Guia do Desenvolvedor

Amazon Simple Workflow Service

Versão da API 2012-01-25

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Simple Workflow Service: Guia do Desenvolvedor

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é o Amazon SWF? ... 1
componentes do fluxo de trabalho ... 2
- componentes do fluxo de trabalho ... 2
- Executando seu fluxo de trabalho ... 4
- Configuração do seu ambiente de desenvolvimento ... 5
- Desenvolva com AWS SDKs ... 5
- Considere o AWS Flow Framework ... 6

Conceitos básicos ... 7
- Sobre o fluxo de trabalho ... 8
- Pré-requisitos ... 9
- Etapas do tutorial ... 9
- Parte 1: Usar o Amazon SWF com o SDK para Ruby ... 9
  - Inclua o AWS SDK para Ruby ... 10
  - Configurando a sessão AWS ... 10
  - Registrar um domínio Amazon SWF ... 12
  - Próximas etapas ... 13
- Parte 2: Implementar o fluxo de trabalho ... 13
  - Projetar o fluxo de trabalho ... 13
  - Configurar o código do fluxo de trabalho ... 14
  - Registrar o fluxo de trabalho ... 16
  - Fazer sondagens em busca de decisões ... 17
  - Iniciar a execução de fluxo de trabalho ... 21
  - Próximas etapas ... 23
- Parte 3: Implementar as atividades ... 23
  - Definir um tipo de atividade básica ... 23
  - Definindo GetContactActivity ... 25
  - Definindo SubscribeTopicActivity ... 28
  - Definindo WaitForConfirmationActivity ... 31
  - Definindo SendResultActivity ... 34
  - Próximas etapas ... 35
- Parte 4: Implementar o agente de sondagem de tarefas de atividades ... 35
- Executar o fluxo de trabalho ... 39
  - Para onde ir agora? ... 43

Trabalhando no console ... 44

Registro de um domínio ... 44
Registrando tipos de fluxo de trabalho ... 45
Registrando tipos de atividades ... 46
Iniciando um fluxo de trabalho ... 46
Para iniciar uma execução de fluxo de trabalho usando o console ... 47
Gerenciando execuções de fluxo de trabalho ... 47
Conceitos básicos ... 51
Criação de um fluxo de trabalho ... 52
Modelagem do fluxo de trabalho e suas atividades ... 53
Executando fluxos de trabalho ... 54
Histórico do fluxo de trabalho ... 54
Identificadores de objetos ... 59
Domínios ... 60
Atuadores ... 61
O que é um ator no Amazon SWF? ... 61
Iniciadores de fluxo de trabalho ... 62
Agentes de decisão ... 62
Operadores de atividade ... 64
Intercâmbio de dados entre atuadores ... 64
Tarefas ... 65
Listas de tarefas ... 66
Listas de tarefas de decisão ... 67
Listas de tarefas de atividade ... 67
Roteamento de tarefas ... 68
Fechamento da execução do fluxo de ... 68
Ciclo de vida de execução do fluxo de ... 69
Ciclo de vida da execução de fluxo de trabalho ... 70
Pesquisa de tarefas ... 76
Conceitos avançados ... 78
Versionamento ... 78
Sinais ... 79
Fluxos de trabalho infantis ... 81
Marcadores ... 83
Tags ... 84
Gerenciar tags ... 85
Marcar execuções de fluxo de trabalho ... 85

Controle o acesso aos domínios com tags ... 87
Escolha exclusiva ... 88
Temporizadores ... 91
Cancelamento de tarefas de atividade ... 91
Segurança ... 95
Proteção de dados ... 95
Criptografia ... 96
Gerenciamento de Identidade e Acesso ... 97
Público ... 98
Autenticar com identidades ... 99
Gerenciar o acesso usando políticas ... 102
Controle de acesso ... 105
Ações de políticas ... 105
Recursos de políticas ... 106
Chaves de condição de políticas ... 107
ACLs ... 108
ABAC ... 108
Credenciais temporárias ... 108
Permissões de entidade principal ... 109
Perfis de serviço ... 109
Perfis vinculados a serviço ... 110
Políticas baseadas em identidade ... 110
Políticas baseadas em recursos ... 111
Como o Amazon Simple Workflow Service funciona com o IAM ... 111
Exemplos de políticas baseadas em identidade ... 112
Princípios básicos ... 115
Políticas do IAM do Amazon SWF ... 117
Resumo de APIs ... 125
Políticas baseadas em tag ... 133
Endpoints da Amazon VPC ... 134
Solução de problemas ... 136
Registro e Monitoramento ... 138
Métricas do Amazon SWF para CloudWatch ... 139
Visualização das métricas do Amazon SWF ... 149
Gravando para CloudTrail ... 153
EventBridge para Amazon SWF ... 160

Usando Notificações de Usuários da AWS com o Amazon SWF 169
Validação de conformidade 170
Resiliência 170
Segurança da infraestrutura 171
Análise de configuração e vulnerabilidade 172
Usando o AWS CLI 173
Trabalhando com APIs 175
Como fazer solicitações HTTP 175
Conteúdo de cabeçalho HTTP 176
Conteúdo do corpo HTTP 178
Exemplo de solicitação e resposta JSON 178
Cálculo da assinatura HMAC-SHA 179
Lista de ações do Amazon SWF 182
Ações relacionadas a atividades 182
Ações relacionadas a agentes de decisão 183
Ações relacionadas a execuções de fluxo de trabalho 183
Ações relacionadas à administração 183
Ações de visibilidade 184
Registrar um domínio 185
Consulte também 186
Definindo valores de tempo limite 186
Cotas em valores de tempo limite 186
Tempos limite de execução de fluxo de trabalho e tarefas de decisão 186
Tempo limite de tarefas de atividade 187
Consulte também 188
Registrar um tipo de fluxo de trabalho 188
Consulte também 189
Registrar um tipo de atividade 189
Consulte também 189
Tarefas Lambda 190
Sobre AWS Lambda 190
Benefícios e limitações do uso de tarefas Lambda 190
Usar tarefas do Lambda em seus fluxos de trabalho 191
Desenvolver um operador de atividade 196
Sondar tarefas de atividade 197
Realizar a tarefa de atividade 198

Informar heartbeats de tarefas de atividade ... 198
Concluir ou falhar uma tarefa de atividade ... 199
Iniciar operadores de atividade ... 201
Desenvolvendo decisores ... 201
Definir a lógica de coordenação ... 203
Sondar tarefas de decisão ... 203
Aplicar a lógica de coordenação ... 205
Responder com decisões ... 206
Fechar uma execução de fluxo de trabalho ... 207
Iniciar agentes de decisão ... 209
Iniciando fluxos de trabalho ... 210
Definindo a prioridade da tarefa ... 211
Definindo a prioridade de tarefas para fluxos de trabalho ... 212
Definindo a prioridade de tarefas para atividades ... 214
Ações que retornam informações de prioridade de tarefa ... 215
Tratamento de erros ... 215
Erros de validação ... 216
Erros na execução de ações ou decisões ... 216
Tempos limite ... 216
Erros gerados pelo código do usuário ... 217
Erros relacionados ao encerramento de uma execução de fluxo de trabalho ... 217
Cotas ... 218
Cotas gerais de contas para o Amazon SWF ... 218
Cotas na execução do fluxo de trabalho ... 219
Cotas em execuções de tarefas ... 220
Cotas de controle de utilização do Amazon SWF ... 221
Controle de utilização de cotas para todas as regiões ... 221
cotas de decisão para todas as regiões ... 224
Cotas em nível de fluxo de trabalho ... 224
Solicitar um aumento de cota ... 225
Recursos adicionais ... 226
Tipos de tempo limite ... 226
Tempos limites em tarefas de fluxo de trabalho e decisão ... 227
Tempos limites em tarefas de atividade ... 228
Endpoints ... 229
Documentação adicional do ... 230

Amazon Simple Workflow Service API Reference 230
AWS Flow Framework Documentação 230
AWS Documentação do SDK 230
AWS CLI Documentação 232
Recursos da Web 233
Fórum do Amazon SWF 233
Perguntas frequentes sobre o Amazon SWF 233
Vídeos Amazon SWF 233
Opções do Ruby Flow 233
Continuar usando o Ruby Flow Framework 234
Migrar para o Java Flow Framework 234
Migrar para o Step Functions 235
Use a API do Amazon SWF diretamente 236
Histórico do documentos 237
..... ccxlii

O que é o Amazon Simple Workflow Service?

Com o Amazon Simple Workflow Service (Amazon SWF), você pode criar, executar e escalar trabalhos em segundo plano com etapas paralelas ou sequenciais. Você pode coordenar o trabalho entre componentes distribuídos e monitorar o estado das tarefas.

No Amazon SWF, uma tarefa representa uma unidade lógica de trabalho que é executada por um componente do seu aplicativo. A coordenação de tarefas inclui o gerenciamento de dependências entre tarefas, agendamento e simultaneidade no fluxo do seu aplicativo. Com o Amazon SWF, você pode controlar e coordenar tarefas sem se preocupar com complexidades subjacentes, como monitorar o progresso e manter o estado da tarefa.

Ao usar o Amazon SWF, você implementa trabalhadores para realizar tarefas. Os trabalhadores podem executar na infraestrutura de nuvem, como o Amazon Elastic Compute Cloud (Amazon EC2), ou em suas próprias instalações. Você pode criar tarefas de longa duração ou que podem falhar, expirar ou exigir reinicializações, ou que podem ser concluídas com throughput e latência variáveis. O Amazon SWF armazena tarefas e as atribui aos trabalhadores quando elas estão prontas, acompanha o progresso e mantém o estado, incluindo detalhes da conclusão da tarefa.

Para coordenar tarefas, você escreve um programa que obtém o estado mais recente da tarefa do Amazon SWF e usa esse estado para iniciar tarefas subsequentes. O Amazon SWF mantém o estado de execução de um aplicativo de forma durável, para que seu aplicativo seja resiliente a falhas de componentes individuais. Com o Amazon SWF, você pode criar, implantar, escalar e modificar componentes do aplicativo de forma independente.

> **Outros serviços AWS de fluxo de trabalho**
>
> Para a maioria dos casos de uso, recomendamos considerar suas necessidades AWS Step Functions de fluxo de trabalho e orquestração.
> Com o Step Functions, você pode criar fluxos de trabalho, também chamados de máquinas de estado, para criar aplicativos distribuídos, automatizar processos, orquestrar microsserviços e criar pipelines de dados e aprendizado de máquina. No console ou AWS kit de ferramentas do Step Functions no VS Code, você pode usar o Workflow Studio gráfico para visualizar, editar, testar e depurar o fluxo de trabalho do seu aplicativo.
> Para obter mais informações técnicas, consulte o Guia do AWS Step Functions desenvolvedor.

Desenvolvimento de componentes de fluxo de trabalho com o Amazon SWF

O desenvolvimento de aplicativos distribuídos exige coordenar muitos componentes e lidar com a latência e a falta de confiabilidade inerentes à comunicação remota.

Com o Amazon Simple Workflow Service (Amazon SWF), você pode desenvolver aplicativos assíncronos e distribuídos fornecendo um modelo de programação e uma infraestrutura para coordenar componentes distribuídos e manter seu estado de execução de forma confiável. Utilizando o Amazon SWF, você fica livre para se concentrar na criação dos aspectos da aplicação que a diferenciam.

Componentes de um fluxo de trabalho

Componentes de um fluxo de trabalhoO conceito fundamental no Amazon SWF é o fluxo de trabalho. Um fluxo de trabalho é um conjunto de atividades com um certo objetivo, juntamente com a lógica que coordena essas atividades. Por exemplo, um fluxo de trabalho pode receber um pedido de um cliente e realizar as ações necessárias para atender ao pedido.

Cada fluxo de trabalho é executado em um recurso chamado domínio, que controla o escopo do fluxo de trabalho. Uma conta AWS pode ter vários domínios, cada um dos quais pode conter vários fluxos de trabalho, mas os fluxos de trabalho em domínios diferentes não podem interagir.

Ao criar um fluxo de trabalho do Amazon SWF, você define cada uma das atividades necessárias. Em seguida, registre cada atividade no Amazon SWF como um tipo de atividade. Você fornecerá um nome, versão e valores de tempo limite. Por exemplo, um cliente pode ter a expectativa de que um pedido será enviado em até 24 horas.

No processo de realização do fluxo de trabalho, algumas atividades talvez precisem ser realizadas mais de uma vez, possivelmente com diferentes entradas. Por exemplo, em um fluxo de trabalho de pedido de cliente, você pode ter uma atividade que lida com itens comprados. Se o cliente comprar vários itens, essa atividade precisaria ser executada várias vezes. O Amazon SWF tem o conceito de uma tarefa de atividade que representa uma invocação de uma atividade. Em nosso exemplo, o processamento de cada item seria representado por uma única tarefa de atividade.

Um trabalhador ativo é um programa que recebe tarefas de atividade, as executa e fornece resultados. A tarefa pode, na verdade, ser executada por uma pessoa. Por exemplo, um analista

estatístico pode receber conjuntos de dados, analisar os dados e, em seguida, enviar de volta a análise.

As tarefas de atividade e os trabalhadores que as realizam podem ser executadas de forma síncrona ou assíncrona. Os trabalhadores podem trabalhar em um único local ou ser distribuídos em vários computadores, potencialmente em diferentes regiões geográficas. Diferentes operadores de atividade podem ser escritos em diferentes linguagens de programação e executados em diferentes sistemas operacionais. Por exemplo, um funcionário ativo pode estar executando em um servidor na Ásia, enquanto outro pode estar executando em um dispositivo móvel na América do Norte.

A lógica de coordenação em um fluxo de trabalho está contida em um programa de software chamado de agente de decisão. Um decisor agenda tarefas de atividade, fornece informações aos funcionários da atividade, processa eventos que chegam enquanto o fluxo de trabalho está em andamento e encerra (ou fecha) o fluxo de trabalho após o objetivo ter sido atingido.

A função do serviço Amazon SWF é funcionar como um hub central confiável por meio do qual os dados são trocados entre o agente de decisão, os operadores de atividade e outras entidades relevantes, como a pessoa que administra o fluxo de trabalho. O Amazon SWF também mantém o estado de cada execução do fluxo de trabalho, o que evita que a aplicação precise armazenar o estado de forma durável.

O agente de decisão direciona o fluxo de trabalho recebendo tarefas de decisão do Amazon SWF e respondendo de volta ao Amazon SWF com decisões. Uma decisão representa uma ação ou conjunto de ações, que são as próximas etapas no fluxo de trabalho. Uma decisão típica seria agendar uma tarefa de atividade. As decisões também podem ser usadas para atrasar tarefas com cronômetros, solicitar o cancelamento de tarefas em andamento e concluir fluxos de trabalho.

O mecanismo pelo qual os operadores de atividade e o agente de decisão recebem suas tarefas (tarefas de atividade e tarefas de decisão, respectivamente) é a sondagem do serviço Amazon SWF.

O Amazon SWF informa o agente de decisão sobre o estado do fluxo de trabalho, incluindo, em cada tarefa de decisão, uma cópia do histórico de execução do fluxo de trabalho atual. O histórico de execução de fluxo de trabalho é composto de eventos, em que um evento representa uma alteração significativa no estado da execução de fluxo de trabalho. Exemplos de eventos incluem a conclusão da tarefa, o tempo limite da tarefa ou a expiração de um cronômetro. O histórico é um registro completo, consistente e dominante do progresso do fluxo de trabalho.

Usos de controle de acesso do Amazon SWF AWS Identity and Access Management (IAM), para que você possa controlar o acesso aos AWS recursos. Por exemplo, você pode permitir que um usuário

acesse sua conta, mas apenas para executar determinados fluxos de trabalho em um domínio específico.

Executando seu fluxo de trabalho

Veja a seguir uma visão geral das etapas necessárias para desenvolver e executar um fluxo de trabalho no Amazon SWF:

1. Escreva agentes de atividade para realizar as etapas de processamento em seu fluxo de trabalho.
2. Escreva um decisor para lidar com a lógica de coordenação do seu fluxo de trabalho.
3. Registre suas atividades e seu fluxo de trabalho no Amazon SWF.

   Você pode executar essa etapa programaticamente ou usando o. AWS Management Console
4. Inicie seus operadores de atividade e o agente de decisão.

   Esses atores podem ser executados em qualquer dispositivo de computação que possa acessar um endpoint do Amazon SWF. Por exemplo, você pode usar instâncias de computação na nuvem, como Amazon Elastic Compute Cloud EC2 (Amazon); servidores em seu data center; ou até mesmo um dispositivo móvel, para hospedar um tomador de decisão ou funcionário da atividade. Uma vez iniciado, o agente de decisão e os operadores de atividade devem começar a sondar o Amazon SWF em busca de tarefas.
5. Inicie uma ou mais execuções do seu fluxo de trabalho.

   Você pode iniciar fluxos de trabalho programaticamente ou por meio do. AWS Management Console

   Cada execução é feita de forma independente, e você pode fornecer a cada uma seu próprio conjunto de dados de entrada. Quando uma execução é iniciada, o Amazon SWF agenda a tarefa de decisão inicial. Em resposta, seu tomador de decisão começa a gerar decisões que iniciam tarefas de atividade. A execução continua até que o agente de decisão tome a decisão de encerrá-la.
6. Visualize as execuções do fluxo de trabalho usando o. AWS Management Console

   Você pode filtrar e visualizar detalhes completos das execuções em execução e concluídas. Por exemplo, você pode selecionar uma execução aberta para ver quais tarefas foram concluídas e quais foram seus resultados.

Configuração do seu ambiente de desenvolvimento

Você tem a opção de desenvolver para o Amazon SWF em qualquer uma das linguagens de programação suportadas pelo. AWS Para desenvolvedores Java, o também AWS Flow Framework está disponível. Para obter mais informações, consulte o AWS Flow Frameworksite e consulte AWS Flow Framework o Guia do Desenvolvedor Java.

Para reduzir a latência e armazenar dados em um local que atenda aos seus requisitos, o Amazon SWF fornece endpoints em diferentes regiões.

Cada endpoint no Amazon SWF é completamente independente. Quaisquer domínios, fluxos de trabalho e atividades que você tenha registrado em uma região não compartilharão dados ou atributos com aqueles em outra região.

Quando você registra um domínio, fluxo de trabalho ou atividade do Amazon SWF, ele existe somente na região em que você o registrou. Por exemplo, você pode registrar um domínio nomeado `SWF-Flows-1` em duas regiões diferentes, mas elas não compartilharão dados ou atributos entre si — cada uma atuando como um domínio completamente independente.

Para obter uma lista dos endpoints do Amazon SWF, consulte Regiões e endpoints.

Desenvolva com AWS SDKs

O Amazon SWF é suportado pelo AWS SDKs para Java, .NET, Node.js, PHP, Python e Ruby, fornecendo uma maneira conveniente de usar a API HTTP do Amazon SWF na linguagem de programação de sua escolha.

Você pode desenvolver decisores, agentes de atividades ou iniciadores de fluxo de trabalho usando a API exposta por essas bibliotecas. Além disso, você pode usar operações de visibilidade por meio dessas bibliotecas para desenvolver suas próprias ferramentas de monitoramento e emissão de relatórios do Amazon SWF.

Para baixar ferramentas para desenvolver e gerenciar aplicativos em AWS, inclusive SDKs, acesse o Developer Center.

Para obter informações detalhadas sobre as operações do Amazon SWF em cada SDK, consulte a documentação de referência específica do idioma para o SDK.

Considere o AWS Flow Framework

AWS Flow Framework É um SDK aprimorado para escrever programas distribuídos e assíncronos que são executados como fluxos de trabalho no Amazon SWF. A estrutura está disponível para a linguagem de programação Java e fornece classes para escrever programas distribuídos complexos.

Com o AWS Flow Framework, você usa tipos pré-configurados para mapear a definição do seu fluxo de trabalho diretamente para os métodos do seu programa. O AWS Flow Framework suporta conceitos padrão orientados a objetos, como tratamento de erros baseado em exceções. Os programas escritos com o AWS Flow Framework podem ser criados, executados e depurados inteiramente no editor ou IDE de sua preferência. Para obter mais informações, consulte o AWS Flow Frameworksite e consulte AWS Flow Framework o Guia do Desenvolvedor Java.

Começando a usar o Amazon SWF

Você pode começar com o seguinte aplicativo de fluxo de trabalho do Amazon Simple Workflow Service, que consiste em um conjunto de quatro atividades que operam sequencialmente. O tutorial também aborda os seguintes tópicos:

- Definir opções de fluxo de trabalho e atividade padrão e de tempo de execução.
- Sondagem do Amazon SWF para tarefas de decisão e atividade.
- Passagem de dados entre as atividades e o fluxo de trabalho com o Amazon SWF.
- Aguardando tarefas humanas e relatando batimentos cardíacos para o Amazon SWF a partir de uma tarefa de atividade.
- Usar o Amazon SNS para criar um tópico, inscrever um usuário nele e publicar mensagens nos endpoints inscritos.

Você pode usar o Amazon SWF e o Amazon Simple Notification Service (Amazon SNS) juntos para emular um fluxo de trabalho de “tarefa humana”, no qual um trabalhador humano precisa realizar alguma ação e depois se comunicar com o Amazon SWF para iniciar a próxima atividade no fluxo de trabalho.

