un nombre al rol, como swf-lambda y elija Next Step.
4. En Roles de servicio de AWS , seleccione Amazon SWF, y, a continuación, seleccione Paso siguiente.
5. En la pantalla Adjuntar política, seleccione AWSLambdaRol de la lista.
6. Elija Next Step y a continuación Create Role una vez que haya revisado el rol.

Definición de un rol de IAM para proporcionar acceso para invocar una función de Lambda concreta

Si desea proporcionar acceso para invocar una función de Lambda concreta desde un flujo de trabajo, tendrá que definir su propia política de IAM.

Cómo crear una política de IAM para dar acceso a una función de Lambda concreta

1. Abra la [consola de Amazon IAM](#).
2. Elija Políticas, y a continuación Crear política.
3. Elija Copiar una política AWS gestionada y seleccione AWSLambdaRol en la lista. Se generará una política para usted. Tiene la opción de editar su nombre y descripción según sus necesidades.
4. En el campo Recurso del Documento de política, añada el ARN de las funciones de Lambda.
Por ejemplo:

- Recurso: arn:aws:lambda:us-east-1:111122223333:function:hello_lambda_function

Note

Para obtener una descripción completa de los procedimientos que se deben seguir para especificar recursos en un rol de IAM, consulte la [información general sobre las políticas de IAM](#) en la Uso de IAM.

5. Elija Create Policy para finalizar la creación de su política.

A continuación, puede seleccionar esta política al crear un nuevo rol de IAM, y utilizar ese rol para dar acceso para invocar los flujos de trabajo de Amazon SWF. Este procedimiento es muy similar a crear un rol con la política de AWSLambdaRole. En su lugar, elija su propia política al crear el rol.

Cómo crear un rol de Amazon SWF mediante su política de Lambda

1. Abra la [consola de Amazon IAM](#).
2. Elija Roles, y a continuación Create New Role.
3. Dé un nombre al rol, como `swf-lambda-function` y elija Next Step.
4. En Roles de servicio de AWS, seleccione Amazon SWF, y, a continuación, seleccione Paso siguiente.
5. En la pantalla Asociar política, seleccione de la lista la política específica de la función de Lambda.
6. Elija Next Step y a continuación Create Role una vez que haya revisado el rol.

Asociar el rol de IAM al flujo de trabajo

Una vez que haya definido el rol de IAM, tendrá que asociarlo al flujo de trabajo que se utilizará para llamar a las funciones de Lambda cuyo acceso proporcionó a Amazon SWF.

Hay dos lugares donde puede asociar la función a su flujo de trabajo:

- Durante el registro del tipo de flujo de trabajo. Esta función puede usarse entonces como la función Lambda predeterminada para todas las ejecuciones de ese tipo de flujo de trabajo.
- Al iniciar una ejecución de flujo de trabajo. Esta función se usará solo durante la ejecución de este flujo de trabajo (y en toda la ejecución).

A fin de proporcionar una función Lambda predeterminada para un tipo de flujo de trabajo

- Al llamar `RegisterWorkflowType`, establezca el `defaultLambdaRole` campo en el ARN del rol que definió.

Para proporcionar una función Lambda que se va a usar durante una ejecución de flujo de trabajo

- Al llamar `StartWorkflowExecution`, establezca el campo `LambdaRole` en el ARN del rol que definió.

Note

si la cuenta que llama `RegisterWorkflowType` o `StartWorkflowExecution` no tiene permiso para usar el rol en cuestión, la llamada fallará con un `OperationNotPermittedFault`

Llamar a la función de Lambda desde un flujo de trabajo de Amazon SWF

Puede usar el tipo de `ScheduleLambdaFunctionDecisionAttributes` datos para identificar la función Lambda a la que se va a llamar durante la ejecución de un flujo de trabajo.

Durante una llamada a `RespondDecisionTaskCompleted`, proporcione una lista `ScheduleLambdaFunctionDecisionAttributes` de sus decisiones. Por ejemplo:

```
{
  "decisions": [{
    "ScheduleLambdaFunctionDecisionAttributes": {
      "id": "lambdaTaskId",
      "name": "myLambdaFunctionName",
      "input": "inputToLambdaFunction",
      "startToCloseTimeout": "30"
    },
  ]
}
```

Establezca los siguientes parámetros:

- `id` con un identificador para la tarea de Lambda. Esta debe ser una cadena de 1 a 256 caracteres y no debe contener los caracteres: (dos puntos), / (barra inclinada), | (barra vertical). Tampoco debe contener ningún carácter de control (`\u0000` - `\u001f` y `\u007f` - `\u009f`) ni la cadena literal `a\r\n`.
- `name` con el nombre de la función de Lambda. El flujo de trabajo de Amazon SWF debe proporcionarse con un rol de IAM que le conceda acceso para llamar a la función de Lambda. El nombre proporcionado debe seguir las restricciones del `FunctionName` parámetro, como en la acción `Lambda Invoke`.
- `input` con datos de entrada opcionales para la función. Si se establece, debe seguir las restricciones del `ClientContext` parámetro, como en la acción `Lambda Invoke`.
- `startToCloseTiempo` de espera con un período máximo opcional, en segundos, que la función puede tardar en ejecutarse antes de que la tarea falle (con una excepción de tiempo de espera). El valor `NONE` se puede usar para especificar una duración ilimitada.

[Para obtener más información, consulte Implementación de tareas AWS Lambda](#)

Desarrollo de un proceso de trabajo de actividad en Amazon SWF

Un proceso de trabajo de actividad que proporcione la implementación de uno o varios tipos de actividad. Un proceso de trabajo de actividad se comunica con Amazon SWF para recibir tareas de actividad y realizarlas. Puede tener una flota de varios procesos de trabajo de actividad que realicen tareas de actividad del mismo tipo de actividad.

Amazon SWF pone una tarea de actividad a disposición de los procesos de trabajo de actividad cuando el decisor programa la tarea de actividad. Cuando un decisor programa una tarea de actividad, proporciona los datos (que usted determine) necesarios para que el proceso de trabajo de actividad realice la tarea de actividad. Amazon SWF inserta estos datos en la tarea de actividad antes de enviarla al proceso de trabajo de la actividad.

Usted administra los procesos de trabajo de actividad. Se pueden escribir en cualquier lenguaje. Un proceso de trabajo se puede ejecutar en cualquier lugar, siempre que pueda comunicarse con Amazon SWF a través de la API. Como Amazon SWF proporciona toda la información necesaria para realizar una tarea de actividad, todos los procesos de trabajo de actividad pueden carecer de estado. La ausencia de estado permite que sus flujos de trabajo sean altamente escalables; para afrontar los requisitos de capacidad incrementada, solo tiene que añadir más procesos de trabajo de actividad.

En esta sección se explica cómo implementar un proceso de trabajo de actividad. Los procesos de trabajo de actividad deben hacer lo siguiente una y otra vez.

1. Sondear Amazon SWF para encontrar una tarea de actividad.
2. Empezar a realizar la tarea.
3. Enviar un latido a Amazon SWF de forma periódica si la duración de la tarea es prolongada.
4. Informar de que la tarea se ha completado o de que se ha producido un error en la misma y devolver los resultados a Amazon SWF.

Temas

- [Sondeo de tareas de actividad](#)
- [Realización de la tarea de actividad](#)
- [Notificación de los latidos de tareas de actividad](#)
- [Finalización de una tarea de actividad o generación de un error en la misma](#)
- [Lanzamiento de procesos de trabajo de actividad](#)

Sondeo de tareas de actividad

Para realizar tareas de actividad, cada proceso de trabajo de actividad debe sondear Amazon SWF llamando periódicamente a la acción `PollForActivityTask`.

En el siguiente ejemplo, el proceso de trabajo de actividad `ChargeCreditCardWorker01` sondea una tarea de la lista de tareas, `ChargeCreditCard-v0.1`. Si no hay ninguna tarea de actividad disponible, transcurridos 60 segundos, Amazon SWF devuelve una respuesta vacía. Una respuesta vacía es una estructura `Task` en la que el valor de `taskToken` es una cadena vacía.

```
https://swf.us-east-1.amazonaws.com
PollForActivityTask
{
  "domain" : "867530901",
  "taskList" : { "name": "ChargeCreditCard-v0.1" },
  "identity" : "ChargeCreditCardWorker01"
}
```

Si una tarea de actividad está disponible, Amazon SWF la devuelve al proceso de trabajo de actividad. La tarea contiene los datos que especifica el decisor al programar la actividad.

Una vez que un proceso de trabajo de actividad reciba una tarea de actividad, estará listo para realizar el trabajo. En la siguiente sección se proporciona información acerca de la realización de una tarea de actividad.

Realización de la tarea de actividad

Tras recibir una tarea de actividad, el proceso de trabajo de actividad estará listo para realizarla.

Para realizar una tarea de actividad

1. Programe su proceso de trabajo de actividad para interpretar el contenido en el campo de entrada de la tarea. En este campo se incluyen los datos especificados por el decisor durante la programación de la tarea.
2. Programe el proceso de trabajo de actividad para empezar a procesar los datos y ejecutar su lógica.

En la siguiente sección, se describe cómo programar procesos de trabajo de actividad para proporcionar actualizaciones de estado a Amazon SWF para actividades de ejecución prolongada.

Notificación de los latidos de tareas de actividad

Si se ha registrado el tiempo de espera del latido con el tipo de actividad, el proceso de trabajo de actividad debe registrar un latido antes de superarse el tiempo de espera del latido. Si una tarea de actividad no proporciona un latido en el intervalo de tiempo de espera, el tiempo de espera de la tarea se agota, Amazon SWF la cierra y programa una nueva tarea de decisión para informar a un decisor de que se ha agotado el tiempo de espera. Es entonces cuando el decisor puede volver a programar la tarea de actividad o tomar otras medidas.

Si, tras agotarse el tiempo de espera, el proceso de trabajo de actividad intentara contactar con Amazon SWF, por ejemplo, mediante una llamada a `RespondActivityTaskCompleted`, Amazon SWF devolverá un error `UnknownResource`.

En esta sección se describe cómo proporcionar un latido de actividades.

Para registrar un latido de tareas de actividad, programe su proceso de trabajo de actividad para llamar a la acción `RecordActivityTaskHeartbeat`. Esta acción también proporciona un campo de cadena que puede usar para almacenar datos de formato libre para cuantificar el progreso de la forma que funcione para su aplicación.

En este ejemplo, el proceso de trabajo de actividad informa del latido a Amazon SWF y utiliza el campo de detalles para indicar que la tarea de actividad se ha completado en un 40 por ciento. Para informar del latido, el proceso de trabajo de actividad debe especificar el token de tarea de la tarea de actividad.

```
https://swf.us-east-1.amazonaws.com
RecordActivityTaskHeartbeat
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "details" : "40"
}
```

Esta acción en sí misma no crea un evento en el historial de ejecución de flujo de trabajo; sin embargo, si el tiempo de espera de la tarea se agota, el historial de ejecución de flujo de trabajo contendrá un evento `ActivityTaskTimedOut` que incluye la información del último latido generado por el proceso de trabajo de actividad.