Como o Amazon SWF é um serviço da Web baseado em nuvem, a comunicação com o Amazon SWF pode se originar de qualquer lugar em que uma conexão com a Internet esteja disponível. Nesse caso, usaremos o Amazon SNS para nos comunicarmos com o usuário por e-mail, mensagem de texto SMS ou ambos.

Este tutorial usa o AWS SDK para Rubypara acessar o Amazon SWF e o Amazon SNS, mas há muitas opções de desenvolvimento disponíveis, incluindo a para Ruby, que facilita AWS Flow Framework a coordenação e a comunicação com o Amazon SWF.

> **Note**
>
> Este tutorial usa o AWS SDK para Ruby, mas recomendamos que você use o AWS Flow Framework para Java.

Tópicos

- Sobre o fluxo de trabalho

- Pré-requisitos
- Etapas do tutorial
- Tutorial do fluxo de trabalho de assinatura, parte 1: Usando o Amazon SWF com o AWS SDK para Ruby
- Tutorial de fluxo de trabalho de inscrição - Parte 2: Implementar o fluxo de trabalho
- Tutorial de fluxo de trabalho de inscrição - Parte 3: Implementar as atividades
- Tutorial de fluxo de trabalho de inscrição - Parte 4: Implementar o agente de sondagem de tarefas de atividades
- Tutorial de fluxo de trabalho de inscrição: Executar o fluxo de trabalho

Sobre o fluxo de trabalho

O fluxo de trabalho que desenvolveremos consiste em quatro etapas principais:

1. Obtenha um endereço de inscrição (e-mail ou SMS) do usuário.
2. Crie um tópico do SNS e inscreva os endpoints fornecidos nesse tópico.
3. Aguarde até que o usuário confirme a inscrição.
4. Se o usuário confirmar, publique uma mensagem de felicitações no tópico.

Essas etapas incluem atividades completamente automatizadas (etapas 2 e 4) e outras que exigem que o fluxo de trabalho aguarde até que um operador humano forneça alguns dados à atividade antes que o fluxo de trabalho possa avançar (etapas 1 e 3).

Cada etapa depende dos dados gerados pela etapa anterior (você deve ter um endpoint antes de inscrevê-lo em um tópico e deve ter uma assinatura de tópico para aguardar a confirmação, etc.) Este tutorial também abordará como fornecer resultados de atividades após a conclusão e como transmitir informações para uma tarefa que está sendo agendada. O Amazon SWF gerencia a coordenação e a entrega de informações entre as atividades e o fluxo de trabalho, e vice-versa.

Também estamos usando a entrada do teclado e o Amazon SNS para lidar com a comunicação entre o Amazon SWF e o ser humano que está fornecendo dados para o fluxo de trabalho. Na prática, você pode usar muitas técnicas diferentes para se comunicar com usuários humanos, mas o Amazon SNS oferece uma maneira muito fácil de usar e-mail ou mensagens de texto para notificar o usuário sobre eventos no fluxo de trabalho.

Pré-requisitos

Para acompanhar este tutorial, você precisará do seguinte:

- Conta da Amazon Web Services
- Intérprete Ruby
- AWS SDK para Ruby

Se já tiver essa configuração, você estará pronto para continuar. Se você não quiser executar o exemplo, ainda poderá seguir o tutorial — grande parte do conteúdo deste tutorial se aplica ao uso do Amazon SWF e do Amazon SNS, independentemente da opção de desenvolvimento escolhida.

Etapas do tutorial

Este tutorial está dividido nas seguintes etapas:

1. Tutorial do fluxo de trabalho de assinatura, parte 1: Usando o Amazon SWF com o AWS SDK para Ruby
2. Tutorial de fluxo de trabalho de inscrição - Parte 2: Implementar o fluxo de trabalho
3. Tutorial de fluxo de trabalho de inscrição - Parte 3: Implementar as atividades
4. Tutorial de fluxo de trabalho de inscrição - Parte 4: Implementar o agente de sondagem de tarefas de atividades
5. Tutorial de fluxo de trabalho de inscrição: Executar o fluxo de trabalho

Tutorial do fluxo de trabalho de assinatura, parte 1: Usando o Amazon SWF com o AWS SDK para Ruby

Tópicos

- Inclua o AWS SDK para Ruby
- Configurando a sessão AWS
- Registrar um domínio Amazon SWF
- Próximas etapas

Inclua o AWS SDK para Ruby

Comece criando um arquivo chamado `utils.rb`. O código nesse arquivo obterá ou criará, se necessário, o domínio Amazon SWF usado pelo fluxo de trabalho e pelo código de atividades e fornecerá um local para colocar o código que é comum a todas as nossas classes.

Primeiro, precisamos incluir a biblioteca `aws-sdk-v1` em nosso código, para que possamos usar os recursos fornecidos pelo SDK para Ruby.

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

Isso nos dá acesso ao AWS namespace, que fornece a capacidade de definir valores globais relacionados à sessão, como suas AWS credenciais e região, e também fornece acesso ao serviço. AWS APIs

Configurando a sessão AWS

Configuraremos a AWS sessão definindo nossas AWS credenciais (que são necessárias para acessar os AWS serviços) e a AWS região a ser usada.

Há várias maneiras de definir AWS credenciais no AWS SDK para Ruby: definindo-as em variáveis de ambiente (_KEY_ID AWS_SECRET e _ACCESS_KEYAWS_ACCESS) ou definindo-as com. `AWS.config` Usaremos o último método, carregando-os a partir de um arquivo de configuração YAML, `aws-config.txt`, que tem a seguinte aparência.

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

Crie esse arquivo agora, substituindo as sequências que começam com REPLACE_WITH_ pelo ID da chave de acesso e pela chave de AWS acesso secreta. Para obter informações sobre suas chaves de AWS acesso, consulte Como faço para obter credenciais de segurança? na Referência geral da Amazon Web Services.

Também precisamos definir a AWS região a ser usada. Como usaremos o Short Message Service (SMS) para enviar mensagens de texto para o telefone do usuário com o Amazon SNS, precisamos ter certeza de que estamos usando a região atendida pelo Amazon SNS. Consulte as regiões e os países compatíveis no Guia do desenvolvedor do Amazon Simple Notification Service.

> **Note**
>
> Se você não tiver acesso a us-east-1, ou não se importar em executar a demonstração com mensagens SMS habilitadas, fique à vontade para usar qualquer região desejada. É possível remover a funcionalidade de SMS da amostra e usar o e-mail como o único endpoint para assinar o tópico do Amazon SNS.
> Para obter mais informações sobre o envio de mensagens SMS, consulte Envio e recebimento de notificações SMS usando o Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Agora, adicionaremos algum código ao `utils.rb` para carregar o arquivo de configuração, obter as credenciais do usuário e, em seguida, fornecer as credenciais e a região à `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
```

Registrar um domínio Amazon SWF

Para usar o Amazon SWF, você precisa configurar um domínio: uma entidade nomeada que manterá seus fluxos de trabalho e atividades. Você pode ter vários domínios do Amazon SWF registrados, mas todos eles devem ter nomes exclusivos em sua AWS conta, e os fluxos de trabalho não podem interagir entre domínios: todos os fluxos de trabalho e atividades do seu aplicativo devem estar no mesmo domínio para interagir uns com os outros.

Como usaremos o mesmo domínio em todo o nosso aplicativo, criaremos uma função em call*`init_domain`*, que recuperará o domínio Amazon SWF *`utils.rb`* chamado DomainSWFSample.

Depois de ter registrado um domínio, você poderá reutilizá-lo para muitas execuções de fluxo de trabalho. No entanto, é um erro tentar registrar um domínio que já existe e, portanto, nosso código primeiro verificará se o domínio existe e usará o domínio existente se ele for encontrado. Se o domínio não puder ser encontrado, nós o criaremos.

Para trabalhar com domínios Amazon SWF no SDK para Ruby, AWS::Simpleuse Workflow.domains, que retorna DomainCollectionum que pode ser usado tanto para enumerar quanto para registrar domínios:

- Para verificar se um domínio já está registrado, consulte a lista fornecida por AWS::Simpleworkflow.domains.registered.
- Para registrar um novo domínio, use AWS::Simpleworkflow.domains.register.

Aqui está o código para `init_domain` em `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
```

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

  return domain
end
```

Próximas etapas

Em seguida, você criará o fluxo de trabalho e o código do iniciador em Tutorial de fluxo de trabalho de inscrição - Parte 2: Implementar o fluxo de trabalho.

Tutorial de fluxo de trabalho de inscrição - Parte 2: Implementar o fluxo de trabalho

Até agora, nosso código está bastante genérico. Esta é a parte em que começamos a definir realmente o que nosso fluxo de trabalho fará e quais atividades precisaremos realizar para implementá-lo.

Tópicos

- Projetar o fluxo de trabalho
- Configurar o código do fluxo de trabalho
- Registrar o fluxo de trabalho
- Fazer sondagens em busca de decisões
- Iniciar a execução de fluxo de trabalho
- Próximas etapas

Projetar o fluxo de trabalho

Se você se lembra, a ideia inicial para esse fluxo de trabalho consistia nas seguintes etapas:

1. Obtenha um endereço de inscrição (e-mail ou SMS) do usuário.
2. Crie um tópico do SNS e inscreva os endpoints fornecidos nesse tópico.

3. Aguarde até que o usuário confirme a inscrição.
4. Se o usuário confirmar, publique uma mensagem de felicitações no tópico.

Podemos pensar em cada etapa do nosso fluxo de trabalho como uma atividade que ele deve realizar. Nosso fluxo de trabalho é responsável por agendar cada atividade no momento apropriado e por coordenar a transferência de dados entre essas atividades.

Para este fluxo de trabalho, criaremos uma atividade separada para cada uma dessas etapas, fornecendo a elas nomes descritivos:

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

Essas atividades serão executadas em ordem, e os dados de cada uma serão usados na etapa subsequente.

Podemos projetar nossa aplicação de modo que todo o código estivesse em um único arquivo de origem, mas isso é contrário à forma como o Amazon SWF foi projetado. Ele foi concebido para fluxos de trabalho cujo escopo pode abranger toda a Internet. Por isso, vamos dividir o aplicativo em dois executáveis separados:

- `swf_sns_workflow.rb` - Contém o fluxo de trabalho e o iniciador do fluxo de trabalho.
- `swf_sns_activities.rb` - Contém as atividades e o iniciador das atividades.

As implementações de fluxo de trabalho e atividades podem ser executadas em janelas separadas, computadores separados ou até mesmo em diferentes partes do mundo. Como o Amazon SWF controla os detalhes de seus fluxos de trabalho e atividades, seu fluxo de trabalho pode coordenar o agendamento e a transferência de dados de suas atividades, independentemente de onde elas estejam sendo executadas.

Configurar o código do fluxo de trabalho

Começaremos criando um arquivo chamado `swf_sns_workflow.rb`. Nesse arquivo, declare uma classe chamada SampleWorkflow. Aqui está a declaração da classe e seu construtor, o 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
```

Como você pode ver, estamos mantendo os seguintes dados de instância de classe:

- `domain` - O nome do domínio recuperado de `init_domain` em `utils.rb`.
- `workflowId` - A lista de tarefas transmitidas para `initialize`.
- `activity_list` - A lista de atividades, que tem os nomes e as versões das atividades que vamos executar.

O nome do domínio, o nome da atividade e a versão da atividade são suficientes para que o Amazon SWF identifique positivamente um tipo de atividade, portanto, esses são todos os dados que precisamos manter sobre nossas atividades para programá-las.

A lista de tarefas será usada pelo código do agente de decisão do fluxo de trabalho para sondar tarefas de decisão e agendar atividades.

No final dessa função, chamaremos um método que ainda não definimos: `register_workflow`. Definiremos esse método a seguir.

Registrar o fluxo de trabalho

Para usar um tipo de fluxo de trabalho, primeiro devemos registrá-lo. Como um tipo de atividade, um tipo de fluxo de trabalho é identificado por seu domínio, nome e versão. Além disso, como domínios e tipos de atividades, você não pode repetir o registro de um tipo de fluxo de trabalho existente. Se precisar mudar algo sobre um tipo de fluxo de trabalho, você deverá fornecer a ele uma nova versão, o que essencialmente cria um novo tipo.

Aqui está o código para `register_workflow`, que é usado para recuperar o tipo de fluxo de trabalho existente que registramos em uma execução anterior ou para registrar o fluxo de trabalho caso ele ainda não tenha sido 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

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

Primeiro, verificamos se o nome e a versão do fluxo de trabalho já estão registrados, iterando pela coleção workflow_types do domínio. Se encontrarmos uma correspondência, usaremos o tipo de fluxo de trabalho que já estava registrado.

Se não encontrarmos uma correspondência, um novo tipo de fluxo de trabalho será registrado (chamando register na mesma `workflow_types` coleção em que estávamos pesquisando o fluxo de trabalho) com o nome swf-sns-workflow ", versão '1' e as seguintes opções.

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

As opções transmitidas durante o registro são usadas para definir o comportamento padrão do nosso tipo de fluxo de trabalho e, portanto, não precisamos definir esses valores cada vez que iniciamos uma nova execução de fluxo de trabalho.

Aqui, apenas definimos alguns valores de tempo limite: o tempo máximo desde o início de uma tarefa até o seu encerramento (uma hora) e o tempo máximo para a conclusão da execução de fluxo de trabalho (24 horas). Se qualquer um desses tempos for excedido, a tarefa ou o fluxo de trabalho expirará.

Para obter mais informações sobre valores de tempo limite, consulte Tipos de tempo limite do Amazon SWF .

Fazer sondagens em busca de decisões

No centro de cada execução de fluxo de trabalho, há um agente de decisão. A responsabilidade do agente de decisão é administrar a execução do próprio fluxo de trabalho. O agente de decisão recebe tarefas de decisão e responde a elas, agendando novas atividades, cancelando e reiniciando atividades ou definindo o estado da execução de fluxo de trabalho como concluída, cancelada ou com falha.

O agente de decisão usa o nome da lista de tarefas da execução de fluxo de trabalho para receber tarefas de decisão a serem respondidas. Para sondar tarefas de decisão, chame poll na coleção decision_tasks do domínio para percorrer as tarefas de decisão disponíveis. Em seguida, você pode verificar se há novos eventos na tarefa de decisão, iterando sobre sua coleção new_events.

Os eventos retornados são AWS::SimpleWorkflow::HistoryEventobjetos, e você pode obter o tipo do evento usando o membro event_type do evento retornado. Para obter uma lista e uma descrição dos tipos de eventos históricos, consulte HistoryEventa Referência da API do Amazon Simple Workflow Service.

Veja a seguir o início da lógica do agente de sondagem de tarefas de decisão. Um novo método em nossa classe de fluxo de trabalho chamou `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
```

Agora, vamos ramificar a execução do nosso agente de decisão com base no `event_type` recebido. O primeiro que provavelmente receberemos é WorkflowExecutionStarted. Quando esse evento é recebido, significa que o Amazon SWF está sinalizando para o seu agente de decisão que ele deve iniciar a execução do fluxo de trabalho. Começaremos agendando a primeira atividade, chamando schedule_activity_task na tarefa que recebemos durante a sondagem.

Transmitiremos a primeira atividade que declaramos em nossa lista de atividades, que, devido ao fato de termos invertido a lista para que pudéssemos usá-la como uma pilha, ocupa a posição `last` na lista. As “atividades” que definimos são apenas mapas que consistem em um nome e um número de versão, mas isso é tudo o que o Amazon SWF precisa para identificar a atividade para agendamento, supondo que a atividade já tenha sido registrada.

```
        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" } )
```

Quando agendamos uma atividade, o Amazon SWF envia uma tarefa de atividade para a lista de tarefas de atividade que passamos durante o agendamento, sinalizando o início da tarefa. Lideremos com tarefas da atividade em Tutorial de fluxo de trabalho de inscrição - Parte 3: Implementar as atividades, mas vale a pena notar que não executamos essa tarefa aqui. Apenas informamos ao Amazon SWF que ele deve ser agendado.

A próxima atividade que precisaremos abordar é o ActivityTaskCompletedevento, que ocorre quando o Amazon SWF recebe uma resposta de atividade concluída de uma tarefa de atividade.

```
          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
```

Como estamos executando nossas tarefas de forma linear e apenas uma atividade está sendo executada ao mesmo tempo, aproveitaremos esta oportunidade para retirar a tarefa concluída da `activity_list` pilha. Se isso resultar em uma lista vazia, saberemos que nosso fluxo de trabalho está concluído. Nesse caso, sinalizamos ao Amazon SWF que nosso fluxo de trabalho está concluído chamando complete_workflow_execution na tarefa.

No caso de a lista ainda ter entradas, agendaremos a próxima atividade da lista (novamente, na última posição). Desta vez, no entanto, verificaremos se a atividade anterior retornou algum dado de resultado para o Amazon SWF após a conclusão, que é fornecido ao fluxo de trabalho nos atributos do evento, na chave opcional `result`. Se a atividade tiver gerado um resultado, transmitiremos esse resultado como a opção `input` à próxima atividade agendada, juntamente com a lista de tarefas de atividade.

Ao recuperar os valores `result` de atividades concluídas e ao definir os valores `input` de atividades agendadas, podemos transmitir dados de uma atividade para a próxima ou podemos usar os dados de uma atividade para alterar o comportamento no nosso agente de decisão com base nos resultados de uma atividade.

Para os fins deste tutorial, esses dois tipos de eventos são os mais importantes na definição do comportamento do nosso fluxo de trabalho. No entanto, uma atividade pode gerar outros eventos além de ActivityTaskCompleted. Encerraremos nosso código decisor fornecendo um código manipulador de demonstração para os ActivityTaskFailedeventos ActivityTaskTimedOute e para o WorkflowExecutionCompletedevento, que será gerado quando o Amazon SWF processar `complete_workflow_execution` a chamada que fazemos quando ficamos sem atividades para executar.

```
          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
```

Iniciar a execução de fluxo de trabalho

Antes de qualquer tarefa de decisão ser gerada para sondagem pelo fluxo de trabalho, precisamos iniciar a execução de fluxo de trabalho.

Para iniciar a execução do fluxo de trabalho, chame start_execution no tipo de fluxo de trabalho registrado (). AWS::SimpleWorkflow::WorkflowType Definiremos um pequeno wrapper em torno disso para usar o membro da instância `workflow_type` que recuperamos no construtor da classe.

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

Assim que o fluxo de trabalho estiver em execução, eventos de decisão começarão a aparecer na lista de tarefas do fluxo de trabalho, que é transmitida como uma opção de execução de fluxo de trabalho em start_execution.

Ao contrário das opções que são fornecidas quando o tipo de fluxo de trabalho é registrado, as opções que são transmitidas ao `start_execution` não são consideradas parte do tipo de fluxo de trabalho. Você pode alterá-las para cada execução de fluxo de trabalho, sem alterar a versão do fluxo de trabalho.

Como gostaríamos que o fluxo de trabalho começasse a ser executado quando executamos o arquivo, adicione um código que instancie a classe e, em seguida, chame o `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 conflitos de nomeação de lista de tarefas, usaremos `SecureRandom.uuid` para gerar um UUID aleatório que podemos usar como o nome da lista de tarefas, garantindo que um nome de lista de tarefas diferente seja usado para cada execução de fluxo de trabalho.

> **Note**
>
> Listas de tarefas são usadas para registrar eventos sobre uma execução de fluxo de trabalho. Por isso, se você usar a mesma lista de tarefa para várias execuções do mesmo tipo de fluxo de trabalho, poderá obter eventos que foram gerados durante uma execução anterior, especialmente se essas execuções ocorrerem sucessivamente em um curto intervalo de tempo, o que costuma ser o caso em experimentos com um novo código ou em execuções de testes.

Para evitar o problema de ter que lidar com artefatos de execuções anteriores, podemos usar uma nova lista de tarefas para cada execução, especificando-a ao iniciarmos a execução do fluxo de trabalho.

Há também um pouco de código aqui para fornecer instruções para a pessoa que o está executando (provavelmente você) e fornecer a versão de "atividade" da lista de tarefas. O agente de decisão usa

esse nome de lista de tarefas para agendar atividades para o fluxo de trabalho, e a implementação das atividades fará escutas por eventos de atividade nesse nome de lista de tarefas para saber quando iniciar as atividades agendadas e para fornecer atualizações sobre a execução das atividades.

O código também espera que o usuário comece a executar o iniciador de atividades antes de iniciar a execução do fluxo de trabalho. Dessa forma, o iniciador de atividades estará pronto para responder quando as tarefas de atividade começarem a aparecer na lista de tarefas fornecidas.

Próximas etapas

Você implementou o fluxo de trabalho. Em seguida, você definirá as atividades e um iniciador de atividades, em Tutorial de fluxo de trabalho de inscrição - Parte 3: Implementar as atividades.

Tutorial de fluxo de trabalho de inscrição - Parte 3: Implementar as atividades

Agora, implementaremos cada uma das atividades no nosso fluxo de trabalho, começando com uma classe base que fornece alguns recursos comuns para o código de atividade.

Tópicos

- Definir um tipo de atividade básica
- Definindo GetContactActivity
- Definindo SubscribeTopicActivity
- Definindo WaitForConfirmationActivity
- Definindo SendResultActivity
- Próximas etapas

Definir um tipo de atividade básica

Ao projetar o fluxo de trabalho, identificamos as seguintes atividades:

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

Implementaremos cada uma dessas atividades agora. Como nossas atividades compartilharão alguns recursos, vamos fazer um pequeno trabalho de base e criar um código comum que eles possam compartilhar. Vamos chamá-lo BasicActivitye defini-lo em um novo arquivo chamado`basic_activity.rb`.

Como acontece com os outros arquivos de origem, incluiremos `utils.rb` para acessar a função `init_domain` e configurar o domínio da amostra.

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

Em seguida, declararemos a classe de atividade básica e alguns dados comuns que serão de nosso interesse para cada atividade. Salvaremos a AWS::SimpleWorkflow::ActivityTypeinstância, o nome e os resultados da atividade nos atributos da classe.