Finalización de una tarea de actividad o generación de un error en la misma

Después de ejecutar una tarea, el proceso de trabajo de actividad debe informar de si la tarea de actividad se ha completado o de si se ha producido un error en la misma.

Finalización de una tarea de actividad

Para completar una tarea de actividad, programe el proceso de trabajo de actividad para llamar a la acción `RespondActivityTaskCompleted` tras completar correctamente una tarea de actividad, especificando el token de tarea.

En este ejemplo, el proceso de trabajo de actividad indica que la tarea se ha completado correctamente.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "results": "40"
}
```

Al completarse la actividad, Amazon SWF programa una tarea de decisión nueva para la ejecución de flujo de trabajo a la que se asocia la actividad.

Programa el proceso de trabajo de actividad para sondear otra tarea de actividad después de que haya completado la tarea en cuestión. De este modo se crea un bucle en el que el proceso de trabajo de actividad sondea y completa tareas continuamente.

Si la actividad no responde dentro del `StartToCloseTimeout` período o si se `ScheduleToCloseTimeout` ha superado, Amazon SWF agota el tiempo de espera de la tarea de actividad y programa una tarea de decisión. Esto permite a un decisor tomar las medidas adecuadas, por ejemplo, reprogramar la tarea.

Por ejemplo, si una EC2 instancia de Amazon está ejecutando una tarea de actividad y la instancia falla antes de que se complete la tarea, el decisor recibe un evento de tiempo de espera en el historial de ejecución del flujo de trabajo. Si la tarea de actividad utiliza un latido, el decisor recibe el evento cuando la tarea no puede entregar el siguiente latido después de que la EC2 instancia de Amazon falle. De lo contrario, el decisor recibe finalmente el evento cuando la tarea de actividad no logra completarse antes de llegar a uno de sus valores de tiempo de espera generales. Depende entonces del decisor reasignar la tarea o tomar otras medidas.

Generación de un error en una tarea de actividad

Si un proceso de trabajo de actividad no puede realizar una tarea de actividad por algún motivo, pero todavía puede comunicarse con Amazon SWF, es posible programarlo para que produzca un error en la tarea.

Para programar un proceso de trabajo de actividad de modo que produzca un error en una tarea de actividad, programe el proceso de trabajo de actividad para llamar a la acción `RespondActivityTaskFailed` que especifica el token de tarea de la tarea.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskFailed
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "reason" : "CC-Invalid",
  "details" : "Credit Card Number Checksum Failed"
}
```

Como desarrollador, se ocupa de definir los valores que se almacenan en los campos de detalles o motivo. Estas son cadenas de formato libre; puede utilizar cualquier convención de código de error que sirva a su aplicación. Amazon SWF no procesa estos valores. Sin embargo, Amazon SWF puede mostrar estos valores en la consola.

Cuando se produce un error en una tarea de actividad, Amazon SWF programa una tarea de decisión para la ejecución de flujo de trabajo a la que está asociada la tarea para informar al decisor del error. Programe su decisor para controlar las actividades que han producido un error, por ejemplo, reprogramando la actividad o no superando la ejecución de flujo de trabajo, dependiendo de la naturaleza del error.

Lanzamiento de procesos de trabajo de actividad

Para lanzar procesos de trabajo de actividad, empaquete su lógica en un ejecutable que puede usar en su plataforma de proceso de trabajo de actividad. Por ejemplo, puede empaquetar su código de actividad como ejecutable de Java que puede ejecutar tanto en servidores Linux como en servidores Windows.

Tras su lanzamiento, sus procesos de trabajo comienzan a sondear tareas. Hasta que el decisor programa tareas de actividad, sin embargo, el tiempo de espera de estos sondeos se agota sin tareas y sus procesos de trabajo se limitan a seguir realizando sondeos.

Puesto que los sondeos son solicitudes salientes, el proceso de trabajo de la actividad se puede ejecutar en cualquier red con acceso al punto de conexión de Amazon SWF.

Puede lanzar tantos procesos de trabajo de actividad como desee. A medida que el decisor programa tareas de actividad, Amazon SWF las distribuye automáticamente a los procesos de trabajo de actividad de sondeo.

Desarrollo de decisores en Amazon SWF

Un decisor es una implementación de la lógica de coordinación de su tipo de flujo de trabajo que se ejecuta durante la ejecución de su flujo de trabajo. Puede ejecutar diversos decisores para un solo tipo de flujo de trabajo.

Como el estado de ejecución de una ejecución de flujo de trabajo se almacena en su historial de flujo de trabajo, los decisores pueden carecer de estado. Amazon SWF mantiene el historial de ejecución del flujo de trabajo y se lo proporciona al decisor con cada tarea de decisión. Esto le permite añadir y quitar decisores de forma dinámica, en caso necesario, lo que hace que el procesamiento de sus flujos de trabajo sea muy escalable. A medida que crece la carga en su sistema, simplemente añada más decisores para gestionar la mayor capacidad. Sin embargo, tenga en cuenta que solo puede haber una tarea de decisión abierta en un momento dado para la ejecución de un determinado flujo de trabajo.

Cada vez que se produce un cambio de estado en una ejecución de flujo de trabajo, Amazon SWF programa una tarea de decisión. Cada vez que un decisor recibe una tarea de decisión, efectúa las siguientes operaciones:

- Interpreta el historial de ejecución del flujo de trabajo proporcionado con la tarea de decisión.
- Aplica la lógica de coordinación basada en el historial de ejecución del flujo de trabajo y toma decisiones sobre qué hacer a continuación. Cada decisión se representa por una estructura de decisión.
- Completa la tarea de decisión y proporciona una lista de decisiones a Amazon SWF.

En esta sección se describe cómo desarrollar un decisor, lo que implica:

- Programar el decisor para que realice sondeos para obtener tareas de decisión
- Programar el decisor para interpretar el historial de ejecución del flujo de trabajo y tomar decisiones
- Programar el decisor para que responda a una tarea de decisión

Los ejemplos de esta sección muestran cómo puede programar un decisor para un flujo de trabajo de e-commerce.

Puede implementar el decisor en cualquier lenguaje de su elección y ejecutarlo en cualquier lugar, siempre que pueda comunicarse con Amazon SWF a través de su API de servicio.

Temas

- [Definición de la lógica de coordinación](#)
- [Sondeo de tareas de decisión](#)
- [Aplicación de la lógica de coordinación](#)
- [Respuesta con decisiones](#)
- [Cierre de una ejecución de flujo de trabajo](#)
- [Lanzamiento de decisores](#)

Definición de la lógica de coordinación

Lo primero que debe hacer al desarrollar un decisor es definir la lógica de coordinación. En el ejemplo de e-commerce, la lógica de coordinación que programa cada actividad después de que se complete la actividad anterior se parecerá a lo siguiente:

```
IF lastEvent = "StartWorkflowInstance"
  addToDecisions ScheduleVerifyOrderActivity

ELSIF lastEvent = "CompleteVerifyOrderActivity"
  addToDecisions ScheduleChargeCreditCardActivity

ELSIF lastEvent = "CompleteChargeCreditCardActivity"
  addToDecisions ScheduleCompleteShipOrderActivity

ELSIF lastEvent = "CompleteShipOrderActivity"
  addToDecisions ScheduleRecordOrderCompletion

ELSIF lastEvent = "CompleteRecordOrderCompletion"
  addToDecisions CloseWorkflow

ENDIF
```

El decisor aplica la lógica de coordinación al historial de ejecución del flujo de trabajo y crea una lista de decisiones cuando completa la tarea de decisión mediante la acción `RespondDecisionTaskCompleted`.

Sondeo de tareas de decisión

Cada decisor realiza un sondeo para comprobar si hay tareas de decisión. Las tareas de decisión contienen la información que el decisor usa para generar decisiones como la programación de las tareas de actividad. Para realizar un sondeo para comprobar si hay tareas de decisión, el decisor utiliza la acción `PollForDecisionTask`.

En este ejemplo, el decisor realiza un sondeo para comprobar si hay una tarea de decisión, especificando la lista de tareas `customerOrderWorkflow-0.1`.

```
https://swf.us-east-1.amazonaws.com
PollForDecisionTask
{
  "domain": "867530901",
```

```
"taskList": {"name": "customerOrderWorkflow-v0.1"},
"identity": "Decider01",
"maximumPageSize": 50,
"reverseOrder": true
}
```

Si una tarea de decisión está disponible en la lista de tareas especificada, Amazon SWF la devolverá de inmediato. Si no hay ninguna tarea de decisión disponible, Amazon SWF mantendrá abierta la conexión durante un máximo de 60 segundos y devolverá una tarea en cuanto haya alguna disponible. Si no hay tareas disponibles, Amazon SWF devolverá una respuesta vacía. Una respuesta vacía es una estructura Task en la que el valor de taskToken es una cadena vacía. Asegúrese de programar el decisor para que sondee si hay otra tarea si recibe una respuesta vacía.

Si hay una tarea de decisión disponible, Amazon SWF devolverá una respuesta que contenga la tarea de decisión, así como una vista paginada del historial de ejecución del flujo de trabajo.

En este ejemplo, el tipo del evento más reciente indica que la ejecución del flujo de trabajo ha comenzado y el elemento de entrada contiene la información necesaria para realizar la primera tarea.

```
{
  "events": [
    {
      "decisionTaskStartedEventAttributes": {
        "identity": "Decider01",
        "scheduledEventId": 2
      },
      "eventId": 3,
      "eventTimestamp": 1326593394.566,
      "eventType": "DecisionTaskStarted"
    }, {
      "decisionTaskScheduledEventAttributes": {
        "startToCloseTimeout": "600",
        "taskList": { "name": "specialTaskList" }
      },
      "eventId": 2,
      "eventTimestamp": 1326592619.474,
      "eventType": "DecisionTaskScheduled"
    }, {
      "eventId": 1,
      "eventTimestamp": 1326592619.474,
      "eventType": "WorkflowExecutionStarted",
      "workflowExecutionStartedEventAttributes": {
```

```
    "childPolicy" : "TERMINATE",
    "executionStartToCloseTimeout" : "3600",
    "input" : "data-used-decider-for-first-task",
    "parentInitiatedEventId": 0,
    "tagList" : ["music purchase", "digital", "ricoh-the-dog"],
    "taskList": { "name": "specialTaskList" },
    "taskStartToCloseTimeout": "600",
    "workflowType": {
      "name": "customerOrderWorkflow",
      "version": "1.0"
    }
  }
}
],
...
}
```

Después de recibir el historial de ejecución del flujo de trabajo, el decisor interpreta el historial y toma decisiones basadas en su lógica de coordinación.

Como el número de eventos del historial del flujo de trabajo para una sola ejecución de flujo de trabajo puede ser elevado, el resultado devuelto podría repartirse en una serie de páginas. Para recuperar las páginas siguientes, realice llamadas adicionales a `PollForDecisionTask` las `nextPageToken` devueltas por la llamada inicial. Tenga en cuenta que no llama `GetWorkflowExecutionHistory` con este `nextPageToken`. En cambio, llame de nuevo a `PollForDecisionTask`.

Aplicación de la lógica de coordinación

Después de que el decisor reciba una tarea de decisión, prográmelo para que interprete el historial de ejecución del flujo de trabajo para determinar qué eventos han tenido lugar hasta el momento. Según esta información, debe generar una lista de decisiones.

En el ejemplo de e-commerce, solo nos interesa el último evento del historial del flujo de trabajo para definir la siguiente lógica.

```
IF lastEvent = "StartWorkflowInstance"
  addToDecisions ScheduleVerifyOrderActivity

ELSIF lastEvent = "CompleteVerifyOrderActivity"
  addToDecisions ScheduleChargeCreditCardActivity
```

```
ELSIF lastEvent = "CompleteChargeCreditCardActivity"  
  addToDecisions ScheduleCompleteShipOrderActivity  
  
ELSIF lastEvent = "CompleteShipOrderActivity"  
  addToDecisions ScheduleRecordOrderCompletion  
  
ELSIF lastEvent = "CompleteRecordOrderCompletion"  
  addToDecisions CloseWorkflow  
  
ENDIF
```

Si el último evento es `CompleteVerifyOrderActivity`, debe añadir la actividad `ScheduleChargeCreditCardActivity` a la lista de decisiones.

Después de que el decisor determine las decisiones que se deban tomar, puede responder a Amazon SWF con las decisiones apropiadas.

Respuesta con decisiones

Después de interpretar el historial de ejecución del flujo de trabajo y generar una lista de decisiones, el decisor estará listo para responder a Amazon SWF con esas decisiones.

Programa el decisor para que extraiga los datos que necesita del historial de ejecución del flujo de trabajo, después cree las decisiones que especifican las siguientes acciones apropiadas para el flujo de trabajo. El decisor transmite esta decisión de vuelta a Amazon SWF mediante la acción `RespondDecisionTaskCompleted`. Consulte la referencia de la API de Amazon Simple Workflow Service para ver una lista de los [tipos de decisiones](#) disponibles.

En el ejemplo de e-commerce, cuando el decisor responde con el conjunto de decisiones que ha generado, también incluye la entrada de la tarjeta de crédito del historial de ejecución del flujo de trabajo. Entonces, el trabajo de actividad tiene toda la información que necesita para realizar la tarea de actividad.

Cuando se han completado todas las actividades de la ejecución del flujo de trabajo, el decisor cierra la ejecución del flujo de trabajo.

```
https://swf.us-east-1.amazonaws.com  
RespondDecisionTaskCompleted  
{  
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",  
  "decisions" : [  
    {
```

```
"decisionType" : "ScheduleActivityTask",
"scheduleActivityTaskDecisionAttributes" : {
  "control" : "OPTIONAL_DATA_FOR_DECIDER",
  "activityType" : {
    "name" : "ScheduleChargeCreditCardActivity",
    "version" : "1.1"
  },
  "activityId" : "3e2e6e55-e7c4-beef-feed-aa815722b7be",
  "scheduleToCloseTimeout" : "360",
  "taskList" : { "name" : "CC_TASKS" },
  "scheduleToStartTimeout" : "60",
  "startToCloseTimeout" : "300",
  "heartbeatTimeout" : "60",
  "input" : "4321-0001-0002-1234: 0212 : 234"
}
}
]
```

Cierre de una ejecución de flujo de trabajo

Cuando el decisor determina que el proceso comercial se ha completado, esto es, ya no hay más actividades que realizar, el decisor genera una decisión para cerrar la ejecución del flujo de trabajo.

Para cerrar una ejecución de flujo de trabajo, programe el decisor para que interprete los eventos del historial de ejecución del flujo de trabajo para determinar qué ha tenido lugar en la ejecución hasta el momento y ver si la ejecución de flujo de trabajo se debe cerrar.

Si el flujo de trabajo se ha completado satisfactoriamente, cierre la ejecución del flujo de trabajo llamando a `RespondDecisionTaskCompleted` con la decisión `CompleteWorkflowExecution`. Alternativamente, puede detener una ejecución errónea con la decisión `FailWorkflowExecution`.

En el ejemplo de e-commerce, el decisor revisa el historial de ejecución del flujo de trabajo y, basándose en la lógica de coordinación, añade a su lista de decisiones una decisión de cerrar la ejecución del flujo de trabajo, e inicia una acción `RespondDecisionTaskCompleted` con una decisión de cerrar el flujo de trabajo.

Note

Hay algunos casos en los que el cierre de una ejecución de flujo de trabajo no tiene éxito. Por ejemplo, si se recibe una señal mientras el decisor está cerrando la ejecución del flujo de trabajo, la decisión de cierre producirá un error. Para gestionar esta posibilidad, asegúrese

de que el decisor continúa realizando sondeos para obtener tareas de decisión. Además, asegúrese de que el decisor que recibe la siguiente tarea de decisión responda al evento (en este caso, a una señal) que impidió el cierre de la ejecución.

También puede cancelar las ejecuciones de flujos de trabajo. Esto puede ser muy útil en flujos de trabajo de larga duración. Para permitir la cancelación, el decisor debe gestionar el evento `WorkflowExecutionCancelRequested` en el historial. Este evento indica que se ha solicitado la cancelación de la ejecución. El decisor debe realizar las acciones de limpieza adecuadas, como cancelar las tareas de actividad en curso, y cerrar el flujo de trabajo llamando a la acción `RespondDecisionTaskCompleted` con la decisión `CancelWorkflowExecution`.

En el ejemplo siguiente, llame a `RespondDecisionTaskCompleted` para especificar que se ha cancelado la ejecución del flujo de trabajo en curso.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions" : [
    {
      "decisionType": "CancelWorkflowExecution",
      "CancelWorkflowExecutionAttributes": {
        "Details": "Customer canceled order"
      }
    }
  ]
}
```

Amazon SWF se asegura de que la decisión de cerrar o cancelar la ejecución del flujo de trabajo sea la última decisión que haya enviado el decisor. Es decir, no es posible tener un conjunto de decisiones en el que hay decisiones después de la decisión de cerrar el flujo de trabajo.

Lanzamiento de decisores

Después de completar el desarrollo del decisor, está listo para lanzar uno o más decisores.

Para lanzar decisores, empaquete su lógica en un ejecutable que puede usar en su plataforma de decisor. Por ejemplo, puede empaquetar su código de decisor como ejecutable de Java que puede ejecutar tanto en equipos Linux como Windows.

Tras su lanzamiento, los decisores deben comenzar a sondear si hay tareas en Amazon SWF. Hasta que no comiencen las ejecuciones de flujos de trabajo y Amazon SWF programe las tareas de decisión, estos sondeos agotarán el tiempo de espera y obtendrán respuestas vacías. Una respuesta vacía es una estructura `Task` en la que el valor de `taskToken` es una cadena vacía. Los decisores deben simplemente continuar el sondeo.

Amazon SWF se asegura de que solo pueda haber una tarea de decisión activa para una ejecución de flujo de trabajo en cada momento. Esto evita problemas como decisiones contradictorias. Además, Amazon SWF se asegura de que solo se asigne cada tarea de decisión a un solo decisor, sin importar el número de decisores que se estén ejecutando.

Si ocurre algo que genere una tarea de decisión mientras un decisor está procesando otra tarea de decisión, Amazon SWF pone en la cola la tarea nueva hasta que se complete la tarea actual. Después de que se complete la tarea actual, Amazon SWF hace que la nueva tarea de decisión esté disponible. Además, las tareas de decisión se agrupan en lotes. Es decir, si se completan varias actividades mientras un decisor está procesando una tarea de decisión, Amazon SWF creará una única tarea de decisión nueva para dar cuenta de varias finalizaciones de tareas. Sin embargo, cada tarea finalizada recibirá un evento individual en el historial de ejecución del flujo de trabajo.

Puesto que los sondeos son solicitudes salientes, los decisores se pueden ejecutar en cualquier red con acceso al punto de conexión de Amazon SWF.

Para que las ejecuciones de flujos de trabajo progresen, deben estar ejecutándose uno o más decisores. Puede lanzar tantos decisores como desee. Amazon SWF admite el sondeo de varios decisores en la misma lista de tareas.

Inicio de flujos de trabajo en Amazon SWF

Puede iniciar una ejecución de flujo de trabajo de un tipo de flujo de trabajo registrado en cualquier aplicación mediante la acción `StartWorkflowExecution`. Al comenzar la ejecución, puede asociar un identificador, denominado `workflowId`, con ella. El `workflowId` puede ser cualquier cadena adecuada para su aplicación, como el número de pedido de una aplicación de procesamiento de pedidos. No puede usar el mismo `workflowId` para varias ejecuciones de flujo de trabajo abiertas dentro del mismo dominio. Por ejemplo, si inicia dos ejecuciones de flujo de trabajo con `workflowId Customer Order 01`, la segunda ejecución de flujo de trabajo no comenzará y se producirá un error en la solicitud. Sin embargo, puede reutilizar el `workflowId` de una ejecución cerrada. Amazon SWF también asocia un identificador único generado por el sistema, denominado `runId`, a cada ejecución de flujo de trabajo.

Una vez registrados los tipos de flujo de trabajo y actividad, comience el flujo de trabajo llamando a la acción `StartWorkflowExecution`. El valor del parámetro `input` puede ser cualquier cadena especificada por la aplicación que inicia el flujo de trabajo. `executionStartToCloseTimeout` es la duración en segundos que la ejecución de flujo de trabajo puede consumir desde que se inicia hasta que finaliza. La superación de este límite hace que se agote el tiempo de espera de la ejecución de flujo de trabajo. A diferencia de algunos de los otros parámetros de tiempo de espera de Amazon SWF, no puede especificar un valor de `NONE` para este tiempo de espera; hay un límite máximo de un año en el que una ejecución de flujo de trabajo puede ejecutarse. Del mismo modo, `taskStartToCloseTimeout` es el tiempo en segundos que puede tardar una tarea de decisión asociada a la ejecución de este flujo de trabajo antes de que se agote el tiempo de espera.

```
https://swf.us-east-1.amazonaws.com
StartWorkflowExecution
{
  "domain" : "867530901",
  "workflowId" : "20110927-T-1",
  "workflowType" : {
    "name" : "customerOrderWorkflow", "version" : "1.1"
  },
  "taskList" : { "name" : "specialTaskList" },
  "input" : "arbitrary-string-that-is-meaningful-to-the-workflow",
  "executionStartToCloseTimeout" : "1800",
  "tagList" : [ "music purchase", "digital", "ricoh-the-dog" ],
  "taskStartToCloseTimeout" : "1800",
  "childPolicy" : "TERMINATE"
}
```

Si la acción `StartWorkflowExecution` se realiza correctamente, Amazon SWF devuelve el `runId` para la ejecución de flujo de trabajo. El `runId` para una ejecución de flujo de trabajo es exclusivo dentro de una región específica. Guarde el `runId` en caso de que tenga que especificar posteriormente esta ejecución de flujo de trabajo en una llamada a Amazon SWF. Por ejemplo, usaría el `runId` si tuviera que enviar posteriormente una señal a la ejecución de flujo de trabajo.