```
class BasicActivity

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

Esses atributos acessam dados de instância definidos no método `initialize` da classe, que recebe um nome de atividade e uma versão opcional e um mapa de opções a serem usados ao registrar a atividade com o 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
```

Como acontece com o registro do tipo de fluxo de trabalho, se um tipo de atividade já estiver registrado, poderemos recuperá-lo observando a coleção activity_types do domínio. Se a atividade não puder ser encontrada, ela será registrada.

Além disso, como em tipos de fluxo de trabalho, você pode definir opções padrão que são armazenadas com seu tipo de atividade quando você o registra.

A última coisa que nossa atividade básica obtém é uma maneira consistente de executá-la. Definiremos um método `do_activity` que usa uma tarefa de atividade. Conforme mostrado, podemos usar a tarefa de atividade transmitida para receber dados através de seu atributo de instância `input`.

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

Isso encerra a BasicActivityaula. Agora, vamos usá-la para tornar a definição das nossas atividades simples e consistente.

Definindo GetContactActivity

A primeira atividade que é executada durante a execução de um fluxo de trabalho é `get_contact_activity`, que recupera as informações de assinatura do tópico do Amazon SNS do usuário.

Crie um novo arquivo chamado`get_contact_activity.rb`, e exija ambos`yaml`, que usaremos para preparar uma string para passar para o Amazon SWF e`basic_activity.rb`, que usaremos como base para essa GetContactActivityclasse.

```
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 inserimos o código de registro da atividade BasicActivity, o `initialize` método GetContactActivityé bem simples. Simplesmente chamamos o construtor da classe base com o nome da atividade, `get_contact_activity`. Isso é tudo o que é necessário para registrar nossa atividade.

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

Agora, vamos definir o método do`_activity`, que solicita o e-mail e/ou número de telefone do usuário.

```
    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

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

No final de `do_activity`, usamos o e-mail e o número de telefone recuperados do usuário, colocamos esses dados em um mapa e, em seguida, usamos `to_yaml` para converter o mapa inteiro em uma string YAML. Há um motivo importante para isso: todos os resultados que você passar para o Amazon SWF ao concluir uma atividade devem ser apenas dados de string de caracteres. A habilidade do Ruby de converter objetos facilmente em strings YAML e depois novamente em objetos é, felizmente, bem adaptada para esse propósito.

Esse é o fim da implementação de `get_contact_activity`. Esses dados serão usados em seguida na implementação de `subscribe_topic_activity`.

Definindo SubscribeTopicActivity

Agora, vamos nos aprofundar no Amazon SNS e criar uma atividade que use as informações geradas por `get_contact_activity` para inscrever o usuário em um tópico do Amazon SNS.

Crie um novo arquivo chamado `subscribe_topic_activity.rb`, adicione os mesmos requisitos que usamos para `get_contact_activity`, declare sua classe e forneça seu 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
```

Agora que já temos o código para configurar e registrar a atividade, adicionaremos algum código para criar um tópico do Amazon SNS. Para fazer isso, usaremos o método create_topic do AWS::SNS::Clientobjeto.

Adicione o método `create_topic` à sua classe, que recebe um objeto de cliente Amazon SNS passado.

```
  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
```

Depois de termos o Amazon Resource Name (ARN) do tópico, podemos usá-lo com o método set_topic_attributes do cliente Amazon SNS para definir o tópico DisplayName, que é necessário para enviar mensagens SMS com o Amazon SNS.

Por fim, definiremos o método `do_activity`. Começaremos coletando quaisquer dados que tenham sido transmitidos por meio da opção `input` quando a atividade foi agendada. Conforme mencionado anteriormente, isso deve ser transmitido como uma string, que nós criamos usando `to_yaml`. Ao recuperá-lo, usaremos `YAML.load` para transformar os dados em objetos Ruby.

Este é o início de `do_activity`, no qual recuperamos os dados 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))
```

Se não recebermos nenhuma entrada, não haverá muito a fazer, então vamos simplesmente marcar a atividade como falha.

No entanto, supondo que tudo esteja bem, continuaremos preenchendo nosso `do_activity` método, obteremos um cliente Amazon SNS com AWS SDK para Ruby o e o passaremos para `create_topic` nosso método para criar o tópico do 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
```

Há algumas coisas que merecem destaque aqui:

- Usamos `AWS.config.with` para definir a região do nosso cliente Amazon SNS. Como queremos enviar mensagens SMS, usamos a região habilitada para SMS que declaramos em `utils.rb`.
- Salvamos o ARN do tópico em nosso mapa `activity_data`. Isso faz parte dos dados que serão transmitidos à próxima atividade no nosso fluxo de trabalho.

Por fim, essa atividade inscreve o usuário no tópico do Amazon SNS, usando os endpoints passados (e-mail e SMS). Não exigimos que o usuário insira ambos os endpoints, mas precisamos de pelo menos um.

```
# 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 usa o tópico ARN, o protocolo (que, de forma inteligente, disfarçamos como a chave do mapa para `activity_data` o endpoint correspondente).

Por fim, reempacotamos as informações para a próxima atividade no formato YAML, para que possamos enviá-las de volta ao 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
```

Isso completa a implementação do `subscribe_topic_activity`. Em seguida, definiremos `wait_for_confirmation_activity`.

Definindo WaitForConfirmationActivity

Depois que um usuário se inscrever em um tópico do Amazon SNS, ele ainda precisará confirmar a solicitação de inscrição. Nesse caso, aguardaremos que o usuário confirme por e-mail ou mensagem SMS.

A atividade que aguarda a confirmação da inscrição pelo usuário é chamada de `wait_for_confirmation_activity`, e nós a definiremos aqui. Para começar, crie um novo arquivo chamado `wait_for_confirmation_activity.rb` e configure-o como fizemos nas atividades 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
```

Em seguida, vamos começar a definir o método `do_activity` e a recuperar todos os dados de entrada em uma variável local chamada de `subscription_data`.

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

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

Agora que temos o ARN do tópico, podemos recuperar o tópico criando uma nova instância do AWS::SNS::Topice passando o ARN para ela.

```
      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
```

Agora, verificamos o tópico para ver se o usuário confirmou a assinatura usando um dos endpoints. Só exigiremos que um endpoint tenha sido confirmado para considerar a atividade como bem-sucedida.

Um tópico do Amazon SNS mantém uma lista das assinaturas desse tópico, e podemos verificar se o usuário confirmou ou não uma assinatura específica, verificando se o ARN da assinatura está definido como algo diferente de `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
```

Se obtivermos um ARN para a assinatura, vamos salvá-lo nos dados do resultado da atividade, convertê-lo em YAML e retornar "true" de `do_activity`, que indica que a atividade foi concluída com êxito.

Como esperar pela confirmação de uma assinatura pode demorar um pouco, ocasionalmente solicitamos `record_heartbeat` a tarefa da atividade. Isso sinaliza para o Amazon SWF que a atividade ainda está sendo processada e também pode ser usado para fornecer atualizações sobre o progresso da atividade (se você estiver fazendo algo, como processar arquivos, para o qual possa relatar o progresso).

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

Isso finaliza nosso loop `while`. Se, de alguma forma, sairmos do loop sem sucesso, informaremos a falha e terminamos o 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
```

Isso acaba com a implementação de `wait_for_confirmation_activity`. Temos apenas mais uma atividade para definir: `send_result_activity`.

Definindo SendResultActivity

Se o fluxo de trabalho tiver progredido até aqui, teremos inscrito com êxito o usuário em um tópico do Amazon SNS e o usuário terá confirmado a inscrição.

Nossa última atividade, `send_result_activity`, envia ao usuário uma confirmação da inscrição bem-sucedida no tópico, usando o tópico no qual o usuário se inscreveu e o endpoint com o qual o usuário confirmou a assinatura.

Crie um novo arquivo chamado `send_result_activity.rb` e configure-o como configuramos todas as atividades até agora.

```
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
```

Nosso `do_activity` método também começa da mesma forma, obtendo os dados de entrada do fluxo de trabalho, convertendo-os do YAML e usando o tópico ARN para criar uma instância. AWS::SNS::Topic

```
  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
```

Quando tivermos o tópico, publicaremos uma mensagem nele (e também o ecoaremos na tela).

```
      @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
```

A publicação em um tópico do Amazon SNS envia a mensagem que você forneceu a todos os endpoints inscritos e confirmados que existem para esse tópico. Portanto, se o usuário confirmar com um e-mail e também com um número de SMS, ele receberá duas mensagens de confirmação, uma em cada endpoint.

Próximas etapas

Isso completa a implementação de `send_result_activity`. Agora, você combinará todas essas atividades em um aplicativo de atividade que lida com as tarefas de atividades e que pode iniciar atividades em resposta, em Tutorial de fluxo de trabalho de inscrição - Parte 4: Implementar o agente de sondagem de tarefas de atividades.

Tutorial de fluxo de trabalho de inscrição - Parte 4: Implementar o agente de sondagem de tarefas de atividades

No Amazon SWF, as tarefas de atividade para uma execução de fluxo de trabalho em andamento aparecem na lista de tarefas de atividade, que é fornecida quando você agenda uma atividade no fluxo de trabalho.

Implementaremos um poller de atividade básico para lidar com essas tarefas em nosso fluxo de trabalho e o usaremos para iniciar nossas atividades quando o Amazon SWF colocar uma tarefa na lista de tarefas de atividade para iniciar a atividade.

Para começar, crie um novo arquivo chamado `swf_sns_activities.rb`. Vamos usá-lo para:

- Instanciar as classes de atividades que criamos.
- Registre cada atividade no Amazon SWF.
- Sondar atividades e chamar `do_activity` para cada atividade quando seu nome aparecer na lista de tarefas de atividade.

Em `swf_sns_activities.rb`, adicione as seguintes instruções para exigir cada uma das classes de atividades 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'
```

Agora, criaremos a classe e forneceremos um código de inicialização.

```
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
```

Além de salvar o domínio e a lista de tarefas transmitidos, esse código instancia cada uma das classes de atividades que criamos. Como cada classe registra sua atividade associada (consulte `basic_activity.rb` se precisar revisar esse código), isso é suficiente para que o Amazon SWF saiba sobre todas as atividades que serão executadas.

Cada atividade instanciada será armazenada em um mapa usando o nome da atividade como chave (por exemplo, `get_contact_activity`), para que possamos pesquisá-la com facilidade no código do agente de sondagem de atividades, que definiremos a seguir.

Crie um novo método chamado `poll_for_activities` e chame poll nas activity_tasks mantidas pelo domínio para obter tarefas de atividades.

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

Podemos obter o nome da atividade do membro activity_type da tarefa. Em seguida, usaremos o nome da atividade associado a essa tarefa para pesquisar a classe na qual executar `do_activity`, transmitindo a ela a tarefa (que inclui quaisquer dados de entrada que devem ser transferidos para a atividade).

```
    # 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
```

O código apenas aguarda a conclusão do `do_activity` e, em seguida, chama concluído! ou falha! na tarefa com base no código de retorno.

> **Note**
>
> Esse código sai do poller assim que a atividade final é lançada, porque ele completou sua missão e lançou todas as atividades. Em seu próprio código Amazon SWF, se suas atividades puderem ser executadas novamente, talvez você queira manter o poller de atividades em execução indefinidamente.

Esse é o fim do código da nossa ActivitiesPollerclasse, mas adicionaremos um pouco mais de código no final do arquivo para permitir que o usuário o execute a partir da linha de comando.

```
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
```

Se o usuário executar o arquivo a partir da linha de comando (transmitindo a ele uma lista de tarefas de atividade como o primeiro argumento), esse código instanciará a classe poller e iniciará a sondagem em busca de atividades. Após a conclusão do agente de sondagem (depois que ele tiver iniciado a atividade final), simplesmente imprimimos uma mensagem e saímos.

Isso é tudo para o agente de sondagem de atividades. Tudo o que resta a fazer é executar o código e ver como ele funciona, em Tutorial de fluxo de trabalho de inscrição: Executar o fluxo de trabalho.

Tutorial de fluxo de trabalho de inscrição: Executar o fluxo de trabalho

Agora que você concluiu a implementação do seu fluxo de trabalho, das atividades e dos agentes de sondagem de fluxo de trabalho e atividade, está pronto para executá-lo.

Se ainda não tiver feito isso, você precisará fornecer suas chaves de AWS acesso no `aws-config.txt` arquivo, como Configurando a sessão AWS na Parte 1 do tutorial.

Agora, acesse sua linha de comando e mude para o diretório onde os arquivos de origem do tutorial estão localizados. Você terá os seguintes arquivos:

```
.
|-- 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
```

Agora, inicie o fluxo de trabalho com o seguinte comando.

```
ruby swf_sns_workflow.rb
```

Isso iniciará o fluxo de trabalho e imprimirá uma mensagem com uma linha que você pode copiar e colar em uma janela de linha de comando separada (ou até mesmo em outro computador, se tiver copiado os arquivos de origem do tutorial nele).

```
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...
```

O código do fluxo de trabalho aguardará pacientemente até que você inicie o agente de sondagem de atividade em uma janela separada.

Abra uma nova janela de linha de comando, mude para o diretório em que os arquivos de origem estão novamente localizados e use o comando fornecido pelo arquivo `swf_sns_workflow.rb` para iniciar o agente de sondagem de atividades. Por exemplo, se você recebeu a saída anterior, digite (ou cole) o seguinte.

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

Depois de começar a executar seu agente de sondagem de atividades, ele começará a exibir informações sobre o registro de atividades.

```
** 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
```

Agora, você pode retornar à sua janela de linha de comando original e pressionar Return para iniciar a execução do fluxo de trabalho. Ele registrará o fluxo de trabalho e agendará a primeira atividade.

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

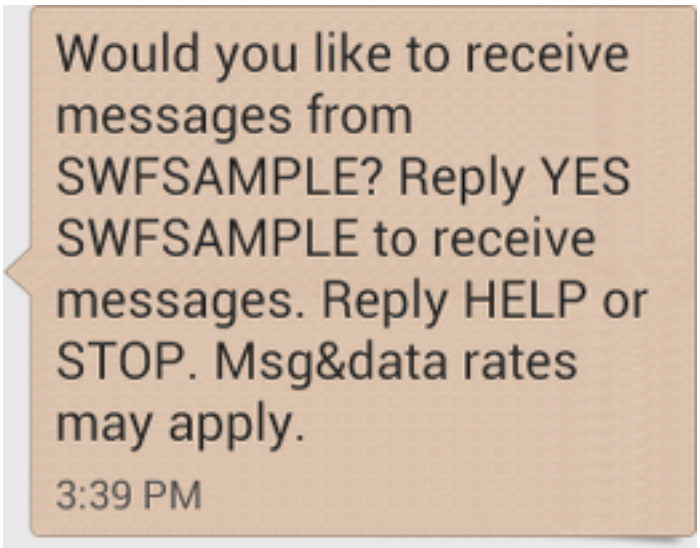
Volte para a outra janela, na qual o seu agente de sondagem está sendo executado. O resultado da primeira atividade em execução é exibido, fornecendo um prompt para você inserir seu e-mail ou número de telefone SMS. Insira um desses dados, ou ambos, e depois confirme sua entrada de texto.

```
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.
```

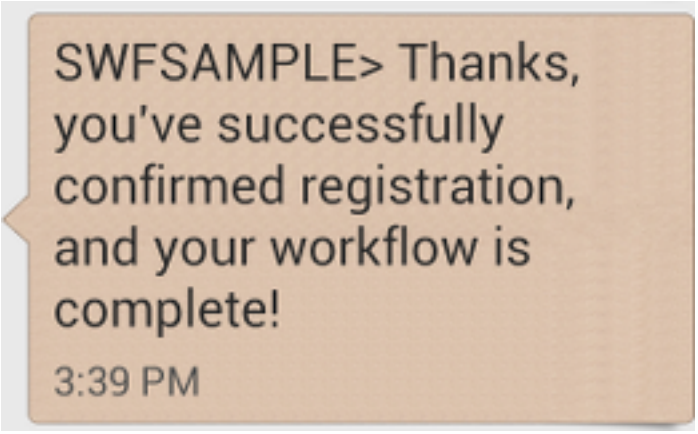
```
If you enter a phone number, it must be able to receive SMS messages, and must
be 11 digits (such as 12065550101 to represent the number 1-206-555-0101).

Email: me@example.com
Phone: 12065550101

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

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

> **Note**
>
> O número de telefone fornecido é fictício e é usado apenas para fins ilustrativos. Use seu próprio número de telefone e endereço de e-mail aqui!

Logo após inserir essas informações, você deverá receber um e-mail ou mensagem de texto do Amazon SNS, solicitando a confirmação da assinatura do tópico. Se tiver digitado um número de SMS, verá algo como o seguinte aparecer no seu telefone.



Se você responder a esta mensagem com `YES`, receberá a resposta que fornecemos em `send_result_activity`.



Enquanto tudo isso estava acontecendo, você viu o que estava acontecendo na sua janela de linha de comando? Os agentes de sondagem de fluxo de trabalho e de atividades tiveram um trabalho duro.

Aqui está a saída do agente de sondagem de fluxo de trabalho.

```
** 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...
```

Veja a seguir a saída do agente de sondagem de atividades, que estava acontecendo ao mesmo tempo em outra janela de linha de comando.

```
++ 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!
```

Parabéns! Seu fluxo de trabalho está completo e este tutorial também!

Você pode querer repetir a execução do fluxo de trabalho para ver como os tempos limite funcionam ou para inserir dados diferentes. Apenas lembre-se de que, depois de se inscrever em um tópico,

você continuará inscrito até cancelar o processo. Executar novamente o fluxo de trabalho antes de cancelar a assinatura dos tópicos provavelmente resultará em sucesso automático, pois `wait_for_confirmation_activity` verá que sua assinatura já foi confirmada.

Para cancelar a assinatura do tópico do Amazon SNS

- Responda negativamente (envie `STOP`) à mensagem de texto.
- Escolha o link de cancelamento de inscrição que você recebeu em seu e-mail.

Agora, você está pronto para repetir a inscrição no tópico.

Para onde ir agora?

Este tutorial abordou muitos tópicos, mas ainda há muito mais que você pode aprender sobre o AWS SDK para Ruby Amazon SWF ou o Amazon SNS. Para mais informações e muitos outros exemplos, consulte a documentação oficial de cada um:

- Documentação do AWS SDK para Ruby
- Documentação do Amazon Simple Notification Service
- Documentação do serviço de fluxo de trabalho simples da Amazon

Trabalhando no console Amazon SWF

O console do Amazon SWF fornece opções para configurar, iniciar e gerenciar execuções de fluxo de trabalho.

Com o console do Amazon SWF, você pode:

- Registrar domínios de fluxo de trabalho.
- Registre tipos de fluxo de trabalho e tipos de atividades.
- Inicie, visualize, sinalize, cancele, encerre e reinicie as execuções do fluxo de trabalho.

Registro de um domínio

Os fluxos de trabalho são executados em um AWS recurso chamado domínio, que controla o escopo do fluxo de trabalho. Uma conta AWS pode ter vários domínios, cada um dos quais pode conter vários fluxos de trabalho, mas os fluxos de trabalho em domínios diferentes não podem interagir.

O registro de domínio é a única funcionalidade disponível inicialmente no console. Depois que pelo menos um domínio for registrado, você poderá realizar as seguintes ações para o domínio:

- Registre o fluxo de trabalho e os tipos de atividades.
- Iniciar execuções de fluxo de trabalho.
- Cancelar, finalizar e enviar sinais para execuções de fluxo de trabalho em andamento.
- Reiniciar execuções de fluxo de trabalho encerradas.

Você também pode realizar ações de gerenciamento de domínio, como reprovação e não depreciação de domínios.

Após descontinuar um domínio, não é possível usá-lo para criar novas execuções de fluxo de trabalho ou registrar novos fluxos de trabalho. A descontinuação de um domínio também suspende todas as atividades e fluxos de trabalho registrados no domínio. As execuções que foram iniciadas antes da descontinuação do domínio continuam sendo executadas.

Depois de descontinuar um domínio anteriormente obsoleto, você pode continuar usando o domínio para registrar tipos de fluxo de trabalho e iniciar novas execuções de fluxo de trabalho.

Para obter mais informações sobre essas ações de gerenciamento de domínio, consulte DeprecateDomainUndeprecateDomaine.

Registrando tipos de fluxo de trabalho

Você pode registrar tipos de fluxo de trabalho no console Amazon SWF depois de registrar pelo menos um domínio.

Um tipo de fluxo de trabalho é um conjunto de tipos de atividades que realizam um objetivo e contêm a lógica que coordena as atividades. Os tipos de fluxo de trabalho coordenam e gerenciam a execução de atividades que podem ser executadas de forma assíncrona em vários dispositivos de computação e apresentam métodos de processamento sequencial e paralelo.

Para registrar um tipo de fluxo de trabalho do Amazon SWF usando o console

1. Abra o domínio no qual você deseja registrar um fluxo de trabalho.
2. Escolha Registrar e, em seguida, Registrar fluxo de trabalho.
3. Na página Registrar fluxo de trabalho, insira o nome do fluxo de trabalho e a versão do fluxo de trabalho. Opcionalmente, você também pode especificar uma lista de tarefas padrão que será usada para agendar tarefas de decisão para execuções desse fluxo de trabalho.
4. (Opcional) Selecione Opções avançadas para especificar os seguintes detalhes do seu fluxo de trabalho:
   - Prioridade de tarefa padrão: a prioridade de tarefa padrão a ser atribuída ao fluxo de trabalho.
   - Tempo limite padrão do início ao fim da execução: a duração máxima padrão para as execuções desse fluxo de trabalho.
   - Tempo limite padrão do início ao encerramento da tarefa: a duração máxima padrão das tarefas de decisão para esse fluxo de trabalho.
   - Política secundária padrão: a política padrão a ser usada para as execuções de fluxo de trabalho secundárias.
   - Função do Lambda padrão: o perfil do IAM padrão anexado a esse fluxo de trabalho.
5. Escolha Registrar fluxo de trabalho.

Registrando tipos de atividades

Atividades são tarefas que você deseja que seu tipo de fluxo de trabalho coordene e execute (por exemplo: verificar o pedido do cliente, cobrar cartão de crédito etc.). A ordem na qual as atividades são executadas é determinada pela lógica de coordenação do tipo de fluxo de trabalho.

Você pode registrar tipos de atividade depois que pelo menos um domínio for registrado.

Para registrar um tipo de atividade do Amazon SWF usando o console

1. Abra o domínio no qual deseja registrar uma atividade.
2. Escolha Registrar e, em seguida, Registrar atividade.
3. Na página Registrar atividade, insira o Nome da atividade e a Versão da atividade. Opcionalmente, você também pode especificar uma lista de tarefas padrão que será usada para agendar tarefas dessa atividade.
4. (Opcional) Escolha Opções avançadas para especificar os seguintes detalhes para sua atividade:
   - Prioridade de tarefa padrão: a prioridade de tarefa padrão a ser atribuída à atividade.
   - Tempo limite padrão para o agendamento de tarefas ser iniciado: a duração máxima padrão que uma tarefa dessa atividade pode esperar antes de ser atribuída a um operador.
   - Tempo limite padrão do início ao encerramento da tarefa: a duração máxima padrão que um operador pode levar para processar as tarefas dessa atividade.
   - Tempo limite padrão para o agendamento de tarefas ser encerrado: a duração máxima padrão para uma tarefa dessa atividade.
   - Tempo limite padrão da pulsação da tarefa — O tempo máximo padrão antes do qual um trabalhador que processa uma tarefa desse tipo deve relatar o progresso chamando. RecordActivityTaskHeartbeat
5. Escolha Registrar atividade.

Iniciando um fluxo de trabalho

Você pode iniciar a execução de um fluxo de trabalho no console do Amazon SWF. Não é possível iniciar a execução de um fluxo de trabalho até que você tenha registrado pelo menos um fluxo de trabalho.

Para iniciar uma execução de fluxo de trabalho usando o console

1. Abra o console do Amazon SWF e, no painel de navegação esquerdo, escolha Domínios.
2. Abaixo do nome do domínio, selecione Fluxos de trabalho.
3. Na página Fluxos de trabalho, escolha o fluxo de trabalho que você deseja executar.
4. Selecione Iniciar execução.
5. Na página Iniciar execução, digite o nome do fluxo de trabalho e o ID de execução para identificar sua execução por um nome. Opcionalmente, você também pode especificar uma Lista de tarefas que será usada para as tarefas de decisão geradas para a execução desse fluxo de trabalho.
6. (Opcional) Escolha Opções avançadas para especificar os seguintes detalhes para a execução do fluxo de trabalho:
   - Prioridade da tarefa: a prioridade da tarefa a ser usada para a execução desse fluxo de trabalho.
   - Tempo limite padrão do início ao fim da execução: a duração total da execução desse fluxo de trabalho.
   - Tempo limite padrão do início ao encerramento da tarefa: a duração máxima das tarefas de decisão para a execução desse fluxo de trabalho.
   - Política secundária — A política a ser usada para a execução do fluxo de trabalho secundário dessa execução do fluxo de trabalho se ela for encerrada, chamando a TerminateWorkflowExecutionação explicitamente ou devido a um tempo limite expirado.
   - Função Lambda — A função do IAM a ser anexada à execução desse fluxo de trabalho.
7. Selecione Iniciar execução.

Gerenciando execuções de fluxo de trabalho

Você pode filtrar suas execuções de fluxo de trabalho por nome, status, ID e tag. Você pode enviar sinais com entradas para execuções ativas do fluxo de trabalho. Se precisar cancelar ou encerrar um fluxo de trabalho, você pode usar a opção Tentar cancelar. O cancelamento é preferível ao encerramento da execução de um fluxo de trabalho, pois o cancelamento dá ao fluxo de trabalho a oportunidade de realizar qualquer tarefa de limpeza e, em seguida, fechar adequadamente.

No console, você pode gerenciar as execuções do fluxo de trabalho que estão atualmente em execução e/ou fechadas.

Para gerenciar suas execuções de fluxo de trabalho

1. Abra um domínio para gerenciar suas execuções de fluxo de trabalho.
2. Escolha Localizar execução.
3. Na página Execuções do fluxo de trabalho, escolha Filtrar execuções por propriedade e, em Propriedades, escolha um dos seguintes filtros:

Escolher	Para aplicar este filtro
Fluxo de trabalho	Escolha esse filtro para listar as execuções de um fluxo de trabalho específico. Por exemplo, para ver as execuções do `fiction-books-order-workflow`, faça o seguinte: 1. Escolha Fluxo de trabalho. 2. Em Operadores, escolha Equals. 3. Em Fluxos de trabalho, escolha fiction-books-order-workflow. 4. (Opcional) Escolha Limpar filtros para remover o filtro e iniciar uma nova busca por execuções.
Status	Escolha esse filtro para listar as execuções com um status específico. Por exemplo, para ver as execuções com o status Falha, faça o seguinte: 1. Escolha Status. 2. Em Operadores, escolha Equals. 3. Em Status, escolha Falha. 4. (Opcional) Escolha Limpar filtros para remover o filtro e iniciar uma nova busca por execuções.
ID de execução	Escolha esse filtro para visualizar a execução de um fluxo de trabalho com base em seu ID. Por exemplo, para visualizar a execução com ID `fiction-books-order-category1`, faça o seguinte: 1. Escolha ID de execução. 2. Em Operadores, escolha Equals. 3. Em Execução IDs, escolha fiction-books-order-category1.

Escolher	Para aplicar este filtro
	4. (Opcional) Escolha Limpar filtros para remover o filtro e iniciar uma nova busca por execuções.
Tag	Escolha esse filtro para listar as execuções com uma tag específica. Por exemplo, para ver as execuções com o status `purchaseOrder` , faça o seguinte: 1. Escolha Tag. 2. Em Operadores, escolha Equals. 3. Em Tag, escolha purchaseOrder. 4. (Opcional) Escolha Limpar filtros para remover o filtro e iniciar uma nova busca por execuções.

4. (Opcional) Depois de aplicar o filtro necessário para listar as execuções do fluxo de trabalho, você pode realizar as seguintes operações em uma execução ativa:

   - Sinal: use essa opção para enviar dados adicionais de execução de um fluxo de trabalho em execução. Para fazer isso:
     1. Escolha a execução para a qual você deseja enviar dados adicionais.
     2. Escolha Signal e, em seguida, especifique os dados na caixa de diálogo de execução do Signal.
     3. Escolha Signal.
   - Tentar cancelar: use essa opção para tentar cancelar a execução de um fluxo de trabalho. É preferível cancelar uma execução de fluxo de trabalho a finalizá-la. O cancelamento fornece à execução de fluxo de trabalho uma oportunidade de realizar tarefas de limpeza e depois ser encerrada corretamente.
     1. Selecione a execução que você deseja cancelar.
     2. Escolha Tentar cancelar.
   - Encerrar: use essa opção para encerrar a execução de um fluxo de trabalho. Observe que é preferível cancelar uma execução de fluxo de trabalho a finalizá-la.
     1. Escolha a execução que você deseja encerrar.
     2. Em Política para crianças, verifique se a opção Encerrar está selecionada.
     3. (Opcional) Especifique o motivo e os detalhes para encerrar a execução.

4. Escolha Encerrar.

5. (Opcional) Processar novamente a execução: use essa opção para processar novamente uma execução de fluxo de trabalho fechado.

   1. Na lista de execuções de fluxo de trabalho, selecione a execução encerrada a ser novamente processada. Quando você seleciona uma execução fechada, o botão Processar novamente a execução fica habilitado. Escolha Processar novamente a execução.
   2. Na página Processar novamente a execução, especifique os detalhes da execução do fluxo de trabalho conforme mencionado em Iniciando um fluxo de trabalho.

Conceitos básicos de fluxo de trabalho no Amazon SWF

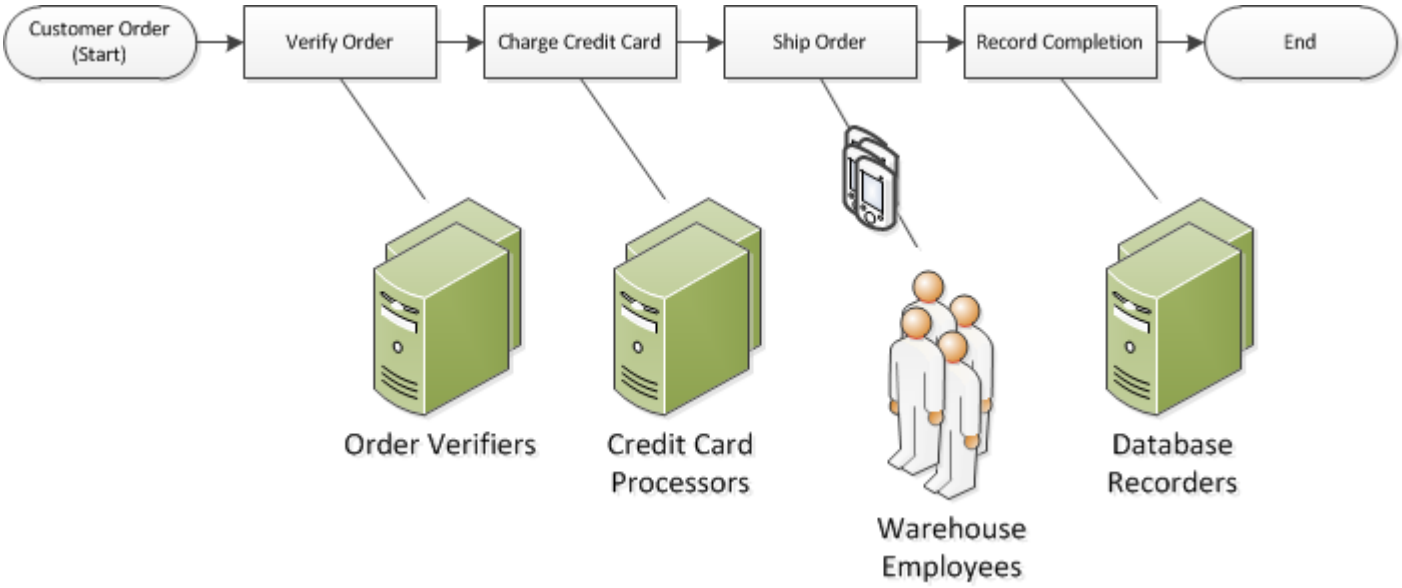
> **Note**
>
> Os conceitos deste capítulo fornecem uma visão geral do Amazon Simple Workflow Service e descrevem seus principais recursos. Se você estiver procurando exemplos, consulte Trabalhando com o Amazon SWF APIs.

Usando o Amazon Simple Workflow Service (Amazon SWF), você pode implementar aplicações distribuídas e assíncronas como fluxos de trabalho. Fluxos de trabalho coordenam e gerenciam a execução de atividades que podem ser executadas de forma assíncrona em vários dispositivos de computação e que podem apresentar processamento em sequência e em paralelo.

Ao projetar um fluxo de trabalho, você analisa seu aplicativo para identificar suas tarefas de componente. No Amazon SWF, essas tarefas são representadas por atividades. A ordem em que as atividades são realizadas é determinada pela lógica de coordenação do fluxo de trabalho.

Exemplo de fluxo de trabalho para um aplicativo de comércio eletrônico

A figura a seguir mostra um fluxo de trabalho de processamento de pedidos de comércio eletrônico envolvendo pessoas e processos automatizados:



O fluxo de trabalho do aplicativo de comércio eletrônico começa quando um cliente faz um pedido e inclui quatro tarefas:

1. Verifique a ordem.
2. Se o pedido for válido, cobrar o cliente.
3. Se o pagamento for feito, enviar o pedido.
4. Se o pedido for enviado, salvar os detalhes do pedido.

As tarefas nesse fluxo de trabalho são sequenciais: um pedido deve ser verificado antes que um cartão de crédito possa ser debitado; um cartão de crédito deve ser debitado com sucesso antes que um pedido possa ser enviado e um pedido deve ser enviado antes que ele possa ser registrado. Mesmo assim, como o Amazon SWF oferece suporte a processos distribuídos, essas tarefas podem ser executadas em diferentes locais. Se as tarefas forem de natureza programática, elas também poderão ser escritas em diferentes linguagens de programação ou usando diferentes ferramentas.

Além do processamento sequencial de tarefas, o Amazon SWF também comporta fluxos de trabalho com processamento paralelo de tarefas. Tarefas paralelas são realizadas ao mesmo tempo e podem ser concluídas de forma independente por diferentes aplicativos ou operadores humanos. Seu fluxo de trabalho toma decisões sobre como proceder depois que uma ou mais tarefas paralelas tiverem sido concluídas.

Conceitos adicionais

- Criação de um fluxo de trabalho no Amazon SWF
- Executando fluxos de trabalho no Amazon SWF
- Histórico do fluxo de trabalho no Amazon SWF
- Identificadores de objetos no Amazon SWF
- Domínios no Amazon SWF
- Atores no Amazon SWF
- Tarefas no Amazon SWF
- Listas de tarefas no Amazon SWF
- Encerramento da execução do fluxo de trabalho no Amazon SWF
- Ciclo de vida de um fluxo de trabalho do Amazon SWF
- Pesquisa para tarefas no Amazon SWF

Criação de um fluxo de trabalho no Amazon SWF

A criação de um fluxo de trabalho sequencial básico envolve as etapas a seguir.

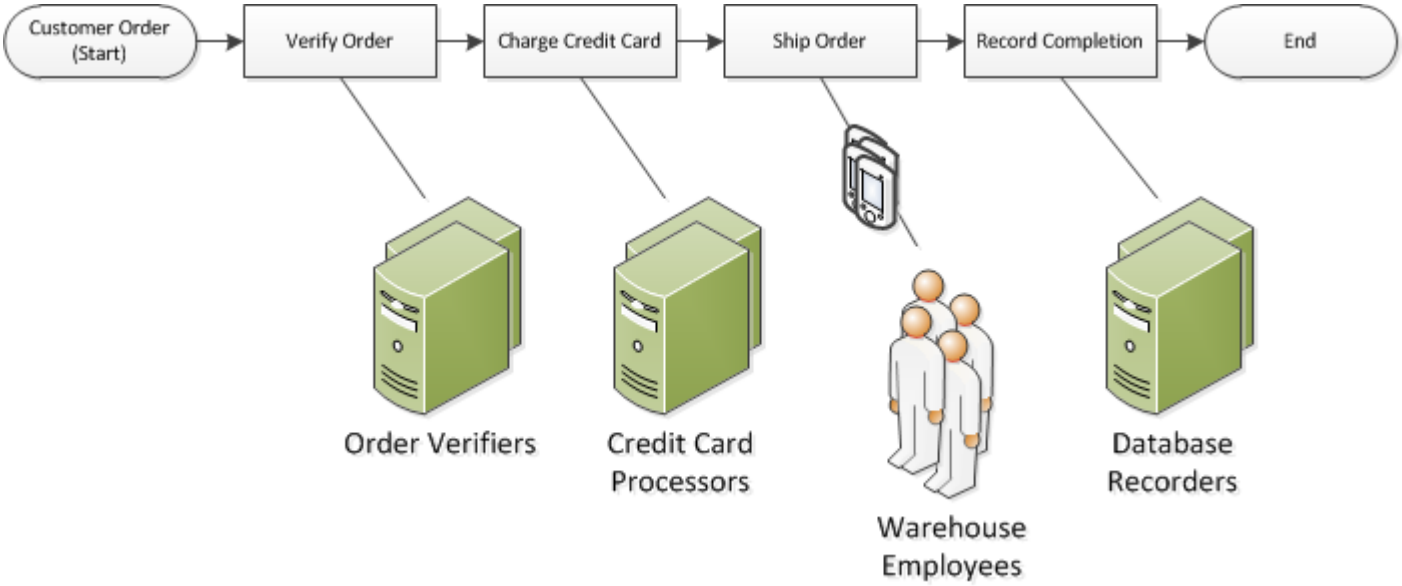
- Modelagem de um fluxo de trabalho, registro do seu tipo e registro dos seus tipos de atividade
- Desenvolvimento e inicialização de operadores de atividade que realizam tarefas de atividade
- Desenvolvimento e inicialização de agentes de decisão que utilizam o histórico de fluxo de trabalho para determinar o que fazer em seguida
- Desenvolvimento e inicialização de iniciadores de fluxo de trabalho, ou seja, aplicativos que iniciam execuções de fluxo de trabalho

Modelagem do fluxo de trabalho e suas atividades

Para usar o Amazon SWF, modele as etapas lógicas em sua aplicação como atividades. Uma atividade representa uma única tarefa ou etapa lógica no seu fluxo de trabalho. Por exemplo, autorizar um cartão de crédito é uma atividade que envolve fornecer um número de cartão de crédito e outras informações, bem como receber um código de aprovação ou uma mensagem de que o cartão foi recusado.

Além de definir atividades, você também precisa definir a lógica de coordenação que lida com pontos de decisão. Por exemplo, a lógica de coordenação pode agendar uma atividade de acompanhamento diferente, dependendo de o cartão de crédito ter sido aprovado ou recusado.

A imagem a seguir mostra um exemplo de um fluxo de trabalho sequencial de pedido de cliente com quatro atividades (Verificar o pedido, Debitar o cartão de crédito, Enviar o pedido e Registrar a conclusão).



Executando fluxos de trabalho no Amazon SWF

Depois que a lógica de coordenação e as atividades tiverem sido projetadas, você registrará esses componentes como tipos de fluxo de trabalho e atividade no Amazon SWF. Durante o registro, você especifica um nome, uma versão e valores de configuração padrão para cada tipo.

Somente os tipos de atividade e fluxo de trabalho registrados podem ser usados com o Amazon SWF. No exemplo de comércio eletrônico, você registraria o tipo CustomerOrder de fluxo de trabalho e os tipos de RecordCompletion atividade VerifyOrder ChargeCreditCard ShipOrder,, e.

Depois de registrar seu tipo de fluxo de trabalho, você poderá executá-lo sempre que quiser. Uma execução de fluxo de trabalho é uma instância em execução de um fluxo de trabalho.

Uma execução de fluxo de trabalho pode ser iniciada por qualquer processo ou aplicativo, até mesmo por outra execução de fluxo de trabalho. No exemplo de comércio eletrônico, uma nova execução de fluxo de trabalho é iniciada com cada pedido do cliente. O tipo de aplicativo que inicia o fluxo de trabalho depende de como o cliente faz o pedido. O fluxo de trabalho pode ser iniciado por um site ou aplicativo móvel ou por um representante de atendimento ao cliente usando um aplicativo interno da empresa.

Com o Amazon SWF, você pode associar um identificador, chamado `workflowId`, às execuções de fluxo de trabalho, para que possa integrar os identificadores de negócios existentes ao fluxo de trabalho. No exemplo de comércio eletrônico, cada execução do fluxo de trabalho pode ser identificada usando o número da fatura do cliente.

Além do identificador que você fornece, o Amazon SWF associa um identificador exclusivo gerado pelo sistema (um `runId`) a cada execução de fluxo de trabalho. O Amazon SWF permite que apenas uma execução de fluxo de trabalho com esse identificador seja executada em um determinado momento; embora você possa ter várias execuções de fluxos de trabalho do mesmo tipo, cada execução de fluxo de trabalho tem um `runId` distinto.

Histórico do fluxo de trabalho no Amazon SWF

O Amazon SWF registra o progresso de cada execução do fluxo de trabalho no histórico do fluxo de trabalho — um registro detalhado, completo e consistente de cada evento que ocorreu desde o início da execução do fluxo de trabalho.

Um evento representa uma mudança discreta no estado da execução do seu fluxo de trabalho, como uma nova atividade sendo agendada ou uma atividade em execução sendo concluída. O histórico

de fluxo de trabalho contém todos os eventos que fazem com que o estado da execução de fluxo de trabalho seja alterado, como atividades agendadas e concluídas, tempos limites de tarefas e sinais.

Em geral, as operações que não alteram o estado da execução de fluxo de trabalho não aparecem no histórico de fluxo de trabalho. Por exemplo, o histórico de fluxo de trabalho não mostra tentativas de sondagem ou o uso de operações de visibilidade.

O histórico de fluxo de trabalho tem vários benefícios-chave:

- Os aplicativos podem ser sem estado, porque todas as informações sobre a execução de um fluxo de trabalho são armazenadas em seu histórico de fluxo de trabalho.
- Para cada execução de fluxo de trabalho, o histórico fornece um registro de quais atividades foram agendadas, seu status atual e seus resultados. A execução de fluxo de trabalho usa essas informações para determinar as próximas etapas.
- O histórico fornece uma trilha de auditoria detalhada que você pode usar para monitorar execuções de fluxo de trabalho em andamento e verificar execuções de fluxo de trabalho concluídas.

O cenário a seguir é uma visão conceitual do histórico de fluxo de trabalho de comércio eletrônico:

```
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
```

No exemplo anterior, o pedido está em espera para envio. No exemplo a seguir, o pedido está concluído. Como o histórico de fluxo de trabalho é cumulativo, os eventos mais recentes são acrescentados:

```
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
```

Programaticamente, os eventos no histórico de execução do fluxo de trabalho são representados como JavaScript objetos de notação de objeto (JSON). O histórico propriamente dito é uma matriz JSON desses objetos. Cada evento tem o seguinte:

- Um tipo, como WorkflowExecutionStartedou ActivityTaskCompleted
- Um carimbo de data/hora no formato de hora do Unix
- Um ID que identifica exclusivamente o evento

Além disso, cada tipo de evento tem um conjunto distinto de atributos descritivos que são apropriados para esse tipo. Por exemplo, o `ActivityTaskCompleted` evento tem atributos que contêm os IDs eventos que correspondem à hora em que a tarefa da atividade foi agendada e quando foi iniciada, bem como um atributo que contém os dados do resultado.

Você pode obter uma cópia do estado atual do histórico de execução do fluxo de trabalho usando a GetWorkflowExecutionHistoryação. Além disso, como parte da interação entre o Amazon SWF e o agente de decisão do seu fluxo de trabalho, o agente de decisão recebe periodicamente cópias do histórico.

Veja abaixo uma seção de um exemplo de histórico de execução de fluxo de trabalho no 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 obter uma lista detalhada dos diferentes tipos de eventos que podem aparecer no histórico de execução do fluxo de trabalho, consulte o tipo de HistoryEventdados na Referência da API do Amazon Simple Workflow Service.

O Amazon SWF armazena o histórico completo de todas as execuções de fluxo de trabalho por um número configurável de dias após o encerramento da execução. Esse período, conhecido como o período de retenção do histórico de fluxo de trabalho, é especificado quando você registra um Domínio para seu fluxo de trabalho. Domínios são discutidos com mais detalhes posteriormente nesta seção.

Identificadores de objetos no Amazon SWF

A lista a seguir descreve como os objetos do Amazon SWF, como as execuções de fluxo de trabalho, são identificados de forma exclusiva.

- Tipo de fluxo de trabalho: um tipo de fluxo de trabalho registrado é identificado por seu domínio, nome e versão. Tipos de fluxo de trabalho são especificados na chamada para `RegisterWorkflowType`.

- Tipo de atividade: um tipo de atividade registrado é identificado pelo respectivo domínio, nome e versão. Tipos de atividades são especificados na chamada para `RegisterActivityType`.
- Tarefas de decisão e tarefas de atividade: cada tarefa de decisão e tarefa de atividade é identificada por um token de tarefa exclusivo. O token de tarefa é gerado pelo Amazon SWF e é retornado com outras informações sobre a tarefa na resposta de `PollForDecisionTask` ou `PollForActivityTask`. Embora o token seja mais comumente usado pelo processo que recebeu a tarefa, esse processo pode transmiti-lo para outro processo, que, por sua vez, pode informar a conclusão ou a falha da tarefa.
- Execução do fluxo de trabalho: uma única execução de um fluxo de trabalho é identificada pelo domínio, ID do fluxo de trabalho e ID de execução. Os dois primeiros são parâmetros que são passados para StartWorkflowExecution. O ID de execução é retornado por `StartWorkflowExecution`.

Domínios no Amazon SWF

Os fluxos de trabalho são executados em um AWS recurso chamado domínio, que fornece uma forma de definir o escopo dos recursos do Amazon SWF em sua conta. AWS Todos os componentes de um fluxo de trabalho, como o tipo de fluxo de trabalho e os tipos de atividades, devem ser especificados para estarem em um domínio.

Uma conta AWS pode ter vários domínios, cada um dos quais pode conter vários fluxos de trabalho, mas os fluxos de trabalho em domínios diferentes não podem interagir.

Ao configurar um novo fluxo de trabalho, antes de configurar qualquer um dos outros componentes do fluxo de trabalho, você precisa registrar um domínio caso ainda não tenha feito isso.

Ao registrar um domínio, você especifica um período de retenção de histórico de fluxo de trabalho. O período de retenção é o período em que o Amazon SWF continuará a reter informações sobre a execução do fluxo de trabalho após a conclusão da execução do fluxo de trabalho.

O registro de domínio é a única funcionalidade disponível inicialmente no console. Depois que pelo menos um domínio for registrado, você poderá realizar as seguintes ações para o domínio:

- Registre o fluxo de trabalho e os tipos de atividades.
- Iniciar execuções de fluxo de trabalho.
- Cancelar, finalizar e enviar sinais para execuções de fluxo de trabalho em andamento.
- Reiniciar execuções de fluxo de trabalho encerradas.

Você também pode realizar ações de gerenciamento de domínio, como reprovação e não depreciação de domínios.

Após descontinuar um domínio, não é possível usá-lo para criar novas execuções de fluxo de trabalho ou registrar novos fluxos de trabalho. A descontinuação de um domínio também suspende todas as atividades e fluxos de trabalho registrados no domínio. As execuções que foram iniciadas antes da descontinuação do domínio continuam sendo executadas.

Depois de descontinuar um domínio anteriormente obsoleto, você pode continuar usando o domínio para registrar tipos de fluxo de trabalho e iniciar novas execuções de fluxo de trabalho.

Para obter mais informações sobre essas ações de gerenciamento de domínio, consulte DeprecateDomainUndeprecateDomaine.

Atores no Amazon SWF

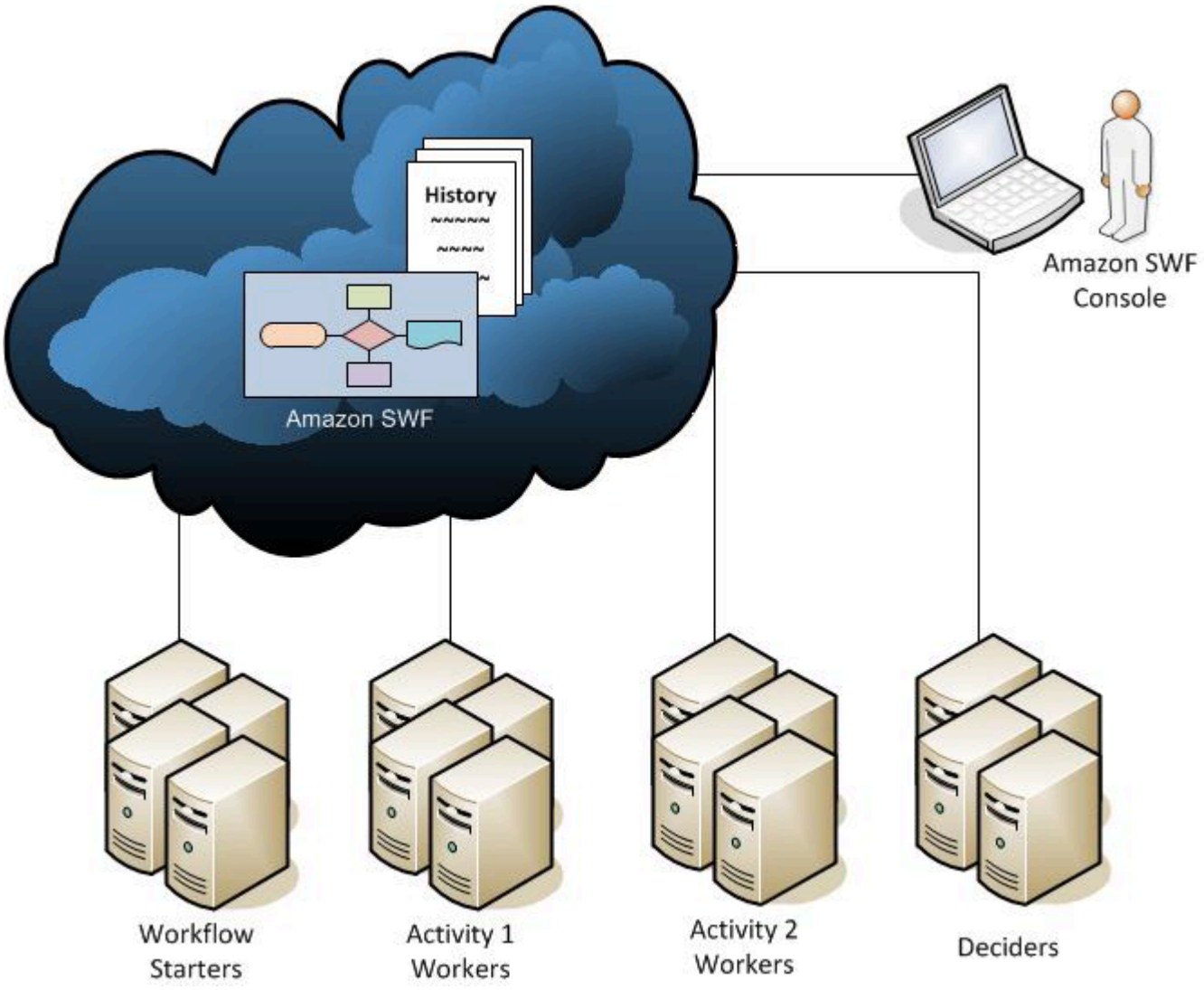
Tópicos

- O que é um ator no Amazon SWF?
- Iniciadores de fluxo de trabalho
- Agentes de decisão
- Operadores de atividade
- Intercâmbio de dados entre atuadores

O que é um ator no Amazon SWF?

No decorrer de suas operações, o Amazon SWF interage com vários tipos diferentes de atores programáticos. Esses atuadores podem ser iniciadores de fluxo de trabalho, agentes de decisão ou operadores de atividade. Esses atores se comunicam com o Amazon SWF por meio de sua API. Você pode desenvolver esses atuadores em qualquer linguagem de programação.

O diagrama a seguir mostra a arquitetura do Amazon SWF, incluindo o Amazon SWF e seus atores.



Iniciadores de fluxo de trabalho

Um iniciador de fluxo de trabalho é qualquer aplicativo capaz de iniciar execuções de fluxo de trabalho. No exemplo de comércio eletrônico, um iniciador de fluxo de trabalho pode ser o site no qual o cliente faz um pedido. Outro iniciador de fluxo de trabalho pode ser um aplicativo ou sistema móvel usado por um representante de atendimento ao cliente para fazer o pedido em nome do cliente.

Agentes de decisão

Um agente de decisão é uma implementação da lógica de coordenação de um fluxo de trabalho. Agentes de decisão controlam o fluxo de tarefas da atividade em uma execução de fluxo de trabalho. Sempre que uma alteração ocorre durante uma execução de fluxo de trabalho, como a conclusão

de uma tarefa, uma tarefa de decisão, incluindo o histórico de fluxo de trabalho inteiro, é transmitida a um agente de decisão. Quando o agente de decisão recebe a tarefa de decisão do Amazon SWF, ele analisa o histórico de execução do fluxo de trabalho para determinar as próximas etapas apropriadas na execução do fluxo de trabalho. O agente de decisão comunica essas etapas de volta ao Amazon SWF usando decisões. Uma decisão é um tipo de dados do Amazon SWF que pode representar várias ações seguintes. Para obter uma lista das possíveis decisões, acesse Decisão na Amazon Simple Workflow Service API Reference.

Veja a seguir um exemplo de uma decisão no formato JSON, o formato no qual ela é transmitida para o Amazon SWF. Essa decisão agenda uma nova tarefa de atividade.