```
{"runId": "9ba33198-4b18-4792-9c15-7181fb3a8852"}
```

Configuración de la prioridad de las tareas en Amazon SWF

De forma predeterminada, las tareas de una lista de tareas se entregan en función de su hora de llegada: las tareas que se programan primero se suelen ejecutar primero, en la medida de lo posible.

Al establecer una prioridad de las tareas opcional, puede dar prioridad a algunas tareas: Amazon SWF intentará realizar las tareas de prioridad más alta de una lista de tareas antes que las de prioridad más baja.

Note

Las tareas que se programan primero suelen ejecutarse primero, pero no está garantizado.

Puede establecer prioridades de las tareas tanto para flujos de trabajo como para actividades. La prioridad de las tareas de un flujo de trabajo no afectará a la prioridad de ninguna tarea de actividad que programe ni tampoco a ningún flujo de trabajo secundario que inicie. La prioridad predeterminada de una actividad o de un flujo de trabajo se establece (la establece el usuario, o bien Amazon SWF) durante el registro, y la prioridad de las tareas registrada siempre se utiliza a menos que se anule al programar la actividad o al iniciar una ejecución de flujo de trabajo.

Los valores de prioridad de las tareas pueden ir de "-2147483648" a "2147483647", con números más elevados que indican mayor prioridad. Si no establece la prioridad de las tareas para una actividad o flujo de trabajo, se asignará una prioridad de cero ("0").

Temas

- [Establecimiento de prioridad de las tareas para flujos de trabajo](#)
- [Establecimiento de prioridad de las tareas para actividades](#)
- [Acciones que devuelven información acerca de la prioridad de las tareas](#)

Establecimiento de prioridad de las tareas para flujos de trabajo

Puede establecer la prioridad de las tareas para un flujo de trabajo al registrarlo o iniciarlo. La prioridad de las tareas que se establece al registrarse el tipo de flujo de trabajo se usa como valor predeterminado de cualquier ejecución de flujo de trabajo de ese tipo, a menos que se anule al iniciar la ejecución de flujo de trabajo.

Para registrar un tipo de flujo de trabajo con una prioridad de tarea predeterminada, defina la `defaultTaskPriority` opción cuando utilice la [RegisterWorkflowType](#) acción:

```
{
  "domain": "867530901",
  "name": "expeditedOrderWorkflow",
```

```

"version": "1.0",
"description": "Expedited customer orders workflow",
"defaultTaskStartToCloseTimeout": "600",
"defaultExecutionStartToCloseTimeout": "3600",
"defaultTaskList": {"name": "mainTaskList"},
"defaultTaskPriority": "10",
"defaultChildPolicy": "TERMINATE"
}

```

Puede anular la prioridad de tarea registrada de un tipo de flujo de trabajo al iniciar la ejecución de un flujo de trabajo con [StartWorkflowExecution](#):

```

{
  "childPolicy": "TERMINATE",
  "domain": "867530901",
  "executionStartToCloseTimeout": "1800",
  "input": "arbitrary-string-that-is-meaningful-to-the-workflow",
  "tagList": ["music purchase", "digital", "ricoh-the-dog"],
  "taskList": {"name": "specialTaskList"},
  "taskPriority": "-20",
  "taskStartToCloseTimeout": "600",
  "workflowId": "20110927-T-1",
  "workflowType": {"name": "customerOrderWorkflow", "version": "1.0"},
}

```

También puedes anular la prioridad de la tarea registrada al iniciar un flujo de trabajo secundario o al continuar con un flujo de trabajo como nuevo, por ejemplo, cuando respondes a una decisión con [RespondDecisionTaskCompleted](#)

Para establecer una prioridad de las tareas del flujo de trabajo secundario, proporcione el valor en `startChildWorkflowExecutionDecisionAttributes`:

```

{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "StartChildWorkflowExecution",
      "startChildWorkflowExecutionDecisionAttributes": {
        "childPolicy": "TERMINATE",
        "control": "digital music",
        "executionStartToCloseTimeout": "900",
        "input": "201412-Smith-011x",

```

```

    "taskList": {"name": "specialTaskList"},
    "taskPriority": "5",
    "taskStartToCloseTimeout": "600",
    "workflowId": "verification-workflow",
    "workflowType": {
      "name": "MyChildWorkflow",
      "version": "1.0"
    }
  }
}
]
}

```

Al continuar un flujo de trabajo como nuevo, establezca la prioridad de las tareas en `continueAsNewWorkflowExecutionDecisionAttributes`:

```

{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "ContinueAsNewWorkflowExecution",
      "continueAsNewWorkflowExecutionDecisionAttributes": {
        "childPolicy": "TERMINATE",
        "executionStartToCloseTimeout": "1800",
        "input": "5634-0056-4367-0923,12/12,437",
        "taskList": {"name": "specialTaskList"},
        "taskStartToCloseTimeout": "600",
        "taskPriority": "100",
        "workflowTypeVersion": "1.0"
      }
    }
  ]
}

```

Establecimiento de prioridad de las tareas para actividades

Puede establecer la prioridad de las tareas para una actividad al registrarla o al programarla. La prioridad de las tareas que se establece al registrar un tipo de actividad se usa como prioridad predeterminada cuando se establece la actividad, a menos que se anule al programar la actividad.

Para establecer la prioridad de las tareas al registrar un tipo de actividad, defina la `defaultTaskPriority` opción cuando utilice la [RegisterActivityType](#) acción:

```
{
  "defaultTaskHeartbeatTimeout": "120",
  "defaultTaskList": {"name": "mainTaskList"},
  "defaultTaskPriority": "10",
  "defaultTaskScheduleToCloseTimeout": "900",
  "defaultTaskScheduleToStartTimeout": "300",
  "defaultTaskStartToCloseTimeout": "600",
  "description": "Verify the customer credit card",
  "domain": "867530901",
  "name": "activityVerify",
  "version": "1.0"
}
```

Para programar una tarea con una prioridad de tarea, utilice la opción `TaskPriority` al programar la actividad con la [RespondDecisionTaskCompleted](#) acción:

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "ScheduleActivityTask",
      "scheduleActivityTaskDecisionAttributes": {
        "activityId": "verify-account",
        "activityType": {
          "name": "activityVerify",
          "version": "1.0"
        },
        "control": "digital music",
        "input": "abab-101",
        "taskList": {"name": "mainTaskList"},
        "taskPriority": "15"
      }
    }
  ]
}
```

Acciones que devuelven información acerca de la prioridad de las tareas

Puede obtener información sobre la prioridad de las tareas establecida (o establecer la prioridad predeterminada de las tareas) mediante las siguientes acciones de Amazon SWF:

- [DescribeActivityType](#) devuelve el tipo defaultTaskPriority de actividad de la configuration sección de la respuesta.
- [DescribeWorkflowExecution](#) devuelve la TaskPriority de la ejecución del flujo de trabajo en la executionConfiguration sección de la respuesta.
- [DescribeWorkflowType](#) devuelve el tipo defaultTaskPriority de flujo de trabajo de la configuration sección de la respuesta.
- [GetWorkflowExecutionHistory](#) y [PollForDecisionTask](#) proporcione información sobre la prioridad de las tareas en workflowExecutionStartedEventAttributes las secciones activityTaskScheduledEventAttributes decisionTaskScheduledEventAttributes workflowExecutionContinuedAsNewEventAttributes y de la respuesta.

Gestión de errores en Amazon SWF

Hay una serie de diferentes tipos de errores que pueden ocurrir durante el curso de una ejecución de flujo de trabajo.

Temas

- [Errores de validación](#)
- [Errores en la aplicación de acciones o decisiones](#)
- [Tiempos de espera](#)
- [Errores planteados el código de usuario](#)
- [Errores relacionados con el cierre de una ejecución de flujo de trabajo](#)

Errores de validación

Los errores de validación ocurren cuando se produce un error en una solicitud a Amazon SWF porque no tiene el formato correcto o contiene datos no válidos. En este contexto, una solicitud puede ser una acción como `DescribeDomain` o una decisión como `StartTimer`. Si la solicitud es una acción, Amazon SWF devuelve un código de error en la respuesta. Compruebe el código de error porque puede proporcionar información sobre qué aspecto de la solicitud provocó el error. Por ejemplo, puede que uno o más argumentos transferidos con la solicitud no sean válidos. Para obtener una lista de los códigos de error más comunes, consulte el tema correspondiente a la acción en la referencia de la API de Amazon Simple Workflow Service (Amazon SWF).

Si la solicitud que produjo el error es una decisión, se mostrará un evento apropiado en el historial de ejecución del flujo de trabajo. Por ejemplo, si la decisión `StartTimer` produce un error, verá el evento `StartTimerFailed` en el historial. El decisor debe comprobar la presencia de estos eventos cuando recibe el historial en respuesta a `PollForDecisionTask` o `GetWorkflowExecutionHistory`. A continuación se muestra una lista de posibles eventos de error que pueden ocurrir si la decisión no tiene el formato correcto o contiene datos no válidos.

Errores en la aplicación de acciones o decisiones

Incluso si la solicitud tiene el formato correcto, pueden producirse errores cuando Amazon SWF intente llevar a cabo la solicitud. En ese caso, uno de los siguientes eventos en el historial indicará que se ha producido un error. Examine el campo `reason` del evento para determinar la causa del error.

- [CancelTimerFailed](#)
- [RequestCancelActivityTaskFailed](#)
- [RequestCancelExternalWorkflowExecutionFailed](#)
- [ScheduleActivityTaskFailed](#)
- [SignalExternalWorkflowExecutionFailed](#)
- [StartChildWorkflowExecutionFailed](#)
- [StartTimerFailed](#)

Tiempos de espera

Los [decisores](#), los [procesos de trabajo de actividad](#) y las [ejecuciones de flujos de trabajo](#) están sometidos a las limitaciones de los periodos de tiempo de espera. En este tipo de error, una tarea o un flujo de trabajo secundario agota el tiempo de espera. Se mostrará en el historial un evento que describe el tiempo de espera. Para gestionar este evento el decisor debe, por ejemplo, reprogramar la tarea o reiniciar el flujo de trabajo secundario. Para obtener más información acerca de tiempos de espera, consulte [Tipos de tiempo de espera de Amazon SWF](#)

- [ActivityTaskTimedOut](#)
- [ChildWorkflowExecutionTimedOut](#)
- [DecisionTaskTimedOut](#)
- [WorkflowExecutionTimedOut](#)

Errores planteados el código de usuario

Este tipo de condición de error corresponde, por ejemplo, a errores de tareas de actividad y de flujos de trabajo secundarios. Al igual que con los errores de tiempo de espera, Amazon SWF añade un evento apropiado al historial de ejecución del flujo de trabajo. Para gestionar este evento el decisor puede, por ejemplo, reprogramar la tarea o reiniciar el flujo de trabajo secundario.

- [ActivityTaskFailed](#)
- [ChildWorkflowExecutionFailed](#)

Errores relacionados con el cierre de una ejecución de flujo de trabajo

Los decisores también pueden ver los siguientes eventos si intentan cerrar un flujo de trabajo con una tarea de decisión pendiente.

- [FailWorkflowExecutionFailed](#)
- [CompleteWorkFlowExecutionFailed](#)
- [ContinueAsNewWorkflowExecutionFailed](#)
- [CancelWorkflowExecutionFailed](#)

Para obtener más información sobre los eventos indicados anteriormente, consulte [Eventos del historial](#) en la Referencia de la API de Amazon SWF.

Cuotas de Amazon SWF

Amazon SWF impone cuotas en los tamaños de ciertos parámetros del flujo de trabajo, como el número de dominios por cuenta y el tamaño del historial de ejecución del flujo de trabajo. Estas cuotas se han diseñado para impedir que flujos de trabajo erróneos consuman todos los recursos del sistema, pero no son inflexibles. Si constata que su aplicación supera frecuentemente esas cuotas, puede [solicitar un aumento de la cuota de servicio](#).

Contenido

- [Cuotas de cuentas generales para Amazon SWF](#)
- [Cuotas de ejecución de flujos de trabajo](#)
- [Cuotas de ejecuciones de tareas](#)
- [Cuotas de limitación de Amazon SWF](#)
 - [Cuotas de limitación para todas las regiones](#)
 - [Cuotas de decisión para todas las regiones](#)
 - [Cuotas a nivel de los flujos de trabajo](#)
- [Solicitud de aumento de cuota](#)

Cuotas de cuentas generales para Amazon SWF

- Dominios registrados como máximo: 100

Esta cuota incluye tanto dominios registrados como dominios obsoletos.

- Tipos de flujo de trabajo y actividad como máximo: 10 000 por cada dominio

Esta cuota incluye tanto tipos registrados como tipos obsoletos.

- Cuota de llamadas a la API: además de los picos infrecuentes, las aplicaciones pueden sufrir limitaciones si realizan un gran número de llamadas a la API en un periodo de tiempo muy corto.
- Tamaño máximo de solicitud: 1 MB por solicitud

Este es el tamaño total de los datos para cada solicitud API de Amazon SWF, incluido el encabezado de la solicitud y todos los demás datos de solicitud relacionados.

- Respuestas truncadas para el recuento APIs: indica que se alcanzó una cuota interna y que la respuesta no es el recuento completo.

Es posible que algunas consultas alcancen internamente la cuota de 1 MB mencionada anteriormente antes de devolver una respuesta total. Las siguientes acciones pueden devolver una respuesta troncada en lugar del recuento total.

- [CountClosedWorkflowExecutions](#)
- [CountOpenWorkflowExecutions](#)
- [CountPendingActivityTasks](#)
- [CountPendingDecisionTasks](#)

En cada una de estas acciones, si se configura la respuesta `truncated` en verdadero, el recuento es menor que la cantidad total. Esta cuota interna no se puede aumentar.

- Número máximo de etiquetas: 50 etiquetas por recurso.

Si intenta añadir más de 50 etiquetas, se producirá un error 400: `TooManyTagsFault`.

Cuotas de ejecución de flujos de trabajo

- Ejecuciones de flujos de trabajo abiertos como máximo: 100 000 por dominio

Este número incluye ejecuciones de flujos de trabajo secundarios.

- Tiempo máximo de ejecución de flujos de trabajo: 1 año. Esta es una cuota rígida que no se puede cambiar.
- Tamaño máximo del historial de ejecuciones de flujos de trabajo: 25 000 eventos. Esta es una cuota rígida que no se puede cambiar.

La práctica recomendada es estructurar cada flujo de trabajo de modo tal que su historial no contenga más de 10 000 eventos. Como el decisor tiene que buscar el historial del flujo de trabajo, un historial más pequeño le permite completar el proceso con mayor rapidez. Si utilizas el [Flow Framework](#), puedes usarlo `ContinueAsNew` para continuar un flujo de trabajo con un historial nuevo.

- Ejecuciones de flujos de trabajo secundarios abiertos como máximo: 1000 por ejecución de flujo de trabajo

Si su caso de uso necesita superar estas cuotas, puede utilizar las características que proporciona Amazon SWF para continuar las ejecuciones y estructurar sus aplicaciones por medio de

ejecuciones de [flujos de trabajo secundarios](#). Si considera que aún necesita un aumento de cuotas, consulte [Solicitud de aumento de cuota](#).

Cuotas de ejecuciones de tareas

- Número máximo de sondeadores por lista de tareas: 1000 por lista de tareas

Puede tener un máximo de 1 000 sondeadores que realicen sondeos simultáneamente en una lista de tareas en particular. Si se supera la cuota de 1000, recibirá un código `LimitExceededException`.

Note

Si bien el máximo es de 1000, pueden encontrarse errores `LimitExceededException` antes de alcanzar esta cuota. Este error no significa que las tareas se estén retrasando. En cambio, significa que tiene la cantidad máxima de sondeadores inactivos en una lista de tareas. Amazon SWF establece este límite para ahorrar recursos tanto en el lado del cliente como en el del servidor. Si se establece el límite, se evita que un número excesivo de sondeadores se queden innecesariamente en espera. Para reducir los errores `LimitExceededException`, puede utilizar varias listas de tareas para distribuir el sondeo.

- Máximo de tareas programadas por segundo: 2000 por lista de tareas

Puede programar un máximo de 2000 tareas por segundo en una lista de tareas determinada. Si superas las 2000, las decisiones `ScheduleActivityTask` fallarán con el error `ACTIVITY_CREATION_RATE_EXCEEDED`.

Note

Si bien el máximo es de 2000, pueden encontrarse errores `ACTIVITY_CREATION_RATE_EXCEEDED` antes de alcanzar esa cuota. Para reducir esos errores, utilice varias listas de tareas para distribuir la carga.

- Tiempo máximo de ejecución de la tarea: 1 año (limitado por el tiempo máximo de ejecución de flujos de trabajo)

Puede configurar los [tiempos de espera de actividad](#) para provocar que se produzca un evento de tiempo de espera si una determinada etapa de la ejecución de su [tarea de actividad](#) tarda demasiado tiempo.

- Tiempo máximo que SWF mantendrá una tarea en cola: 1 año (limitado por la cuota de tiempo de ejecución de flujos de trabajo)

Puede configurar [tiempos de espera de actividad](#) predeterminados durante el registro de la actividad para que se produzca un evento de tiempo de espera si una determinada etapa de la ejecución de la [tarea de actividad](#) tarda demasiado tiempo. También puede omitir los tiempos de espera de actividad predeterminados cuando programa una tarea de actividad en el código del decisor.

- Tareas de actividad abiertas como máximo: 1000 por ejecución de flujo de trabajo.

Esta cuota incluye tanto las tareas de actividad que se han programado como las que los procesos de trabajo están procesando.

- Temporizadores abiertos como máximo: 1000 por ejecución de flujo de trabajo.
- Tamaño máximo input/result de los datos: 32.768 caracteres

Esta cuota afecta a los datos de resultados de las ejecuciones de flujos de trabajo o actividades, a los datos de entrada cuando se programan ejecuciones de flujos de trabajo o tareas de actividad y a las entradas enviadas con una [señal de ejecución del flujo de trabajo](#).

- Máximo de decisiones en la respuesta a una tarea de decisión: varía.

Debido a que la cuota del [tamaño máximo de la solicitud de la API](#) es 1 MB, el número de decisiones devueltas en una sola llamada a [RespondDecisionTaskCompleted](#) se limita en función del tamaño de los datos que utilice cada decisión, incluido el tamaño de todos los datos de entrada proporcionados para las tareas de actividad o las ejecuciones de flujos de trabajo programadas.

Cuotas de limitación de Amazon SWF

Además de las cuotas de servicio descritas anteriormente, se establece una limitación a algunas llamadas a la API de Amazon SWF y a algunos eventos de decisión para mantener el ancho de banda del servicio, con la ayuda de un esquema de [bucket de token](#). Si el número de solicitudes supera constantemente los límites aquí indicados, puede [solicitar un aumento de la cuota de limitación](#).

Las cuotas de limitación y de decisión son las mismas en todas las regiones.

Cuotas de limitación para todas las regiones

Las siguientes cuotas se aplican a los niveles de cuentas individuales. También puede solicitar un aumento para las cuotas siguientes. Para obtener información sobre cómo hacerlo, consulte [Solicitud de aumento de cuota](#).

Nombre de API	Tamaño del bucket	Velocidad de reposición por segundo
CountClosedWorkflowExecutions	2000	6
CountOpenWorkflowExecutions	2000	6
CountPendingActivityTasks	200	6
CountPendingDecisionTasks	200	6
DeleteActivityType	200	6
DeleteWorkflowType	200	6
DeprecateActivityType	200	6
DeprecateDomain	100	6
DeprecateWorkflowType	200	6
DescribeActivityType	2000	6
DescribeDomain	200	6
DescribeWorkflowExecution	2000	6
DescribeWorkflowType	2000	6
GetWorkflowExecutionHistory	2000	60
ListActivityTypes	200	6

Nombre de API	Tamaño del bucket	Velocidad de reposición por segundo
ListClosedWorkflowExecutions	200	6
ListDomains	100	6
ListOpenWorkflowExecutions	200	48
ListTagsForResource	50	30
ListWorkflowTypes	200	6
PollForActivityTask	2000	200
PollForDecisionTask	2000	200
RecordActivityTaskHeartbeat	2000	160
RegisterActivityType	200	60
RegisterDomain	100	6
RegisterWorkflowType	200	60
RequestCancelWorkflowExecution	2000	30
RespondActivityTaskCanceled	2000	200
RespondActivityTaskCompleted	2000	200
RespondActivityTaskFailed	2000	200
RespondDecisionTaskCompleted	2000	200
SignalWorkflowExecution	2000	30
StartWorkflowExecution	2000	200
TagResource	50	30

Nombre de API	Tamaño del bucket	Velocidad de reposición por segundo
TerminateWorkflowExecution	2000	60
UndeprecateActivityType	200	6
UndeprecateDomain	100	6
UndeprecateWorkflowType	200	6
UntagResource	50	30

Cuotas de decisión para todas las regiones

Las siguientes cuotas se aplican a los niveles de cuentas individuales. También puede solicitar un aumento para las cuotas siguientes. Para obtener información sobre cómo hacerlo, consulte [Solicitud de aumento de cuota](#).