```
{
   "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"
   }
}
```

Um agente de decisão recebe uma tarefa de decisão quando a execução de fluxo de trabalho é iniciada e sempre que uma alteração de estado ocorre nessa execução. Os agentes de decisão continuam a avançar na execução do fluxo de trabalho recebendo tarefas de decisão e respondendo ao Amazon SWF com mais decisões até que o agente de decisão determine que a execução do fluxo de trabalho está concluída. Em seguida, ele responde com uma decisão para encerrar a execução de fluxo de trabalho. Após o encerramento da execução do fluxo de trabalho, o Amazon SWF não agendará tarefas adicionais para essa execução.

No exemplo de comércio eletrônico, o agente de decisão determina se cada etapa foi realizada corretamente e, em seguida, agenda a próxima etapa ou gerencia quaisquer condições de erro.

Um agente de decisão representa um único processo ou thread de computador. Vários agentes de decisão podem processar tarefas para o mesmo tipo de fluxo de trabalho.

Operadores de atividade

Um operador de atividade é um processo ou thread que realiza as tarefas de atividade que fazem parte do seu fluxo de trabalho. A tarefa de atividade representa uma das tarefas que você identificou no seu aplicativo.

Para usar uma tarefa de atividade em seu fluxo de trabalho, você deve registrá-la usando o console Amazon SWF ou a RegisterActivityTypeação.

Cada funcionário de atividade pesquisa o Amazon SWF em busca de novas tarefas que sejam apropriadas para serem executadas por esse funcionário; determinadas tarefas podem ser executadas somente por determinados funcionários de atividade. Depois de receber uma tarefa, o operador de atividade processa a tarefa até a conclusão e, em seguida, informa ao Amazon SWF que a tarefa foi concluída e fornece o resultado. Depois disso, o operador de atividade faz uma sondagem em busca de uma nova tarefa. Os operadores de atividades associados a uma execução de fluxo de trabalho continuam dessa maneira, processando tarefas até a execução de fluxo de trabalho propriamente dita seja concluída. No exemplo de comércio eletrônico, os operadores de atividade são processos e aplicativos independentes usados por pessoas, como processadores de cartão de crédito e funcionários de depósitos, que realizam etapas individuais no processo.

Um operador de atividade representa um único processo (ou thread) de computador. Vários operadores de atividade podem processar tarefas do mesmo tipo de atividade.

Intercâmbio de dados entre atuadores

Dados de entrada podem ser fornecidos para uma execução de fluxo de trabalho quando esta é iniciada. Da mesma forma, dados de entrada podem ser fornecidos aos operadores de atividade quando eles agendam tarefas de atividade. Quando uma tarefa de atividade é concluída, o operador de atividade pode retornar os resultados para o Amazon SWF. Da mesma forma, um agente de decisão poderá relatar os resultados de uma execução de fluxo de trabalho quando essa execução estiver concluída. Cada ator pode enviar e receber dados do Amazon SWF por meio de strings, cuja forma é definida pelo usuário. Dependendo do tamanho e da sensibilidade dos dados, você pode passar os dados diretamente ou passar um ponteiro para os dados armazenados em outro sistema ou serviço (como o Amazon S3 ou o DynamoDB). Tanto os dados passados diretamente quanto os ponteiros para outros armazenamentos de dados são registrados no histórico de execução do fluxo

de trabalho; no entanto, o Amazon SWF não copia nem armazena em cache nenhum dos dados de armazenamentos externos como parte do histórico.

Como o Amazon SWF mantém o estado de execução completo de cada execução de fluxo de trabalho, incluindo as entradas e os resultados das tarefas, todos os atores podem ser stateless. Como resultado, o processamento do fluxo de trabalho é altamente escalável. À medida que a carga no seu sistema crescer, basta adicionar mais atuadores para aumentar a capacidade.

Tarefas no Amazon SWF

O Amazon SWF interage com os operadores de atividade e os agentes decisão, fornecendo-lhes atribuições de trabalho conhecidas como tarefas. Existem três tipos de tarefa no Amazon SWF:

- Tarefa de atividade: uma tarefa de atividade diz a um operador de atividade para executar sua função, como verificar o estoque ou cobrar um cartão de crédito. A tarefa de atividade contém todas as informações necessárias para o operador de atividade executar sua função.
- Tarefa do Lambda: uma tarefa do Lambda é semelhante a uma tarefa de atividade, mas executa uma função do Lambda em vez de uma atividade tradicional do Amazon SWF. Para obter mais informações sobre como definir uma tarefa Lambda, consulte AWS Lambda tarefas no Amazon SWF.
- Tarefa de decisão: uma tarefa de decisão informa a um agente de decisão que o estado da execução do fluxo de trabalho foi alterado para que ele possa determinar a próxima atividade que precisa ser realizada. A tarefa de decisão contém o histórico de fluxo de trabalho atual.

O Amazon SWF agenda uma tarefa de decisão quando o fluxo de trabalho é iniciado e sempre que o estado do fluxo de trabalho é alterado, como quando uma tarefa de atividade é concluída. Cada tarefa de decisão contém uma visão paginada de todo o histórico de execução de fluxo de trabalho. O agente de decisão analisa o histórico de execução do fluxo de trabalho e responde de volta ao Amazon SWF com um conjunto de decisões que especificam o que deve ocorrer em seguida na execução do fluxo de trabalho. Essencialmente, cada tarefa de decisão dá ao agente de decisão uma oportunidade de avaliar o fluxo de trabalho e fornecer orientação de volta ao Amazon SWF.

Para garantir que nenhuma decisão conflitante seja processada, o Amazon SWF atribui cada tarefa de decisão a exatamente um agente de decisão e permite que apenas uma tarefa de decisão por vez esteja ativa em uma execução de fluxo de trabalho.

A tabela a seguir mostra a relação entre as diferentes construções relacionadas a fluxos de trabalho e agentes de decisão.

Design lógico	Registrado como	Realizado por	Recebe e realiza	Gera
Fluxo de trabalho	Tipo de fluxo de trabalho	Agente de decisão	Tarefas de decisão	Decisões

Quando um operador de atividade conclui a tarefa da atividade, ele relata ao Amazon SWF que a tarefa foi concluída e inclui todos os resultados relevantes que foram gerados. O Amazon SWF atualiza o histórico de execução do fluxo de trabalho com um evento que indica que a tarefa foi concluída e, em seguida, agenda uma tarefa de decisão para transmitir o histórico atualizado para o agente de decisão.

O Amazon SWF atribui cada tarefa de atividade a exatamente um operador de atividade. Uma vez que a tarefa é atribuída, nenhum outro operador de atividade pode reivindicar ou realizar essa tarefa.

A tabela a seguir mostra a relação entre as diferentes construções relacionadas a atividades.

Design lógico	Registrado como	Realizado por	Recebe e realiza	Gera
Atividade	Tipo de atividade	Operador de atividade	Tarefas de atividade	Dados de resultados

Listas de tarefas no Amazon SWF

Listas de tarefas fornecem uma maneira de organizar as diversas tarefas associadas a um fluxo de trabalho. Você pode pensar em listas de tarefas como sendo semelhantes a filas dinâmicas. Quando uma tarefa é agendada no Amazon SWF, você pode especificar uma fila (lista de tarefas) para colocá-la. Da mesma forma, ao pesquisar o Amazon SWF para uma tarefa, você diz de qual fila (lista de tarefas) obter a tarefa.

Listas de tarefas fornecem um mecanismo flexível para rotear tarefas aos operadores conforme exigido pelo seu caso de uso. Listas de tarefas são dinâmicas na medida em que você não precisa registrá-las ou criá-las explicitamente por meio de ações. O simples agendamento de uma tarefa cria a lista de tarefas caso ela ainda não exista.

Existem listas separadas para tarefas de atividade e tarefas de decisão. Uma tarefa sempre é agendada em apenas uma lista de tarefas; tarefas não são compartilhadas entre listas. Além disso,

assim como atividades e fluxos de trabalho, as listas de tarefas têm como escopo uma AWS região específica e um domínio do Amazon SWF.

Tópicos

- Listas de tarefas de decisão
- Listas de tarefas de atividade
- Roteamento de tarefas

Listas de tarefas de decisão

Cada execução de fluxo de trabalho está associada a uma lista de tarefas de decisão específica. Quando um tipo de fluxo de trabalho é registrado (RegisterWorkflowTypeação), você pode especificar uma lista de tarefas padrão para execuções desse tipo de fluxo de trabalho. Quando o iniciador de fluxo de trabalho inicia a execução de fluxo de trabalho (ação `StartWorkflowExecution`), ele tem a opção de especificar uma lista de tarefas diferente para essa execução de fluxo de trabalho.

Quando um agente de decisão faz uma sondagem em busca de uma nova tarefa de decisão (ação `PollForDecisionTask`), o agente de decisão especifica uma lista de tarefas de decisão da qual extrair. Um único agente de decisão pode servir várias execuções de fluxo de trabalho ao chamar `PollForDecisionTask` várias vezes, usando uma lista de tarefas diferente em cada chamada, em que cada lista de tarefas é específica para uma execução de fluxo de trabalho em particular. Como alternativa, o agente de decisão pode fazer a sondagem de uma única lista de tarefas de decisão que fornece tarefas de decisão para várias execuções de fluxo de trabalho. Também é possível ter vários agentes de decisão servindo uma única execução de fluxo de trabalho, fazendo com que todos eles realizem uma sondagem da lista de tarefas para essa execução de fluxo de trabalho.

Listas de tarefas de atividade

Uma única lista de tarefas de atividade pode conter tarefas de diferentes tipos de atividades. As tarefas são agendadas na lista de tarefas em ordem. O Amazon SWF retorna as tarefas da lista em uma base de melhor esforço. Em algumas circunstâncias, as tarefas podem não sair da lista em ordem.

Quando um tipo de atividade é registrado (RegisterActivityTypeação), você pode especificar uma lista de tarefas padrão para esse tipo de atividade. Por padrão, tarefas de atividade desse tipo serão agendadas na lista de tarefas especificada; no entanto, quando o decisor agenda uma tarefa de

atividade (ScheduleActivityTaskdecisão), o decisor pode, opcionalmente, especificar uma lista de tarefas diferente na qual programar a tarefa. Se o agente de decisão não especificar uma lista de tarefas, a lista de tarefas padrão será usada. Como resultado, você pode colocar tarefas de atividade em listas de tarefas específicas de acordo com os atributos da tarefa. Por exemplo, é possível colocar todas as instâncias de uma tarefa de atividade para um determinado tipo de cartão de crédito em uma lista de tarefas específica.

Roteamento de tarefas

Quando um funcionário da atividade pesquisa uma nova tarefa (PollForActivityTaskação), ele pode especificar uma lista de tarefas da atividade a partir da qual se basear. Se isso for feito, ele aceitará tarefas somente dessa lista. Dessa forma, você pode garantir que certas tarefas sejam atribuídas apenas a operadores de atividade específicos. Por exemplo, você pode criar uma lista de tarefas que contenha tarefas que exigem o uso de um computador de alto desempenho. Apenas os operadores de atividade em execução no hardware apropriado sondarão essa lista de tarefas. Outro exemplo seria criar uma lista de tarefas para uma determinada região geográfica. Dessa forma, você pode garantir que apenas os operadores implantados nessa região selecionarão essas tarefas. Ou você pode criar uma lista de tarefas para ordens de alta prioridade e sempre conferir essa lista primeiro.

A atribuição de tarefas específicas a operadores de atividade particulares dessa forma é um processo chamado de roteamento de tarefas. O roteamento de tarefas é opcional; se você não especificar uma lista de tarefas ao agendar uma tarefa de atividade, esta será automaticamente colocada na lista de tarefas padrão.

Encerramento da execução do fluxo de trabalho no Amazon SWF

Quando você inicia a execução de um fluxo de trabalho, ele é aberto. Uma execução de fluxo de trabalho aberta pode ser encerrada como concluída, cancelada, com falha ou expirada. Ela também pode ser continuada como uma nova execução ou pode ser finalizada. A execução de um fluxo de trabalho pode ser encerrada pelo agente de decisão, pela pessoa que administra o fluxo de trabalho ou pelo Amazon SWF.

Se o agente de decisão determinar que as atividades do fluxo de trabalho terminaram, ele deverá encerrar a execução de fluxo de trabalho como concluída usando a ação `RespondDecisionTaskCompleted`e transmitir a decisão `CompleteWorkflowExecution`.

Como alternativa, um agente de decisão pode encerrar a execução de fluxo de trabalho como cancelada ou com falha. Para cancelar a execução, o agente de decisão deve usar a ação `RespondDecisionTaskCompleted` e transmitir a decisão `CancelWorkflowExecution`.

Um agente de decisão deve marcar a execução de fluxo de trabalho como falha se ela entrar em um estado fora do domínio de uma conclusão normal. Para marcar a execução como falha, o agente de decisão deve usar a ação `RespondDecisionTaskCompleted` e transmitir a decisão `FailWorkflowExecution`.

O Amazon SWF monitora as execuções do fluxo de trabalho para garantir que elas não excedam as configurações de tempo limite especificadas pelo usuário. Se a execução de um fluxo de trabalho atingir o tempo limite, o Amazon SWF o fechará automaticamente. Para obter mais informações sobre valores de tempo limite, consulte a seção Tipos de tempo limite do Amazon SWF .

Um agente de decisão também pode encerrar a execução e continuá-la logicamente como uma nova execução usando a ação `RespondDecisionTaskCompleted` e transmitindo a decisão `ContinueAsNewWorkflowExecution`. Essa estratégia é útil para execuções de fluxo de trabalho de longa duração, para as quais o histórico pode crescer demais com o passar do tempo.

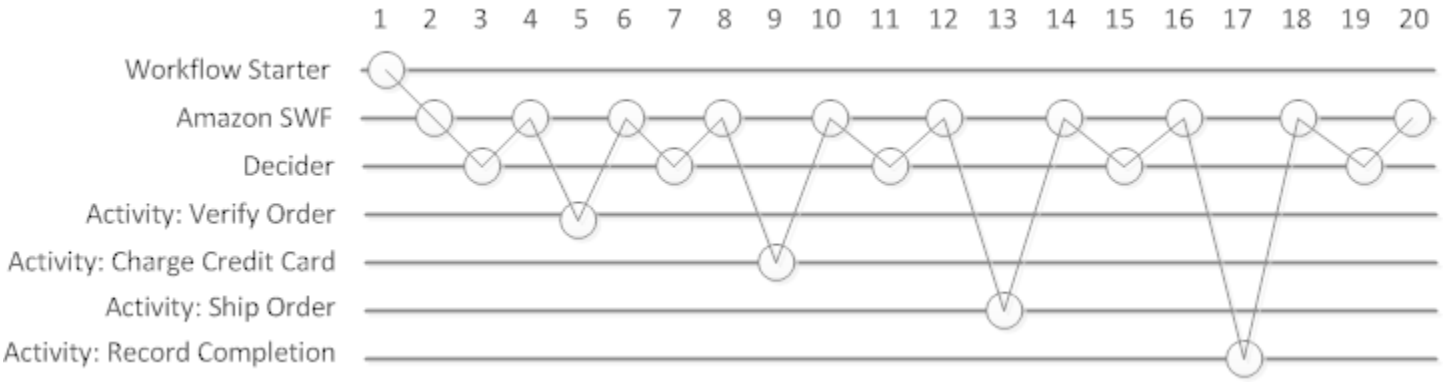
Por fim, você pode encerrar as execuções do fluxo de trabalho diretamente do console do Amazon SWF ou programaticamente usando a API `TerminateWorkflowExecution`. A finalização força o encerramento da execução de fluxo de trabalho. É preferível o cancelamento à finalização, pois os agentes de decisão podem gerenciar o encerramento da execução de fluxo de trabalho.

O Amazon SWF encerra a execução de um fluxo de trabalho se a execução exceder determinados limites definidos pelo serviço. O Amazon SWF encerra um fluxo de trabalho secundário se o fluxo de trabalho principal for encerrado e a política secundária aplicável indicar que o fluxo de trabalho secundário também deve ser encerrado.

Ciclo de vida de um fluxo de trabalho do Amazon SWF

Desde o início da execução de um fluxo de trabalho até sua conclusão, o Amazon SWF interage com os atores atribuindo-lhes tarefas apropriadas, sejam elas tarefas de atividade ou tarefas de decisão.

O diagrama a seguir mostra o ciclo de vida de uma execução de fluxo de trabalho de processamento de pedidos a partir da perspectiva dos componentes que atuam nela.



Ciclo de vida da execução de fluxo de trabalho

A tabela a seguir explica cada tarefa na imagem anterior.

Descrição	Ação, decisão ou evento
1. O iniciador do fluxo de trabalho chama a ação apropriada do Amazon SWF para iniciar a execução do fluxo de trabalho para um pedido, fornecendo as informações do pedido.	Ação `StartWorkflowExecution`.
2. O Amazon SWF recebe a solicitação de início da execução do fluxo de trabalho e, em seguida, agenda a primeira tarefa de decisão.	Evento `WorkflowExecutionStarted` e evento `DecisionTaskScheduled`.
3. O agente de decisão recebe a tarefa do Amazon SWF, analisa o	A ação `PollForDecisionTask`. A ação `RespondDecisionTaskCompleted` e decisão `ScheduleActivityTask`.

Descrição	Ação, decisão ou evento
histórico, aplica a lógica de coordenação para determinar que nenhuma atividade anterior ocorreu, toma a decisão de agendar a atividade Verificar pedido com as informações que o operador de atividade precisa para processar a tarefa e retorna a decisão ao Amazon SWF.	
4. O Amazon SWF recebe a decisão, agenda a tarefa de atividade Verificar pedido e aguarda a conclusão ou o tempo limite da tarefa de atividade.	Evento `ActivityTaskScheduled`
5. Um operador de atividade que pode executar a atividade Verificar pedido recebe a tarefa, executa-a e retorna os resultados para o Amazon SWF.	Ação `PollForActivityTask` e ação `RespondActivityTaskCompleted`.

Descrição	Ação, decisão ou evento
6. O Amazon SWF recebe os resultados da atividade Verificar pedido, adiciona-os ao histórico do fluxo de trabalho e agenda uma tarefa de decisão.	Evento `ActivityTaskCompleted` e evento `DecisionTaskScheduled`.
7. O decisor recebe a tarefa do Amazon SWF, revisa o histórico, aplica a lógica de coordenação, toma a decisão de ChargeCreditCard programar uma tarefa de atividade com as informações de que o funcionário da atividade precisa para processar a tarefa e retorna a decisão ao Amazon SWF.	A ação `PollForDecisionTask`. A ação `RespondDecisionTaskCompleted` com a decisão `ScheduleActivityTask`.
8. O Amazon SWF recebe a decisão, programa a tarefa da ChargeCreditCard atividade e espera que ela seja concluída ou atinja o tempo limite.	Evento `DecisionTaskCompleted` e evento `ActivityTaskScheduled`.

Descrição	Ação, decisão ou evento
9. Um funcionário da atividade que pode realizar a ChargeCreditCard atividade recebe a tarefa, a executa e retorna os resultados para o Amazon SWF.	Ações `PollForActivityTask` e `RespondActivityTaskCompleted`.
10. O Amazon SWF recebe os resultados da tarefa de ChargeCreditCard atividade, os adiciona ao histórico do fluxo de trabalho e agenda uma tarefa de decisão.	Evento `ActivityTaskCompleted` e evento `DecisionTaskScheduled`.
11. O decisor recebe a tarefa do Amazon SWF, revisa o histórico, aplica a lógica de coordenação, toma a decisão de ShipOrder programar uma tarefa de atividade com as informações de que o funcionário da atividade precisa para realizar a tarefa e retorna a decisão ao Amazon SWF.	A ação `PollForDecisionTask`. O `RespondDecisionTaskCompleted` com a decisão `ScheduleActivityTask`.

Descrição	Ação, decisão ou evento
12. O Amazon SWF recebe a decisão, programa uma tarefa de ShipOrder atividade e espera que ela seja concluída ou atinja o tempo limite.	Evento `DecisionTaskCompleted` e evento `ActivityTaskScheduled`.
13. Um funcionário da atividade que pode realizar a ShipOrder atividade recebe a tarefa, a executa e retorna os resultados para o Amazon SWF.	Ação `PollForActivityTask` e ação `RespondActivityTaskCompleted`.
14. O Amazon SWF recebe os resultados da tarefa de ShipOrder atividade, os adiciona ao histórico do fluxo de trabalho e agenda uma tarefa de decisão.	Evento `ActivityTaskCompleted` e evento `DecisionTaskScheduled`.

Descrição	Ação, decisão ou evento
15. O decisor recebe a tarefa do Amazon SWF, revisa o histórico, aplica a lógica de coordenação, toma a decisão de RecordCompletion programar uma tarefa de atividade com as informações de que o funcionário da atividade precisa para realizar a tarefa e retorna a decisão ao Amazon SWF.	A ação `PollForDecisionTask`. A ação `RespondDecisionTaskCompleted` com a decisão `ScheduleActivityTask`.
16. O Amazon SWF recebe a decisão, programa uma tarefa de RecordCompletion atividade e espera que ela seja concluída ou atinja o tempo limite.	Evento `DecisionTaskCompleted` e evento `ActivityTaskScheduled`.
17. Um funcionário da atividade que pode realizar a RecordCompletion atividade recebe a tarefa, a executa e retorna os resultados para o Amazon SWF.	Ação `PollForActivityTask` e ação `RespondActivityTaskCompleted`.

Descrição	Ação, decisão ou evento
18. O Amazon SWF recebe os resultados da tarefa de RecordCompletion atividade, os adiciona ao histórico do fluxo de trabalho e agenda uma tarefa de decisão.	Evento `ActivityTaskCompleted` e evento `DecisionTaskScheduled`.
19. O agente de decisão recebe a tarefa do Amazon SWF, analisa o histórico, aplica a lógica de coordenação, toma uma decisão para encerrar a execução do fluxo de trabalho e retorna a decisão junto com quaisquer resultados para o Amazon SWF.	A ação `PollForDecisionTask`. A ação `RespondDecisionTaskCompleted` com a decisão `CompleteWorkflowExecution`.
20. O Amazon SWF encerra a execução do fluxo de trabalho e arquiva o histórico para referência futura.	Evento `WorkflowExecutionCompleted`.

Pesquisa para tarefas no Amazon SWF

Os agente de decisão e os operadores de atividade se comunicam com o Amazon SWF usando polling longo. O agente de decisão ou operador de atividade inicia periodicamente a comunicação

com o Amazon SWF, notificando-o de sua disponibilidade para aceitar uma tarefa e, em seguida, especifica uma lista de tarefas da qual obterá tarefas.

Se uma tarefa estiver disponível na lista de tarefas especificada, o Amazon SWF a retornará imediatamente na resposta. Se nenhuma tarefa estiver disponível, o Amazon SWF manterá a conexão TCP aberta por até 60 segundos para que, se uma tarefa ficar disponível durante esse período, ela possa ser retornada na mesma conexão. Se nenhuma tarefa se tornar disponível em até 60 segundos, ele retornará uma resposta vazia e encerrará a conexão. (Uma resposta vazia é uma estrutura Task na qual o valor de taskToken é uma string vazia.) Se isso acontecer, o agente de decisão ou o operador de atividade deverá repetir a sondagem.

A sondagem longa funciona bem para o processamento de tarefas de alto volume. Agentes de decisão e operadores de atividade podem gerenciar sua própria capacidade, e o processo é fácil de usar quando eles estão atrás de um firewall.

Para obter mais informações, consulte Sondar tarefas de decisão e Sondar tarefas de atividade.

Conceitos avançados de fluxo de trabalho no Amazon SWF

O exemplo de comércio eletrônico na seção ??? representa um cenário de fluxo de trabalho simplificado. Na vida real, é provável que você queira que o seu fluxo de trabalho realize tarefas simultâneas (enviar um e-mail de confirmação de pedido e, ao mesmo tempo, autorizar um cartão de crédito), registrar os principais eventos (todos os itens embalados), atualizar o pedido com alterações (adicionar ou remover um item) e tomar outras decisões mais avançadas como parte da execução do fluxo de trabalho. Esta seção descreve conceitos avançados de fluxo de trabalho que você pode usar para criar seus fluxos de trabalho.

Conceitos avançados

- Versionamento
- Sinais
- Fluxos de trabalho secundários no Amazon SWF
- Marcadores no Amazon SWF
- Tags no Amazon SWF
- Implementando opções exclusivas com o Amazon SWF
- Temporizadores no Amazon SWF
- Cancelamento de tarefas de atividade no Amazon SWF

Versionamento

Muitas vezes, as necessidades de negócios exigem que você tenha implementações diferentes ou variações do mesmo fluxo de trabalho ou atividade em execução simultaneamente. Por exemplo, você pode querer testar uma nova implementação de um fluxo de trabalho enquanto outra está em produção. Você também pode querer executar duas implementações diferentes com dois conjuntos de recursos distintos, como uma implementação básica e outra especial. O versionamento permite que você execute várias implementações de fluxos de trabalho e atividades simultaneamente, para qualquer finalidade que atenda aos seus requisitos.

Tipos de fluxo de trabalho e atividade possuem uma versão associada, que é especificada na ocasião do registro. A versão é uma string de forma livre, e você pode escolher seu próprio esquema de versionamento. Para criar uma nova versão de um tipo registrado, você deve registrá-lo com o mesmo nome e uma versão diferente. O Listas de tarefas no Amazon SWF , descrito anteriormente,

pode ajudá-lo a implementar o controle de versão. Considere uma situação em que você tenha execuções de fluxo de trabalho de longa duração de um determinado tipo já em andamento, e as circunstâncias exigem que você revise esse fluxo de trabalho, por exemplo, para adicionar um novo recurso. É possível implementar o novo recurso criando novas versões de tipos de atividade e operadores, bem como um novo agente de decisão. Em seguida, você poderia iniciar execuções da nova versão do fluxo de trabalho usando um conjunto diferente de listas de tarefas. Dessa forma, você pode ter execuções de fluxos de trabalho de diferentes versões em execução ao mesmo tempo, sem que uma afete as demais.

Sinais

Sinais permitem que você insira informações em uma execução de fluxo de trabalho em andamento. Em alguns cenários, você pode querer adicionar informações a uma execução de fluxo de trabalho em andamento para informá-la de que algo mudou ou para comunicar um evento externo. Qualquer processo pode enviar um sinal para uma execução de fluxo de trabalho aberta. Por exemplo, uma execução de fluxo de trabalho pode sinalizar outra.

> **Note**
>
> Uma tentativa de enviar um sinal para uma execução de fluxo de trabalho que não esteja aberta resultará na falha de `SignalWorkflowExecution` com `UnknownResourceFault`.

Para usar sinais, defina o nome do sinal e os dados a serem passados para o sinal, se houver. Em seguida, programe o decisor para reconhecer o evento de sinal (WorkflowExecutionSignaled) no histórico e processá-lo adequadamente. Quando um processo quer sinalizar a execução de um fluxo de trabalho, ele faz uma chamada para o Amazon SWF (usando a SignalWorkflowExecutionação ou, no caso de um decisor, usando a SignalExternalWorkflowExecutiondecisão) que especifica o identificador para a execução do fluxo de trabalho de destino, o nome do sinal e os dados do sinal. Em seguida, o Amazon SWF recebe o sinal, o registra no histórico da execução do fluxo de trabalho de destino e programa uma tarefa de decisão para ele. Quando o agente de decisão receber a tarefa de decisão, ele também receberá o sinal dentro do histórico de execução de fluxo de trabalho. O agente de decisão poderá então tomar as medidas apropriadas com base no sinal e seus dados.

Às vezes, convém aguardar um sinal. Por exemplo, um usuário pode cancelar um pedido enviando um sinal, mas somente dentro de uma hora após fazer o pedido. O Amazon SWF não tem uma primitiva para permitir que um agente de decisão espere por um sinal do serviço. A funcionalidade

de pausa precisa ser implementada no próprio agente de decisão. Para fazer uma pausa, o agente de decisão deve iniciar um temporizador, usando a decisão `StartTimer`, que especifica por quanto tempo ele aguardará o sinal enquanto continua a sondar tarefas de decisão. Quando o agente de decisão recebe uma tarefa de decisão, ele deve verificar o histórico para ver se o sinal foi recebido ou se o temporizador foi acionado. Se o sinal tiver sido recebido, o agente de decisão deverá cancelar o temporizador. No entanto, se, em vez disso, o temporizador tiver sido disparado, significa que o sinal não chegou dentro do tempo especificado. Para resumir, a fim de aguardar um sinal específico, faça o seguinte.

1. Crie um temporizador para a quantidade de tempo que o agente de decisão deve esperar.
2. Quando uma tarefa de decisão for recebida, verifique o histórico para ver se o sinal já chegou ou se o temporizador foi acionado.
3. Se um sinal tiver chegado, cancele o temporizador usando uma decisão `CancelTimer` e processe o sinal. Dependendo do cronograma, o histórico pode conter eventos `TimerFired` e `WorkflowExecutionSignaled`. Nesses casos, você pode confiar na ordem relativa dos eventos no histórico para determinar qual ocorreu primeiro.
4. Se o temporizador tiver sido acionado antes do recebimento de um sinal, significa que o agente de decisão atingiu o tempo limite ao aguardar esse sinal. Você pode marcar a execução como falha ou realizar qualquer outra lógica apropriada para o seu caso de uso.

Nos casos em que um fluxo de trabalho deve ser cancelado (por exemplo, o pedido em si foi cancelado pelo cliente), deve-se usar a ação `RequestCancelWorkflowExecution` em vez de enviar um sinal para o fluxo de trabalho.

Algumas aplicações para sinais incluem:

- Pausar o progresso de execuções de fluxo de trabalho até que um sinal seja recebido (por exemplo, aguardar um carregamento de inventário).
- Fornecer informações a uma execução de fluxo de trabalho que possam afetar a lógica de como os agentes de decisão tomam decisões. Isso é útil para fluxos de trabalho afetados por eventos externos (por exemplo, tentativa de finalizar a venda de uma ação após o fechamento do mercado).
- Atualizar uma execução de fluxo de trabalho quando você antecipa que alterações possam ocorrer (por exemplo, alterar quantidades de pedidos depois que estes são efetuados e antes do envio).

No exemplo a seguir, a execução do fluxo de trabalho recebe um sinal para cancelar uma ordem.

```
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"}
```

Se houver sinal na execução do fluxo de trabalho, o Amazon SWF retornará uma resposta HTTP bem-sucedida similar ao seguinte: O Amazon SWF gerará uma tarefa de decisão para informar o tomador de decisão a processar o sinal.

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

Fluxos de trabalho secundários no Amazon SWF

Fluxos de trabalho complicados podem ser divididos em componentes menores, mais gerenciáveis e potencialmente reutilizáveisusando fluxos de trabalho secundários. Um fluxo de trabalho secundário é uma execução de fluxo de trabalho iniciada por outra execução de fluxo de trabalho (principal). Para iniciar um fluxo de trabalho secundário, o agente de decisão do fluxo de trabalho principal usa a decisão `StartChildWorkflowExecution`. Os dados de entrada especificados com essa decisão são disponibilizados ao fluxo de trabalho secundário por meio do seu histórico.

Os atributos da decisão `StartChildWorkflowExecution` também especificam a política do filho, ou seja, como o Amazon SWF deve lidar com a situação em que a execução do fluxo de trabalho pai termina antes da execução do fluxo de trabalho filho. Existem três valores possíveis:

- TERMINATE: o Amazon SWF encerrará as execuções secundárias.
- REQUEST_CANCEL: o Amazon SWF tentará cancelar a execução do filho colocando um evento `WorkflowExecutionCancelRequested` no histórico de execução do fluxo de trabalho do filho.
- ABANDON: o Amazon SWF não tomará nenhuma ação; as execuções secundárias continuarão a ser executadas.

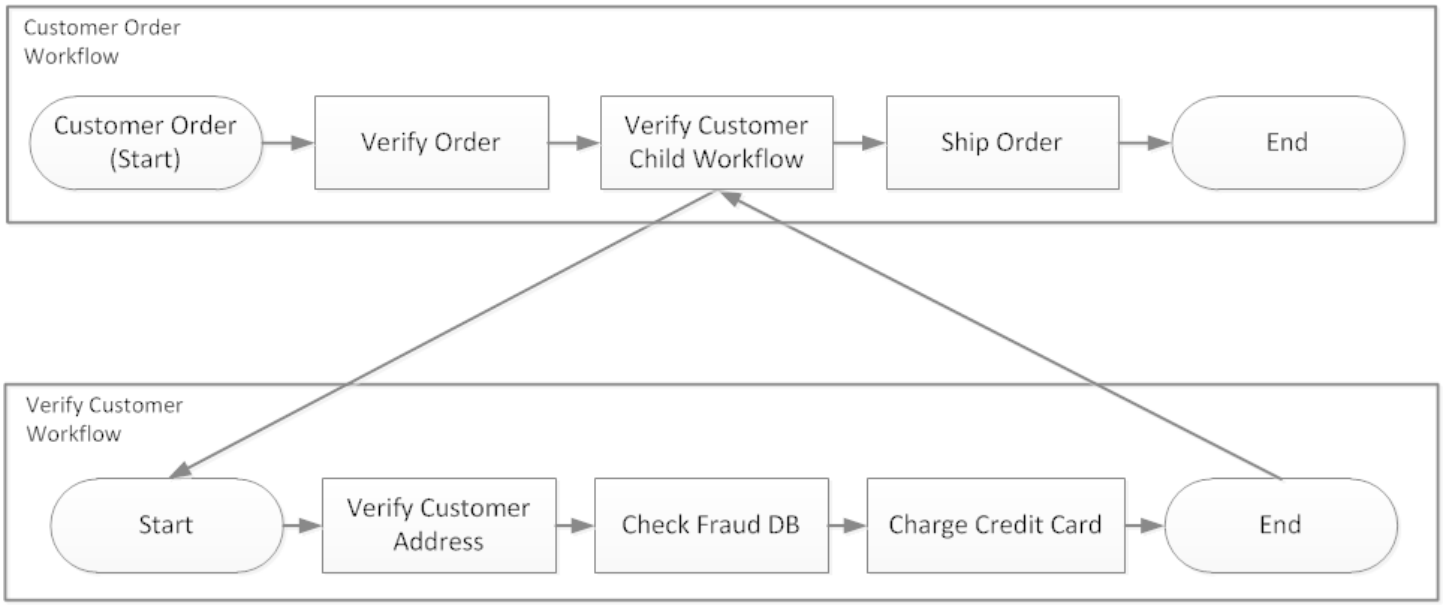
Após o início da execução de fluxo de trabalho secundário, ela é processada como uma execução comum. Quando é concluído, o Amazon SWF registra a conclusão, juntamente com seus resultados,

no histórico do fluxo de trabalho da execução do fluxo de trabalho pai. Exemplos de fluxos de trabalho secundários incluem:

- Fluxo de trabalho secundário de processamento de cartão de crédito usado por fluxos de trabalho em diferentes sites
- Fluxo de trabalho secundário de e-mail que verifica o endereço de e-mail do cliente, verifica a lista de exclusões, envia o e-mail e verifica se ele não foi retornado ou falhou.
- Fluxo de trabalho secundário de armazenamento de banco de dados e recuperação, que combina conexão, configuração, transação e verificação.
- Fluxo de trabalho secundário de compilação de código-fonte, que combina desenvolvimento, empacotamento e verificação.

No exemplo do comércio eletrônico, você pode querer transformar a atividade Debitar o cartão de crédito em um fluxo de trabalho secundário. Para fazer isso, você pode registrar um novo fluxo de trabalho Verificar o cliente, registrar as atividades Verificar o endereço do cliente e Verificar o banco de dados fraudes e definir a lógica de coordenação para as tarefas. Em seguida, um agente de decisão no fluxo de trabalho de Pedido do cliente pode iniciar um fluxo de trabalho secundário Verificar o cliente, agendando a decisão `StartChildWorkflowExecution` que especifica esse tipo de fluxo de trabalho.

A figura a seguir mostra um fluxo de trabalho de pedido do cliente que inclui um novo fluxo de trabalho secundário Verificar o cliente, que verifica o endereço do cliente, pesquisa o banco de dados de fraudes e debita o cartão de crédito.



Vários fluxos de trabalho podem criar execuções de fluxos de trabalho secundários usando o mesmo tipo de fluxo de trabalho. Por exemplo, o fluxo de trabalho secundário Verificar o cliente também pode ser usado em outras partes de uma organização. Os eventos de um fluxo de trabalho secundário estão contidos em seu próprio histórico de fluxo de trabalho e não estão incluídos no histórico de fluxo de trabalho do elemento principal.

Como fluxos de trabalho secundários são simplesmente execuções de fluxo de trabalho iniciadas por um agente de decisão, elas também podem ser iniciadas como execuções de fluxos de trabalho independentes normais.

Marcadores no Amazon SWF

Às vezes, você pode querer registrar informações no histórico de fluxo de trabalho de uma execução de fluxo de trabalho específica para o seu caso de uso. Marcadores permitem que você registre no histórico de execução de fluxo de trabalho informações que podem ser usadas para qualquer finalidade personalizada ou específica de um cenário.

Para usar marcadores, um decisor usa a RecordMarker decisão, nomeia o marcador, anexa os dados desejados à decisão e notifica o Amazon SWF usando a ação. `RespondDecisionTaskCompleted` O Amazon SWF recebe a solicitação, registra o marcador no histórico do fluxo de trabalho e promulga quaisquer outras decisões na solicitação. A partir desse momento, os agentes de decisão podem ver o marcador no histórico de fluxo de trabalho e usá-lo de qualquer maneira que você programar.

Por si só, registrar um marcador não inicia uma tarefa de decisão. Para evitar que a execução de fluxo de trabalho fique presa, deve ocorrer algo que dê continuidade a ela. Por exemplo, isso pode incluir o agendamento de outra tarefa de atividade pelo agente de decisão, o recebimento de um sinal pela execução de fluxo de trabalho ou a conclusão de uma tarefa de atividade anteriormente agendada.

Exemplos de marcadores incluem:

- Um contador que conta o número de loops em um fluxo de trabalho recursivo.
- Progresso da execução de fluxo de trabalho com base nos resultados de atividades.
- Informações resumidas de eventos de histórico de fluxo de trabalho anteriores.

No exemplo de comércio eletrônico, você poderia adicionar uma atividade que verificasse o inventário todos os dias e incrementasse a contagem em um marcador depois de cada verificação.

Em seguida, você poderia adicionar uma lógica que enviasse um e-mail ao cliente ou notificasse um gerente no momento em que a contagem excedesse cinco, sem precisar rever todo o histórico.

No exemplo a seguir, o agente de decisão conclui uma tarefa de decisão e responde com uma ação `RespondDecisionTaskCompleted` que contém uma decisão `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"
          }
      },
  ]
}
```

Se o Amazon SWF registrar o marcador com êxito, ele retornará uma resposta HTTP bem-sucedida semelhante à seguinte.

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

Tags no Amazon SWF

O Amazon SWF é compatível com a marcação de uma execução de fluxo de trabalho. Isso é especialmente útil quando você tem vários recursos.

O Amazon SWF é compatível com a marcação de uma execução de fluxo de trabalho com até cinco tags. Cada tag é uma string de forma livre e pode conter até 256 caracteres. Se quiser usar tags, você deverá atribuí-las quando iniciar uma execução de fluxo de trabalho. Você não pode adicionar tags a uma execução de fluxo de trabalho depois de ela ter sido iniciada, nem pode editar ou remover tags que tenham sido atribuídas a uma execução de fluxo de trabalho.

O IAM é compatível com o controle de acesso aos domínios do Amazon SWF com base em tags. Para controlar o acesso com base em tags, forneça informações sobre as tags no elemento de condição de uma política do IAM.

Gerenciar tags

Gerencie as tags do Amazon Simple Workflow Service usando o AWS SDKs ou interagindo diretamente com a API Amazon SWF. Com o uso da API, é possível adicionar tags ao registrar um domínio, listar tags para um domínio existente e adicionar ou excluir tags para um domínio existente.

> **Note**
>
> Há um limite de 50 tags por recurso. Consulte Cotas gerais de contas para o Amazon SWF

- `RegisterDomain`
- `ListTagsForResource`
- `TagResource`
- `UntagResource`

Para obter mais informações, consulte Trabalhando com o Amazon SWF APIs e Amazon Simple Workflow Service API Reference.

Marcar execuções de fluxo de trabalho

Com o Amazon SWF, você pode associar tags a execuções de fluxo de trabalho e, em seguida, consultar execuções de fluxo de trabalho com base nessas tags. Você pode filtrar a lista ao usar as operações de visibilidade. Ao selecionar cuidadosamente as tags que você atribui a uma execução, você pode usá-las para fornecer listagens significativas.

Por exemplo, suponha que você administre vários centros de atendimento. Com as tags, você pode listar os processos que ocorrem em um centro de distribuição específico. Ou, se um cliente estiver convertendo tipos diferentes de arquivos de mídia, as tags podem indicar processos diferentes ao converter arquivos de vídeo, áudio e imagem.

Você pode associar até cinco tags a uma execução de fluxo de trabalho ao iniciar a execução usando a ação `StartWorkflowExecution`, a decisão `StartChildWorkflowExecution` ou a

decisão `ContinueAsNewWorkflowExecution`. Ao usar ações de visibilidade para listar ou contar as execuções do fluxo de trabalho, você pode filtrar os resultados com base nas suas tags.

Para usar a marcação

1. Elabore uma estratégia de marcação. Pense em seus requisitos de negócios e crie uma lista de tags que são úteis para você. Determine quais execuções obterão quais tags. Mesmo que uma execução possa ser atribuída a um máximo de cinco tags, sua biblioteca de tags pode ter qualquer número de tags. Como cada tag pode ser qualquer valor de string com até 256 caracteres de comprimento, uma tag pode descrever quase qualquer conceito de negócios.
2. Marque uma execução com até cinco tags quando você a criar.
3. Liste ou conte as execuções que estão marcadas com uma determinada tag, especificando o parâmetro tagFilter com as ações `ListOpenWorkflowExecutions`, `ListClosedWorkflowExecutions`, `CountOpenWorkflowExecutions` e `CountClosedWorkflowExecutions`. A ação filtrará as execuções com base nas tags especificadas.

Quando você associa uma tag a uma execução de fluxo de trabalho, ela está permanentemente associada a essa execução e não pode ser removida.

Você pode especificar apenas uma tag no parâmetro `tagFilter` com `ListWorkflowExecutions`. Além disso, a correspondência de tags diferencia maiúsculas e minúsculas, e apenas correspondências exatas retornam resultados.

Suponha que você já tenha configurado duas execuções marcadas da seguinte maneira.

Nome da execução	Tags atribuídas
Execution-One	Consumer, 2011-February
Execution-Two	Wholesale, 2011-March

É possível filtrar a lista de execuções retornadas por `ListOpenWorkflowExecutions` na tag Consumer. Os valores `oldestDate` e `latestDate` são especificados como valores de Hora do Unix.

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

```
  RespondDecisionTaskCompleted
  {
    "domain":"867530901",
    "startTimeFilter":{
        "oldestDate":1262332800,
        "latestDate":1325348400
    },
    "tagFilter":{
      "tag":"Consumer"
      }
  }
```

Controle o acesso aos domínios com tags

Você pode controlar o acesso aos domínios do Amazon Simple Workflow Service referenciando tags associadas aos domínios Amazon SWF no IAM. Por exemplo, você pode restringir domínios que incluam uma tag em que a chave está `environment` e o valor é`production`:

```
{
      "Version": "2012-10-17",
      "Statement": [
          {
              "Effect": "Deny",
              "Action": "swf:*",
              "Resource": "arn:aws:swf:*:123456789012:/domain/*",
              "Condition": {
                  "StringEquals": {"aws:ResourceTag/environment": "production"}
              }
          }
      ]
  }
```

Esta política `Deny` o acesso a todos os domínios que foram marcados como `environment/production`.

Para obter mais informações, consulte:

- Controlar o acesso com tags do IAM
- Políticas baseadas em tag

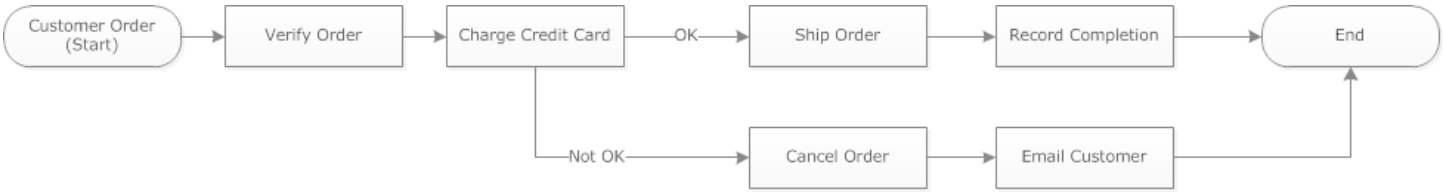
Implementando opções exclusivas com o Amazon SWF

Em alguns cenários, convém agendar um conjunto diferente de atividades com base no resultado de uma atividade anterior. Com o padrão de escolha exclusivo, você pode criar fluxos de trabalho flexíveis que atendam aos requisitos complexos do seu aplicativo.

O Amazon SWF não tem uma ação de escolha exclusiva específica. Para implementar a escolha exclusiva, você deve escrever sua lógica decisória para tomar decisões com base nos resultados de uma atividade anterior. Alguns aplicações para escolha exclusiva incluem:

- Realizar atividades de limpeza quando os resultados de uma atividade anterior não tiveram êxito
- Agendar atividades diferentes com base em se o cliente adquiriu um plano básico ou avançado
- Realizar diferentes atividades de autenticação de cliente com base no histórico de pedidos de um cliente

No exemplo de comércio eletrônico, você pode usar a escolha exclusiva para enviar ou cancelar um pedido com base no resultado da cobrança do cartão de crédito. Na figura a seguir, o agente de decisão agenda as tarefas de atividade Enviar o pedido e Registrar a conclusão quando o cartão de crédito é debitado com êxito. Caso contrário, ele agenda as tarefas de atividade Cancelar o pedido e Enviar e-mail para o cliente.



O agente de decisão agenda a atividade `ShipOrder` quando o cartão de crédito é debitado com êxito. Caso contrário, ele agenda a atividade `CancelOrder`.

Nesse caso, programe o agente de decisão para interpretar o histórico e determinar se o cartão de crédito foi cobrado com êxito. Para fazer isso, você pode ter uma lógica semelhante à seguinte

```
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
```

Se o cartão de crédito tiver sido debitado com êxito, o agente de decisão deverá responder com `RespondDecisionTaskCompleted` para agendar a atividade `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"
            }
        }
    ]
}
```

Se o cartão de crédito não tiver sido debitado com êxito, o agente de decisão deverá responder com `RespondDecisionTaskCompleted` para agendar a atividade `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"
          }
      }
  ]
```

```
}
```

Se o Amazon SWF for capaz de validar os dados na ação `RespondDecisionTaskCompleted`, o Amazon SWF retornará uma resposta HTTP bem-sucedida semelhante à seguinte.

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

Temporizadores no Amazon SWF

Com um cronômetro, você pode notificar seu tomador quando um determinado período de tempo tiver decorrido.

Ao responder a uma tarefa de decisão, o agente de decisão tem a opção de responder com uma decisão `StartTimer`. Essa decisão especifica uma quantidade de tempo após a qual o temporizador deve ser acionado. Depois de decorrido o tempo especificado, o Amazon SWF adicionará um evento `TimerFired` ao histórico de execução do fluxo de trabalho e agendará uma tarefa de decisão. O agente de decisão pode então usar essas informações para informar decisões adicionais. Uma aplicação comum para um temporizador é para atrasar a execução de uma tarefa de atividade. Por exemplo, um cliente pode optar pela entrega de um item.

Cancelamento de tarefas de atividade no Amazon SWF

O cancelamento da tarefa da atividade informa o decisor a encerrar as atividades que não precisam mais ser realizadas. O Amazon SWF usa um mecanismo de cancelamento cooperativo e não interrompe forçosamente a execução de tarefas de atividade. Você deve programar seus operadores de atividade para lidar com solicitações de cancelamento.

O agente de decisão pode optar por cancelar uma tarefa de atividade enquanto está processando uma tarefa de decisão. Para cancelar uma tarefa de atividade, o agente de decisão usa a ação `RespondDecisionTaskCompleted` com a decisão `RequestCancelActivityTask`.

Se a tarefa de atividade ainda não tiver sido adquirida por um operador de atividade, o serviço cancelará a tarefa. Observe que existe uma possível condição de corrida em que um operador de atividade poderia adquirir a tarefa a qualquer momento. Se a tarefa já tiver sido atribuída a um operador de atividade, este será solicitado a cancelá-la.