Nombre de API	Tamaño del bucket	Velocidad de reposición por segundo
RequestCancelExternalWorkflowExecution	1200	120
ScheduleActivityTask	1 000	200
SignalExternalWorkflowExecution	1200	120
StartChildWorkflowExecution	500	12
StartTimer	2000	200

Cuotas a nivel de los flujos de trabajo

Las siguientes cuotas se aplican a nivel de los flujos de trabajo y no se pueden aumentar.

Nombre de API	Tamaño del bucket	Velocidad de reposición por segundo
GetWorkflowExecutionHistory	400	200
SignalWorkflowExecution	1 000	1 000
RecordActivityTaskHeartbeat	1 000	1 000
RequestCancelWorkflowExecution	200	200

Solicitud de aumento de cuota

Para obtener más información, consulte el artículo sobre [AWS Service Quotas](#) en la Referencia general de AWS.

Recursos e información de referencia adicionales para Amazon SWF

Este capítulo ofrece recursos adicionales e información de referencia que resultan útiles cuando se desarrollan flujos de trabajo con Amazon SWF.

Temas

- [Tipos de tiempo de espera de Amazon SWF](#)
- [Puntos de conexión de Amazon Simple Workflow Service](#)
- [Documentación adicional para Amazon Simple Workflow Service](#)
- [Recursos web para Amazon Simple Workflow Service](#)
- [Opciones de migración para Ruby Flow](#)

Tipos de tiempo de espera de Amazon SWF

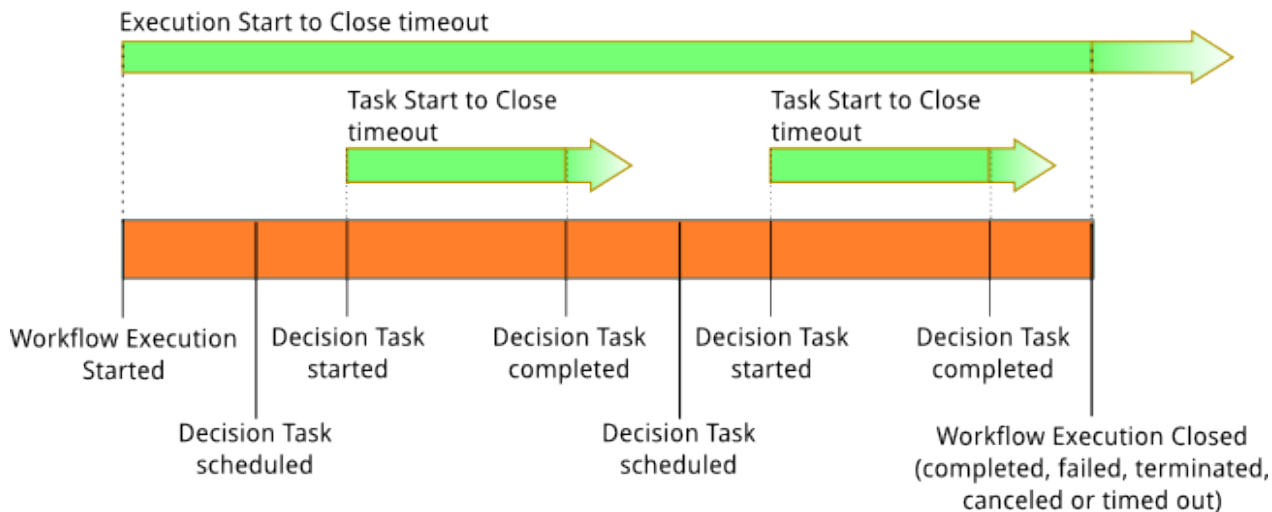
Para garantizar que las ejecuciones del flujo de trabajo se ejecuten correctamente, puede establecer distintos tipos de tiempos de espera con Amazon SWF. Algunos tiempos de espera especifican cuánto puede tardar el flujo de trabajo en ejecutarse en su totalidad. Otros tiempos de espera especifican cuánto pueden tardar las tareas de actividad en asignarse a un proceso de trabajo y cuánto pueden tardar en completarse desde el momento de su programación. Todos los tiempos de espera de la API de Amazon SWF se especifican en segundos. Amazon SWF también admite la cadena NONE como valor de tiempo de espera, lo que indica que no se establecerá ningún tiempo de espera.

Para los tiempos de espera relacionados con las tareas de decisión y actividad, Amazon SWF añade un evento al historial de ejecución del flujo de trabajo. Los atributos del evento proporcionan información acerca del tipo de tiempo de espera que tuvo lugar y la tarea de decisión o actividad que se vio afectada. Amazon SWF también programa una tarea de decisión. Cuando el decisor reciba la nueva tarea de decisión, verá el evento de tiempo de espera en el historial y tomará la acción adecuada al solicitarla. [RespondDecisionTaskCompleted](#)

Una tarea se considera abierta desde el momento en que se programa hasta que se cierra. Por tanto, una tarea se registra como abierta mientras un proceso de trabajo la procesa. Una tarea se cierra cuando un proceso de trabajo la registra como [Completed](#), [Canceled](#) o [Failed](#). Amazon SWF también puede cerrar una tarea como resultado de un tiempo de espera.

Tiempos de espera de las tareas de decisión y flujo de trabajo

En el siguiente diagrama se muestra cómo los tiempos de espera de decisión y flujo de trabajo están relacionados con la vida útil de un flujo de trabajo:



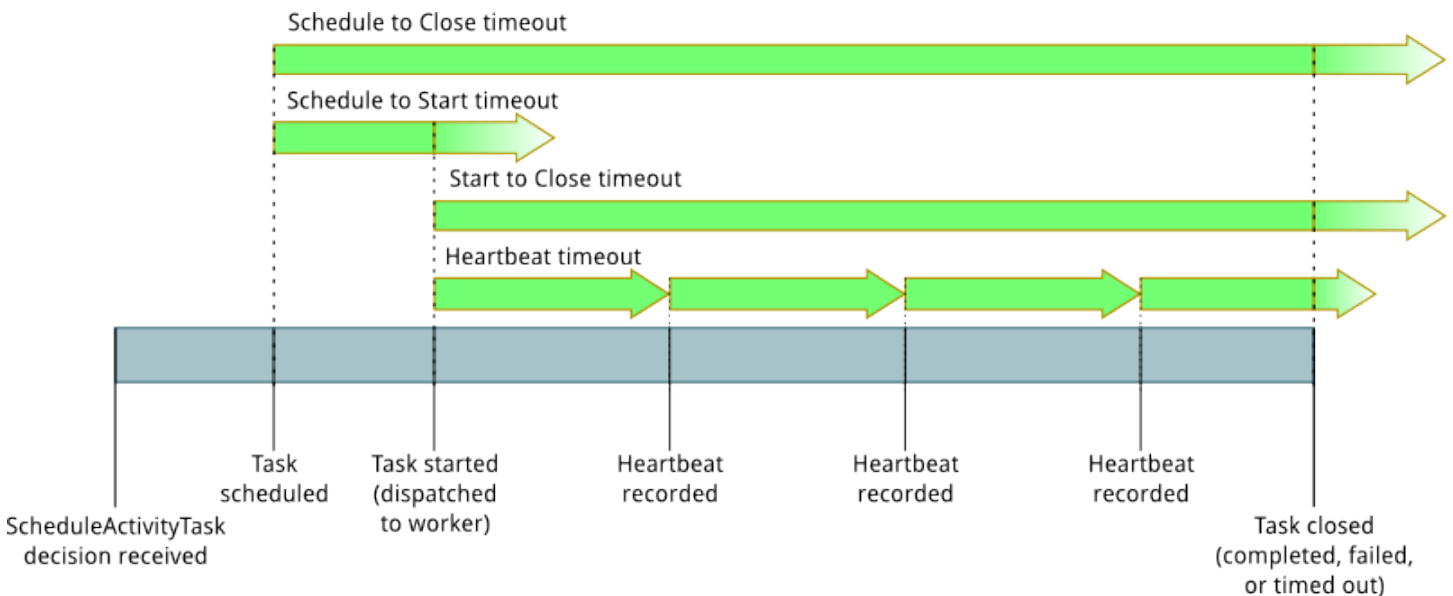
Hay dos tipos de tiempo de espera que son pertinentes para las tareas de decisión y flujo de trabajo:

- Tiempo de espera de inicio a cierre del flujo de trabajo (**timeoutType: START_TO_CLOSE**): este tiempo de espera especifica el tiempo máximo que puede tardar en completarse la ejecución de un flujo de trabajo. Se establece como valor predeterminado durante el registro del flujo de trabajo, pero se puede anular con un valor diferente al iniciarse el flujo de trabajo. Si se supera este tiempo de espera, Amazon SWF cierra la ejecución del flujo de trabajo y añade un [WorkflowExecutionTimedOut](#) tipo de evento al historial de ejecución del flujo de trabajo. Además de `timeoutType`, los atributos del evento especifican la `childPolicy` que se encuentra en vigor para esta ejecución de flujo de trabajo. La política secundaria especifica cómo se controlan las ejecuciones de los flujos de trabajo secundarios si se agota el tiempo de espera de la ejecución de flujo de trabajo principal o termina por otro motivo. Por ejemplo, si `childPolicy` se establece en `TERMINATE`, las ejecuciones de flujos de trabajo secundarios se terminarán. Una vez que se haya agotado el tiempo de espera de una ejecución de flujo de trabajo, la única medida que podrá tomar al respecto son las llamadas de visibilidad.
- Tiempo de espera de inicio a cierre de la tarea de decisión (**timeoutType: START_TO_CLOSE**): este tiempo de espera especifica el tiempo máximo que el decisor correspondiente puede tardar en completar una tarea de decisión. Se establece durante el registro del tipo de flujo de trabajo. Si se supera este tiempo de espera, la tarea se marca como agotada en el historial de ejecución del flujo de trabajo y Amazon SWF añade un [DecisionTaskTimedOut](#) tipo de evento al historial del flujo de trabajo. Los atributos del evento incluirán los eventos IDs correspondientes al momento

en que se programó la tarea de decisión (`scheduledEventId`) y al momento en que se inició (`startedEventId`). Además de añadir el evento, Amazon SWF también programa una tarea de decisión nueva para alertar al decisor de que se ha agotado el tiempo de espera de dicha tarea de decisión. Tras agotarse este tiempo de espera, un intento de completar la tarea de decisión con tiempo de espera agotado mediante `RespondDecisionTaskCompleted` producirá un error.

Tiempos de espera de las tareas de actividad

En el siguiente diagrama se muestra cómo los tiempos de espera están relacionados con la vida útil de una tarea de actividad:



Hay cuatro tipos de tiempo de espera que son pertinentes para las tareas de actividad:

- Tiempo de espera de inicio a cierre de la tarea de actividad (**timeoutType: START_TO_CLOSE**): este tiempo de espera especifica el tiempo máximo que el proceso de trabajo de una actividad puede tardar en procesar una tarea después de haberla recibido. Los intentos de cerrar una tarea de actividad agotada utilizando [RespondActivityTaskCanceled](#), [RespondActivityTaskCompleted](#), y [RespondActivityTaskFailed](#) fallarán.
- Latido de la tarea de actividad (**timeoutType: HEARTBEAT**): este tiempo de espera especifica el tiempo máximo que se puede ejecutar una tarea antes de indicar su progreso en la acción `RecordActivityTaskHeartbeat`.
- Programación de tareas de actividad hasta su inicio (**timeoutType: SCHEDULE_TO_START**): este tiempo de espera especifica cuánto tiempo espera Amazon SWF antes de agotar el tiempo de

espera de la tarea de actividad si no hay procesos de trabajo disponibles para realizarla. Una vez que se agota el tiempo de espera, la tarea caducada no se asignará a otro proceso de trabajo.

- Programación de tareas de actividad hasta su cierre (**timeoutType: SCHEDULE_TO_CLOSE**): este tiempo de espera especifica cuánto tiempo puede tardar la tarea desde el momento en que se programe hasta el momento en que se complete. Como práctica recomendada, este valor no debe ser mayor que la suma del tiempo de espera de la tarea y el schedule-to-start tiempo de espera de la tarea start-to-close.

Note

Cada uno de los tipos de tiempo de espera tiene un valor predeterminado, que generalmente se establece en NONE (infinito). El tiempo máximo de cualquier ejecución de actividad se limita a un año, sin embargo.

Puede establecer valores predeterminados para estos durante el registro del tipo de actividad, pero puede anularlos con nuevos valores al [programar](#) la tarea de actividad. Cuando se agote uno de estos tiempos de espera, Amazon SWF añadirá [un ActivityTaskTimedOut](#) tipo de evento al historial del flujo de trabajo. El atributo de valor timeoutType de este evento especificará cuál de estos tiempos de espera ha tenido lugar. Para cada uno de los tiempos de espera, el valor de timeoutType se muestra entre paréntesis. Los atributos del evento también incluirán los eventos IDs correspondientes al momento en que se programó la tarea de la actividad (scheduledEventId) y al momento en que se inició (startedEventId). Además de añadir el evento, Amazon SWF también programa una tarea de decisión nueva para alertar al decisor de que se ha agotado el tiempo de espera.

Puntos de conexión de Amazon Simple Workflow Service

En Referencia general de Amazon Web Services se proporciona una lista de las [regiones y puntos de conexión actuales de Amazon SWF](#), junto con los puntos de conexión de otros servicios.

Los dominios de Amazon SWF y todos los flujos de trabajo y actividades relacionados deben existir en la misma región para comunicarse entre sí. Además, los dominios, flujos de trabajo y actividades registrados en una región no existen en otras regiones. Por ejemplo, si crea un dominio denominado «MySampleDomain» tanto en us-east-1 como en us-west-2, existen como dominios independientes: ninguno de los flujos de trabajo, listas de tareas, actividades o datos asociados a sus dominios se comparte entre regiones.

Si utiliza otros AWS recursos en sus flujos de trabajo, como EC2 instancias de Amazon, estos también deben existir en la misma región que sus recursos de Amazon SWF. Las únicas excepciones a esto son los servicios que abarcan varias regiones, como Amazon S3 e IAM. Puede obtener acceso a estos servicios desde los flujos de trabajo existentes en cualquier región que los admita.

Documentación adicional para Amazon Simple Workflow Service

Además de esta Guía para desarrolladores, puede que le resulte útil la siguiente documentación.

Referencia de la API de Amazon Simple Workflow Service

La [referencia de la API de Amazon Simple Workflow Service](#) proporciona información detallada acerca de la API HTTP de Amazon SWF, incluidas las acciones, las estructuras de solicitud y respuesta y los códigos de error.

AWS Flow Framework Documentación

El [AWS Flow Framework](#) es un marco de trabajo de programación que simplifica el proceso de implementación de aplicaciones asincrónicas distribuidas que utiliza Amazon SWF para administrar sus flujos de trabajo y actividades, lo que permite al usuario centrarse en la implementación de la lógica de los flujos de trabajo.

Cada uno AWS Flow Framework está diseñado para funcionar idiomáticamente en el idioma para el que está diseñado, por lo que puede trabajar de forma natural con el idioma que prefiera para implementar flujos de trabajo con todas las ventajas de Amazon SWF.

Existe un marco de AWS flujo para Java. La [guía AWS Flow Framework para desarrolladores de Java](#) proporciona información sobre cómo obtener, configurar y utilizar el AWS Flow Framework para Java.

AWS Documentación del SDK

Los kits de desarrollo de AWS software (SDKs) proporcionan acceso a Amazon SWF en muchos lenguajes de programación diferentes. SDKs Siguen de cerca la API HTTP, pero también proporcionan interfaces de programación específicas del idioma para algunas funciones de Amazon SWF. Puede encontrar más información acerca de cada SDK visitando los siguientes enlaces.

Note

Aquí solo SDKs se muestran las aplicaciones compatibles con Amazon SWF en el momento de redactar este artículo. Para ver una lista completa de las disponibles AWS SDKs, visita la página [Herramientas para Amazon Web Services](#).

Java

AWS SDK para Java Proporciona una API de Java para servicios de AWS infraestructura.

Para ver la documentación disponible, consulte la página [Documentación de AWS SDK para Java](#). También puede utilizar estos enlaces para ir directamente a las secciones de Amazon SWF en la referencia del SDK:

- [Class: AmazonSimpleWorkflowClient](#)
- [Class: AmazonSimpleWorkflowAsyncClient](#)
- [Interface: AmazonSimpleWorkflow](#)
- [Interface: AmazonSimpleWorkflowAsync](#)

JavaScript

AWS SDK para JavaScript Permite a los desarrolladores crear bibliotecas o aplicaciones que utilicen los AWS servicios mediante una easy-to-use API simple y disponible tanto en el navegador como dentro de las aplicaciones Node.js del servidor.

Para ver la documentación disponible, consulte la página [Documentación de AWS SDK para JavaScript](#). También puede utilizar este enlace para ir directamente a la sección de Amazon SWF en la referencia del SDK:

- [Class: AWS.SimpleWorkflow](#)

.NET

AWS SDK para .NET Se trata de un único paquete descargable que incluye plantillas de proyectos de Visual Studio, la AWS biblioteca.NET, ejemplos de código de C# y documentación. AWS SDK para .NET Esto facilita a los desarrolladores de Windows la creación de aplicaciones.NET para Amazon SWF y otros servicios.

Para ver la documentación disponible, consulte la página [Documentación de AWS SDK para .NET](#). También puede utilizar estos enlaces para ir directamente a las secciones de Amazon SWF en la referencia del SDK:

- [Namespace: Amazon.SimpleWorkflow](#)
- [Namespace: Amazon.SimpleWorkflow.Model](#)

PHP

AWS SDK for PHP Proporciona una interfaz de programación PHP para Amazon SWF.

Para ver la documentación disponible, consulte la página [Documentación de AWS SDK for PHP](#). También puede utilizar este enlace para ir directamente a la sección de Amazon SWF en la referencia del SDK:

- [Class: SwfClient](#)

Python

AWS SDK para Python (Boto) Proporciona una interfaz de programación en Python para Amazon SWF.

Para ver la documentación disponible, consulte [Boto: una interfaz Python de Amazon Web Services](#). También puede ir directamente a las secciones de Amazon SWF en la documentación mediante estos enlaces:

- [Tutorial de Amazon SWF](#)
- [Referencia de Amazon SWF](#)

Ruby

AWS SDK para Ruby Proporciona una interfaz de programación Ruby para Amazon SWF.

Para ver la documentación disponible, consulte la página [Documentación de AWS SDK para Ruby](#). También puede utilizar este enlace para ir directamente a la sección de Amazon SWF en la referencia del SDK:

- [Clase: AWS::Simple Flujo de trabajo](#)

AWS CLI Documentación

The AWS Command Line Interface (AWS CLI) es una herramienta unificada para administrar sus AWS servicios. Con solo una herramienta para descargar y configurar, puede controlar varios AWS servicios desde la línea de comandos y automatizarlos mediante scripts.

Para obtener más información sobre el AWS CLI, consulte la [AWS Command Line Interface](#) página.

Para obtener información general acerca de los comandos de Amazon SWF, consulte [swf](#) en la Referencia de comandos de la AWS CLI .

Recursos web para Amazon Simple Workflow Service

Hay diversos recursos web que puede utilizar para obtener más información sobre Amazon SWF o para obtener ayuda con el uso del servicio y el desarrollo de flujos de trabajo.

Foro de Amazon SWF

El foro de Amazon SWF proporciona un lugar para hacer preguntas y obtener respuestas de otros desarrolladores de Amazon SWF y miembros del equipo de desarrollo de Amazon SWF en Amazon.

Puede visitar el foro en: [Foro: Amazon Simple Workflow Service](#).

Preguntas frecuentes sobre Amazon SWF

En la sección de preguntas frecuentes sobre Amazon SWF se proporcionan respuestas a preguntas frecuentes sobre Amazon SWF, además de información general de casos de uso comunes, diferencias entre Amazon SWF y otros servicios, etc.

Puede obtener acceso a las preguntas frecuentes aquí: [Preguntas frecuentes sobre Amazon SWF](#).

Vídeos de Amazon SWF

El canal [Amazon Web Services](#) YouTube ofrece formación en vídeo sobre todos los servicios web de Amazon, incluido Amazon SWF. Para obtener una lista completa de los vídeos relacionados con Amazon SWF, utilice la siguiente consulta: [Simple Workflow in Amazon Web Services](#)

Opciones de migración para Ruby Flow

The AWS Flow Framework for Ruby ya no está en desarrollo activo. Aunque el código existente seguirá funcionando indefinidamente, no habrá características ni versiones nuevas. En este tema, se tratarán las opciones de uso y de migración para seguir trabajando con Amazon SWF y la información sobre cómo migrar a Step Functions.

Opción	Descripción
Seguir usando Ruby Flow Framework	Por el momento, Ruby Flow Framework seguirá funcionando. Si no hace nada, el código seguirá funcionando tal como está. Planifique migrar de Ruby AWS Flow Framework a Ruby en un futuro próximo.
Migrar a Java Flow Framework	Java Flow Framework se sigue desarrollando activamente y seguirá recibiendo nuevas características y actualizaciones.
Migrar a Step Functions	Step Functions proporciona una forma de coordinar los componentes de aplicaciones y microservicios distribuidos con flujos de trabajo visuales controlados por una máquina de estado.
Usar la API de SWF directamente , sin Flow Framework	Puede seguir trabajando en Ruby y usar la API de SWF directamente en vez de Ruby Flow Framework.

La ventaja que proporciona Flow Framework, tanto para Ruby como para Java, es que le permite centrarse en la lógica del flujo de trabajo. El marco gestiona gran parte de los detalles de la comunicación y la coordinación, y se abstrae parte de la complejidad. Puede seguir teniendo el mismo nivel de abstracción si migra al Java Flow Framework, o puede interactuar directamente con el SDK de Amazon SWF.

Seguir usando el Ruby Flow Framework

El AWS Flow Framework for Ruby seguirá funcionando como lo hace ahora a corto plazo. Si tiene flujos de trabajo escritos AWS Flow Framework para Ruby, estos seguirán funcionando. Sin actualizaciones, soporte ni correcciones de seguridad, lo mejor es disponer de un plan bien definido para llevar a cabo la migración del AWS Flow Framework para Ruby en un futuro próximo.

Migrar al Java Flow Framework

El AWS Flow Framework para Java permanecerá en desarrollo activo. Conceptualmente, el AWS Flow Framework caso de Java es similar al AWS Flow Framework de Ruby: puede seguir centrándose en la lógica del flujo de trabajo y el marco le ayudará a gestionar la lógica de sus decisiones y facilitará la gestión de otros aspectos de Amazon SWF.

- [AWS Flow Framework para Java](#)

- [AWS Flow Framework para la referencia de la API de Java](#)

Migrar a Step Functions

AWS Step Functions proporciona un servicio similar a Amazon SWF, pero en el que la lógica del flujo de trabajo está controlada por una máquina de estados. Step Functions le permite coordinar los componentes de aplicaciones y microservicios distribuidos mediante flujos de trabajo visuales. La creación de aplicaciones a partir de componentes individuales cada uno de los cuales realiza una función o tarea discreta le permite escalar y modificar las aplicaciones rápidamente. Step Functions proporciona una manera fiable de coordinar los componentes y procesar las funciones de su aplicación. Una consola gráfica proporciona una forma de visualizar los componentes de su aplicación en varios pasos. Activa y monitoriza cada paso de manera automática, y realiza reintentos cuando se producen errores, por lo que su aplicación se ejecuta en orden y según lo previsto en todo momento. Step Functions registra el estado de cada paso, de manera que, cuando algo sale mal, puede diagnosticar y depurar los problemas con rapidez.

En Step Functions, la coordinación de las tareas se administra a través de una máquina de estado escrita en JSON declarativo que se define mediante el [Amazon States Language](#). Al utilizar una máquina de estados, no es necesario escribir ni mantener un programa de decisor para controlar la lógica de la aplicación. Step Functions proporciona un enfoque intuitivo, productivo y ágil para coordinar los componentes de la aplicación mediante flujos de trabajo visuales. Deberías considerar usarlo AWS Step Functions para todas tus nuevas aplicaciones, y Step Functions proporciona una excelente plataforma a la que migrar para los flujos de trabajo que tienes implementados actualmente en Ruby. AWS Flow Framework

Para ayudarle a migrar las tareas a Step Functions, así como a seguir aprovechando sus conocimientos del lenguaje Ruby, Step Functions proporciona un ejemplo de proceso de trabajo de actividad de Ruby. Este ejemplo utiliza las prácticas recomendadas para implementar un proceso de trabajo de actividad y se puede utilizar como plantilla para migrar la lógica de las tareas a Step Functions. Para obtener más información, consulte [Ejemplo de proceso de trabajo de actividad en Ruby](#) en la [Guía para desarrolladores de AWS Step Functions](#).

Note

Para muchos clientes, migrar a Step Functions desde Ruby es la mejor opción. AWS Flow Framework Sin embargo, si necesita que las señales intervengan en sus procesos o si necesita lanzar procesos secundarios que devuelvan un resultado a un elemento principal,

considere la posibilidad de utilizar directamente la API de Amazon SWF o de migrar a la AWS Flow Framework API para Java.

Para obtener más información al respecto AWS Step Functions, consulte:

- [AWS Step Functions Guía para desarrolladores](#)
- [AWS Step Functions Referencia de la API](#)
- [AWS Step Functions Referencia de la línea de comandos](#)

Utilizar la API de Amazon SWF directamente

Si bien AWS Flow Framework for Ruby gestiona parte de la complejidad de Amazon SWF, también puede utilizar la API de Amazon SWF directamente. El uso directo de la API le permite crear flujos de trabajo donde tiene control total de la implementación y coordinación de tareas sin tener que preocuparse de las complejidades que hay detrás, como el seguimiento de progresos o el mantenimiento de estados.

- [Guía para desarrolladores de Amazon Simple Workflow Service.](#)
- [Referencia de la API de Amazon Simple Workflow Service](#)

Historial del documento

En la siguiente tabla, se describen los cambios importantes que se han realizado en la documentación desde la última versión de la Guía para desarrolladores de Amazon Simple Workflow Service.

Cambio	Descripción	Fecha de modificación
Actualización de la documentación únicamente	Amazon SWF ahora incluye una sección sobre las notificaciones de los AWS usuarios, Servicio de AWS que actúa como ubicación central para AWS las notificaciones en el. Consola de administración de AWS Para obtener más información, consulte Uso AWS User Notifications con Amazon Simple Workflow Service .	4 de mayo de 2023
Actualización	Amazon SWF ahora ofrece una nueva experiencia de consola para gestionar los flujos de trabajo de SWF y sus acciones relacionadas con la ejecución. Para obtener más información, consulte los tutoriales sobre la consola de Amazon SWF .	12 de septiembre de 2022
Actualización	Se actualizó la Cuotas de ejecuciones de tareas sección para incluir Maximum tasks scheduled per second y la Métricas de Amazon SWF para CloudWatch página para incluir información sobre el uso de nombres de recursos que no son ASCII con. CloudWatch	12 de mayo de 2021
Nueva característica	Amazon Simple Workflow Service ahora es compatible con Amazon EventBridge. Para obtener más información, consulte: <ul style="list-style-type: none"> • EventBridge para Amazon SWF • EventBridge Guía del usuario 	18 de diciembre de 2020

Cambio	Descripción	Fecha de modificación
Nueva característica	<p>Amazon Simple Workflow Service admite los permisos de IAM mediante etiquetas. Para obtener más información, consulte lo siguiente.</p> <ul style="list-style-type: none"> • Etiquetas en Amazon SWF <ul style="list-style-type: none"> • Administración de etiquetas • Etiqueta las ejecuciones del flujo de trabajo • Controle el acceso a los dominios con etiquetas • TagResource • UntagResource • ListTagsForResource • RegisterDomain 	20 de junio de 2019
Nueva característica	Amazon Simple Workflow Service ahora está disponible en la región Europa (Estocolmo).	12 de diciembre de 2018
Actualización	Se ha mejorado el tema de Amazon Simple Workflow Service sobre CloudTrail la integración. Consulte Grabación de llamadas a la API con AWS CloudTrail .	7 de agosto de 2018
Actualización	Se ha añadido información sobre la nueva PendingTasks métrica de CloudWatch. Para obtener más información, consulte Métricas de Amazon SWF .	18 de junio de 2018
Actualización	Se ha mejorado el resaltado de la sintaxis en los ejemplos de código.	29 de marzo de 2018
Actualización	Se ha agregado un tema en el que se describen las opciones de los usuarios de Ruby Flow para migrar desde esa plataforma. Para obtener más información, consulte Opciones de migración para Ruby Flow .	9 de marzo de 2018

Cambio	Descripción	Fecha de modificación
Actualización	Se ha mejorado la navegación por el tema de conceptos avanzados. Consulte Conceptos de flujo de trabajo avanzados en Amazon SWF .	19 de febrero de 2018
Actualización	Se mejoró la documentación de CloudWatch las métricas al agregar información estadística válida. Consulte Métricas de Amazon SWF para CloudWatch .	4 de diciembre de 2017
Actualización	Se han cambiado los temas para mejorar la estructura de documentos. Se ha añadido nueva información acerca de Métricas de eventos de decisión y API .	9 de noviembre de 2017
Actualización	Se ha actualizado la sección Cuotas de Amazon SWF para incluir límites de limitación controlada para todas las regiones.	18 de octubre de 2017
Actualización	Se ha cambiado <code>task_list</code> a <code>workflowId</code> en Introducción a Amazon SWF para evitar confusiones con <code>activity_list</code> .	25 de julio de 2017
Actualización	Se han limpiado los ejemplos de código en toda esta guía.	5 de junio de 2017
Actualización	Se han simplificado y mejorado la organización y el contenido de esta guía.	19 de mayo de 2017
Actualización	Actualizaciones y correcciones de enlaces.	16 de mayo de 2017
Actualización	Actualizaciones y correcciones de enlaces.	1 de octubre de 2016
Soporte de tareas de Lambda	Puede especificar las tareas de Lambda, además de las tareas de actividad tradicionales en los flujos de trabajo. Para obtener más información, consulte AWS Lambda tareas en Amazon SWF .	21 de julio de 2015

Cambio	Descripción	Fecha de modificación
Compatibilidad para el establecimiento de prioridad de las tareas	Ahora, Amazon SWF incluye soporte para establecer la prioridad de las tareas de una lista de tareas, e intentará realizar las que tengan mayor prioridad antes que las que tengan menor prioridad. En Configuración de la prioridad de las tareas en Amazon SWF encontrará información acerca de cómo establecer la prioridad de las tareas para los flujos de trabajo y las actividades.	17 de diciembre de 2014
Actualización	Se ha añadido un tema nuevo que describe cómo registrar las llamadas a la API de Amazon SWF mediante CloudTrail: Grabación de llamadas a la API con AWS CloudTrail	8 de mayo de 2014
Actualización	Se han agregado dos temas nuevos relacionados con CloudWatch las métricas de Amazon SWF: Métricas de Amazon SWF para CloudWatch , que proporciona una lista y descripciones de las métricas compatibles, y Visualización de las métricas de Amazon SWF para usar el CloudWatch Consola de administración de AWS , que proporciona información sobre cómo ver las métricas y configurar alarmas con. Consola de administración de AWS	28 de abril de 2014
Actualización	Se ha añadido una nueva sección: Recursos e información de referencia adicionales para Amazon SWF . En esta sección, se proporciona información de referencia del servicio e información sobre documentación, ejemplos, código y otros recursos web adicionales para desarrolladores de Amazon SWF.	19 de marzo de 2014
Actualización	Se ha añadido un tutorial acerca del flujo de trabajo. Consulte Introducción a Amazon SWF .	25 de octubre de 2013
Actualización	Se han añadido un ejemplo de la AWS CLI e información de la misma .	26 de agosto de 2013

Cambio	Descripción	Fecha de modificación
Actualización	Actualizaciones y correcciones.	1 de agosto de 2013
Actualización	Se ha actualizado el documento para describir cómo utilizar IAM para el control de acceso.	22 de febrero de 2013
Versión inicial	Esta es la primera versión de la Guía para desarrolladores de Amazon Simple Workflow Service.	16 de octubre de 2012

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.