Neste exemplo, a execução de fluxo de trabalho recebe um sinal para cancelar o 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"}
```

Se houver sinal na execução do fluxo de trabalho, o Amazon SWF retornará uma resposta HTTP bem-sucedida similar ao seguinte: O Amazon SWF gerará uma tarefa de decisão para informar o tomador de decisão a processar o sinal.

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

Quando o agente de decisão processar a tarefa de decisão e vir o sinal no histórico, ele tentará cancelar a atividade pendente que possui o ID de atividade `ShipOrderActivity0001`. O ID de atividade é fornecido no histórico de fluxo de trabalho originado do evento de agendamento de tarefa de atividade.

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

Se o Amazon SWF receber com êxito a solicitação de cancelamento, ele retornará uma resposta HTTP bem-sucedida semelhante à seguinte:

```
HTTP/1.1 200 OK
```

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

A tentativa de cancelamento é registrada no histórico como o evento `ActivityTaskCancelRequested`.

Se a tarefa for cancelada com êxito, conforme indicado por um evento `ActivityTaskCanceled`, programe seu agente de decisão para executar as etapas apropriadas que devem seguir o cancelamento da tarefa, como encerrar a execução do fluxo de trabalho.

Se a tarefa da atividade não puder ser cancelada (por exemplo, se a tarefa for concluída, falhar ou atingir o tempo limite em vez de ser cancelada), o agente de decisão deverá aceitar os resultados da atividade ou realizar qualquer limpeza ou mitigação necessária para o seu caso de uso.

Se a tarefa de atividade já tiver sido adquirida por um operador de atividade, a solicitação de cancelamento será transmitida por meio do mecanismo de heartbeat de tarefa. Os operadores de atividade podem usar periodicamente `RecordActivityTaskHeartbeat` para informar ao Amazon SWF que a tarefa ainda está em andamento.

Observe que operadores de atividade não são obrigados a realizar heartbeats, embora isso seja recomendado para tarefas de longa duração. O cancelamento da tarefa requer que um heartbeat periódico seja registrado. Se o operador não realizar o heartbeat, a tarefa não poderá ser cancelada.

Se o agente de decisão solicitar o cancelamento da tarefa, o Amazon SWF definirá o valor do objeto `cancelRequest` como true. O objeto `cancelRequest` faz parte do objeto `ActivityTaskStatus` que é retornado pelo serviço em resposta a `RecordActivityTaskHeartbeat`.

O Amazon SWF não impede a conclusão bem-sucedida de uma tarefa de atividade cujo cancelamento tenha sido solicitado; cabe à atividade determinar como lidar com a solicitação de cancelamento. Dependendo dos seus requisitos, programe o operador de atividade para cancelar a tarefa de atividade ou ignorar a solicitação de cancelamento.

Se quiser que o operador de atividade indique que o trabalho da tarefa de atividade foi cancelado, programe-o para responder com um `RespondActivityTaskCanceled`. Se quiser que o operador de atividade conclua a tarefa, programe-a para responder com um `RespondActivityTaskCompleted` padrão.

Quando o Amazon SWF recebe a solicitação `RespondActivityTaskCompleted` ou `RespondActivityTaskCanceled`, ele atualiza o histórico de execução do fluxo de trabalho e agenda uma tarefa de decisão para informar o agente de decisão.

Programe o agente de decisão para processar a tarefa de decisão e retornar decisões adicionais. Se a tarefa de atividade tiver sido cancelada com êxito, programe o agente de decisão para realizar as tarefas necessárias para continuar ou encerrar a execução de fluxo de trabalho. Se a tarefa de atividade não tiver sido cancelada com êxito, programe o agente de decisão para aceitar os resultados, ignorar os resultados ou agendar qualquer limpeza necessária.

Segurança no Amazon Simple Workflow Service

Esta seção fornece informações sobre a segurança e a autenticação do Amazon Simple Workflow Service.

Tópicos

- Proteção de dados no Amazon Simple Workflow Service
- Gerenciamento de identidade e acesso no Amazon Simple Workflow Service
- Registro e Monitoramento
- Validação de conformidade para o Amazon Simple Workflow Service
- Resiliência no Amazon Simple Workflow Service
- Segurança de infraestrutura no Amazon Simple Workflow Service
- Configuração e análise de vulnerabilidade no Amazon Simple Workflow Service

O Amazon SWF usa o IAM para controlar o acesso a outros AWS serviços e recursos. Para obter uma visão geral de como o IAM funciona, consulte Visão geral do gerenciamento de acesso no Guia do usuário do IAM. Para obter uma visão geral das credenciais de segurança, consulte Credenciais de segurança da AWS na Referência geral da Amazon Web Services.

Proteção de dados no Amazon Simple Workflow Service

O modelo de responsabilidade AWS compartilhada se aplica à proteção de dados no Amazon Simple Workflow Service. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Amazon SWF ou outro Serviços da AWS usando o console, a API ou. AWS CLI AWS SDKs Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Criptografia no Amazon Simple Workflow Service

Criptografia em repouso

O Amazon SWF sempre criptografa seus dados em repouso. Os dados no Amazon Simple Workflow Service são criptografados em repouso usando criptografia transparente do lado do servidor. Isso ajuda a reduzir a carga e a complexidade operacionais necessárias para proteger dados confidenciais. Com a criptografia em repouso, é possível criar aplicativos que priorizam a segurança e que atendem a requisitos de conformidade e regulamentação de criptografia.

Criptografia em trânsito

Todos os dados que passam entre o Amazon SWF e outros serviços são criptografados usando o Transport Layer Security (TLS).

Gerenciamento de identidade e acesso no Amazon Simple Workflow Service

O acesso ao Amazon SWF requer credenciais que AWS possam ser usadas para autenticar suas solicitações. Essas credenciais devem ter permissões para acessar AWS recursos, como recuperar dados de eventos de outros AWS recursos. As seções a seguir fornecem detalhes sobre como você pode usar o AWS Identity and Access Management (IAM) e o Amazon SWF para ajudar a proteger seus recursos, controlando o acesso a eles.

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar os recursos do Amazon SWF. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Público
- Autenticar com identidades
- Gerenciar o acesso usando políticas
- Controle de acesso
- Ações de política para o Amazon SWF
- Recursos de políticas para o Amazon SWF
- Chaves de condição de política para o Amazon SWF
- ACLs no Amazon SWF
- ABAC com o Amazon SWF
- Usar credenciais temporárias com o Amazon SWF
- Permissões de entidade principal entre serviços para o Amazon SWF
- Perfis de serviço para o Amazon SWF
- Funções vinculadas a serviços para o Amazon SWF

- Políticas baseadas em identidade para o Amazon SWF
- Políticas baseadas em recursos dentro do Amazon SWF
- Como o Amazon Simple Workflow Service funciona com o IAM
- Exemplos de políticas baseadas em identidade para o Amazon Simple Workflow Service
- Princípios básicos
- Políticas do IAM do Amazon SWF
- Resumo de APIs
- Políticas baseadas em tag
- Endpoints da Amazon VPC para o Amazon SWF
- Solução de problemas de identidade e acesso do Amazon Simple Workflow Service

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no Amazon SWF.

Usuário do serviço: se você usar o serviço Amazon SWF para fazer seu trabalho, o administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usar mais recursos do Amazon SWF para fazer seu trabalho, poderá precisar de permissões adicionais. Compreenda como o acesso é gerenciado pode ajudar a solicitar as permissões corretas ao administrador. Se não for possível acessar um recurso no Amazon SWF, consulte Solução de problemas de identidade e acesso do Amazon Simple Workflow Service.

Administrador de serviços: se você é responsável pelos recursos do Amazon SWF em sua empresa, provavelmente tem acesso total ao Amazon SWF. É seu trabalho determinar quais recursos e funcionalidades do Amazon SWF os usuários do seu serviço devem acessar. Envie as solicitações ao administrador do IAM para alterar as permissões dos usuários de serviço. Revise as informações nesta página para compreender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar o IAM com o Amazon SWF, consulte Como o Amazon Simple Workflow Service funciona com o IAM.

Administrador do IAM: se você for um administrador do IAM, talvez queira saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao Amazon SWF. Para ver exemplos de políticas baseadas em identidade do Amazon SWF que você pode usar no IAM, consulte Exemplos de políticas baseadas em identidade para o Amazon Simple Workflow Service.

Autenticar com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para designar solicitações por conta própria, consulte Versão 4 do AWS Signature para solicitações de API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia do usuário do AWS IAM Identity Center e Usar a autenticação multifator da AWS no IAM no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do Usuário do IAM.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte O que é o Centro de Identidade do IAM? no Guia do Usuário do AWS IAM Identity Center .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Casos de uso para usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Para assumir temporariamente uma função do IAM no AWS Management Console, você pode alternar de um usuário para uma função do IAM (console). Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte Métodos para assumir um perfil no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidade de terceiros (federação) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Guia do Usuário do AWS IAM Identity Center .
- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service

(Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

- Perfil de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo solicitações AWS CLI de AWS API. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte Usar uma função do IAM para conceder permissões a aplicativos executados em EC2 instâncias da Amazon no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte Políticas de controle de serviços no Guia AWS Organizations do Usuário.
- Políticas de controle de recursos (RCPs) — RCPs são políticas JSON que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as políticas do IAM anexadas a cada recurso que você possui. O RCP limita as permissões para recursos nas contas dos membros e pode afetar as permissões efetivas para identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da

AWS desse suporte RCPs, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.

- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Controle de acesso

Você pode ter credenciais válidas para autenticar suas solicitações, mas, a menos que tenha permissões, não poderá criar ou acessar os recursos do Amazon SWF. Por exemplo, você deve ter permissões para invocar AWS Lambda alvos do Amazon Simple Notification Service (Amazon SNS) e do Amazon Simple Queue Service (Amazon SQS) associados às suas regras do Amazon SWF.

As seções a seguir descrevem como gerenciar permissões para o Amazon SWF. Recomendamos que você leia a visão geral primeiro.

- Princípios básicos
- Políticas do IAM do Amazon SWF
- Exemplos de política do Amazon SWF

Ações de política para o Amazon SWF

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações do Amazon SWF, consulte Recursos definidos pelo Amazon Simple Workflow Service na Referência de autorização de serviço.

As ações de política no Amazon SWF usam o seguinte prefixo antes da ação:

```
swf
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

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

Para ver exemplos de políticas baseadas em identidade do Amazon SWF, consulte Exemplos de políticas baseadas em identidade para o Amazon Simple Workflow Service.

Recursos de políticas para o Amazon SWF

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de recursos do Amazon SWF e seus ARNs, consulte Ações definidas pelo Amazon Simple Workflow Service na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Recursos definidos pelo Amazon Simple Workflow Service.

Para ver exemplos de políticas baseadas em identidade do Amazon SWF, consulte Exemplos de políticas baseadas em identidade para o Amazon Simple Workflow Service.

Chaves de condição de política para o Amazon SWF

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista das chaves de condição do Amazon SWF, consulte Chaves de condição para o Amazon Simple Workflow Service na Referência de autorização de serviço. Para saber com quais

ações e recursos você pode usar uma chave de condição, consulte Recursos definidos pelo Amazon Simple Workflow Service.

Para ver exemplos de políticas baseadas em identidade do Amazon SWF, consulte Exemplos de políticas baseadas em identidade para o Amazon Simple Workflow Service.

ACLs no Amazon SWF

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o Amazon SWF

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define as permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. Marcar de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Amazon SWF

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS "Trabalhe com o IAM" no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte Alternar para um perfil do IAM (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias no IAM.

Permissões de entidade principal entre serviços para o Amazon SWF

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado um principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Perfis de serviço para o Amazon SWF

Compatível com perfis de serviço: sim

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amazon SWF. Edite os perfis de serviço somente quando o Amazon SWF fornecer orientação para isso.

Funções vinculadas a serviços para o Amazon SWF

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Políticas baseadas em identidade para o Amazon SWF

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amazon SWF

Para ver exemplos de políticas baseadas em identidade do Amazon SWF, consulte Exemplos de políticas baseadas em identidade para o Amazon Simple Workflow Service.

Políticas baseadas em recursos dentro do Amazon SWF

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Como o Amazon Simple Workflow Service funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon SWF, saiba quais recursos do IAM estão disponíveis para uso com o Amazon SWF.

Recursos do IAM que você pode usar com o Amazon Simple Workflow Service

Atributo do IAM	Suporte ao Amazon SWF
Políticas baseadas em identidade	Sim

Atributo do IAM	Suporte ao Amazon SWF
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados a serviço	Não

Para ter uma visão de alto nível de como o Amazon SWF e AWS outros serviços funcionam com a maioria dos recursos do IAM, AWS consulte os serviços que funcionam com o IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amazon Simple Workflow Service

Por padrão, os usuários e as funções não têm permissão para criar ou modificar recursos do Amazon SWF. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo Amazon SWF, incluindo o formato de cada um dos tipos de recursos, consulte Ações, recursos e chaves de condição ARNs para o Amazon Simple Workflow Service na Referência de autorização de serviço.

Tópicos

- Práticas recomendadas de política
- Usar o console do Amazon SWF
- Permitir que os usuários visualizem suas próprias permissões

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon SWF em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas

sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.

- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usar o console do Amazon SWF

Para acessar o console do Amazon Simple Workflow Service, é necessário ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amazon SWF em seu. Conta da AWS Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o console do Amazon SWF, anexe também o Amazon `ConsoleAccess` SWF `ReadOnly` AWS ou a política gerenciada às entidades. Para obter informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

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

```
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Princípios básicos

O controle de acesso do Amazon SWF é baseado principalmente em dois tipos de permissões:

- Permissões de recursos: Quais recursos do Amazon SWF um usuário pode acessar.

  Você pode expressar permissões de recursos apenas para domínios.
- Permissões de API: Quais ações do Amazon SWF um usuário pode chamar.

A abordagem mais simples é conceder acesso total à conta (chamar qualquer ação do Amazon SWF em qualquer domínio) ou negar totalmente o acesso. No entanto, o IAM oferece suporte a uma

abordagem mais granular do controle de acesso, que geralmente é mais útil. Por exemplo, você pode:

- Permitir que um usuário chame qualquer ação do Amazon SWF sem restrições, mas somente em um domínio especificado. Essa política pode ser usada para permitir que aplicativos de fluxo de trabalho em desenvolvimento usem qualquer ação, mas apenas um domínio de "sandbox".
- Permitir que um usuário acesse qualquer domínio, mas restringir como eles usam a API. Essa política pode ser usada para permitir que um aplicativo "auditor" chame a API em qualquer domínio, mas permita apenas acesso de leitura.
- Permitir que um usuário chame apenas um conjunto limitado de ações em determinados domínios. Você poderia usar essa política para permitir que um iniciador de fluxo de trabalho chamasse apenas a ação `StartWorkflowExecution` em um domínio especificado.

O controle de acesso do Amazon SWF é baseado nos seguintes princípios:

- As decisões de controle de acesso baseiam-se apenas nas políticas do IAM; toda a auditoria e manipulação de políticas são feitas por meio do IAM.
- O modelo de controle de acesso usa uma deny-by-default política; qualquer acesso que não seja explicitamente permitido é negado.
- Você controla o acesso aos recursos do Amazon SWF anexando políticas do IAM apropriadas aos atores do fluxo de trabalho.
- Permissões de recursos podem ser expressas somente para domínios.
- Você pode restringir o uso de algumas ações aplicando condições a um ou mais parâmetros.
- Se você conceder permissão de uso RespondDecisionTaskCompleted, poderá expressar permissões para a lista de decisões incluídas nessa ação.

  Cada uma das decisões tem um ou mais parâmetros, como uma chamada de API regular. Para permitir as políticas mais legíveis possíveis, você pode expressar permissões em decisões como se fossem chamadas de API reais, incluindo a aplicação de condições a alguns parâmetros. Esses tipos de permissões são chamados de permissões de pseudo API.

Para um resumo de quais parâmetros regulares e parâmetros de pseudo API podem ser restritos usando condições, consulte Resumo de APIs.

Políticas do IAM do Amazon SWF

Uma política do IAM contém um ou mais elementos `Statement`, cada um dos quais contém um conjunto de elementos que definem a política. Para obter uma lista completa de elementos e uma discussão geral sobre como estruturar políticas, consulte A linguagem da política de acesso. O controle de acesso do Amazon SWF é baseado nos seguintes elementos:

Efeito

(Obrigatório) O efeito da instrução: `deny` ou `allow`.

> **Note**
>
> Você deve permitir explicitamente o acesso; o IAM nega o acesso por padrão.

Recurso

(Obrigatório) O recurso — uma entidade em um AWS serviço com a qual o usuário pode interagir — ao qual a declaração se aplica.

Você pode expressar permissões de recursos apenas para domínios. Por exemplo, uma política pode permitir acesso a apenas certos domínios na sua conta. Para expressar permissões para um domínio, `Resource` defina o Amazon Resource Name (ARN) do domínio, que tem o formato “arn:aws:swf::: /domain/”. *`Region`* *`AccountID`* *`DomainName`* *`Region`*é a AWS região, *`AccountID`* é o ID da conta sem traços e *`DomainName`* é o nome do domínio.

Ação

(Obrigatório) A ação à qual a declaração se aplica, à qual você se refere usando o seguinte formato:*`serviceId`*:*`action`*. Para Amazon SWF, *`serviceID`* defina como. `swf` Por exemplo, `swf:StartWorkflowExecution` refere-se à StartWorkflowExecutionação e é usado para controlar quais usuários têm permissão para iniciar fluxos de trabalho.

Se você conceder permissão de uso RespondDecisionTaskCompleted, também poderá controlar o acesso à lista de decisões incluída usando `Action` para expressar permissões para a pseudo-API. Como o IAM nega o acesso por padrão, a decisão de um agente de decisão deve ser explicitamente permitida ou não será aceita. Você pode usar um valor `*` para permitir todas as decisões.

Condição

(Opcional) Expressa uma restrição em um ou mais dos parâmetros de uma ação, o que restringe os valores permitidos.

As ações do Amazon SWF geralmente têm um escopo amplo, que você pode reduzir usando as condições do IAM. Por exemplo, para limitar quais listas de tarefas a PollForActivityTaskação pode acessar, você inclui um `Condition` e usa a `swf:taskList.name` chave para especificar as listas permitidas.

É possível expressar restrições para as seguintes entidades.

- O tipo de fluxo de trabalho. O nome e a versão têm chaves separadas.
- O tipo de atividade. O nome e a versão têm chaves separadas.
- Listas de tarefas.
- Tags. É possível especificar várias tags para algumas ações. Nesse caso, cada tag tem uma chave separada.

> **Note**
>
> Para o Amazon SWF, os valores são todos strings, portanto, você restringe um parâmetro usando um operador de string, como `StringEquals`, que restringe o parâmetro a uma string especificada. No entanto, os operadores de comparação de string regulares, como `StringEquals`, exigem que todas as solicitações incluam o parâmetro. Se você não incluir o parâmetro explicitamente, e não houver um valor padrão, como a lista de tarefas padrão fornecida durante o registro de tipo, o acesso será negado.
> Muitas vezes, é útil tratar condições como opcionais, para que você possa chamar uma ação sem necessariamente incluir o parâmetro associado. Por exemplo, talvez você queira permitir que um decisor especifique um conjunto de RespondDecisionTaskCompleteddecisões, mas também permita que ele especifique somente uma delas para qualquer chamada específica. Nesse caso, você restringe os parâmetros apropriados usando um operador `StringEqualsIfExists`, o que permite o acesso se o parâmetro atender à condição, mas não nega o acesso se o parâmetro estiver ausente.

Para obter uma lista completa de parâmetros restritivos e as chaves associadas, consulte Resumo de APIs.

A seção a seguir fornece exemplos de como estruturar políticas do Amazon SWF. Para obter detalhes, consulte Condições de string.

Exemplos de política do Amazon SWF

Um fluxo de trabalho consiste em vários atores: atividades, agentes de decisão e assim por diante. Você pode controlar o acesso de cada ator anexando uma política do IAM apropriada. Esta seção fornece alguns exemplos. Veja a seguir o caso mais simples:

```
{
   "Version": "2012-10-17",
   "Statement" : [ {
      "Effect" : "Allow",
      "Action" : "swf:*",
      "Resource" : "arn:aws:swf:*:123456789012:/domain/*"
   } ]
}
```

Se você anexar essa política a um atuador, ele terá acesso total a todas as regiões. Você pode usar curingas para que um único valor represente vários recursos, ações ou regiões.

- O primeiro curinga (`*`) no valor `Resource` indica que as permissões de recursos se aplicam a todas as regiões. Para restringir as permissões a uma única região, substitua o curinga pela string da região apropriada, como us-east-1.
- O segundo curinga (`*`) no valor `Resource` permite que o atuador acesse qualquer um dos domínios da conta nas regiões especificadas.
- O caractere curinga (`*`) no valor de `Action` permite que o ator chame qualquer ação do Amazon SWF.

Para obter detalhes sobre como usar curingas, consulte Descrições de elementos

As seções a seguir mostram exemplos de políticas que concedem permissões de forma mais granular.

Permissões de domínio

Se você quiser restringir os fluxos de trabalho de um departamento a um domínio específico, pode usar algo como:

```
{
```

```
   "Version": "2012-10-17",
   "Statement": [ {
      "Effect" : "Allow",
      "Action" : "swf:*",
      "Resource" : "arn:aws:swf:*:123456789012:/domain/department1"
   } ]
}
```

Se você anexar essa política a um atuador, ele poderá chamar qualquer ação, mas somente para o domínio department1.

Se quiser que um atuador tenha acesso a mais de um domínio, você poderá expressar permissão para cada domínio separadamente, da seguinte forma:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect" : "Allow",
         "Action" : "swf:*",
         "Resource" : "arn:aws:swf:*:123456789012:/domain/department1"
      }, {
         "Effect" : "Allow",
         "Action" : "swf:*",
         "Resource" : "arn:aws:swf:*:123456789012:/domain/department2"
      }
   ]
}
```

Se você anexar essa política a um ator, ele poderá usar qualquer ação do Amazon SWF nos domínios `department1` e `department2`. Você também pode usar curingas para representar vários domínios.

Permissões e restrições da API

Você controla as ações que um atuador pode usar com o elemento `Action`. Opcionalmente, é possível restringir os valores de parâmetros permitidos da ação usando um elemento `Condition`.

Se quiser restringir um atuador apenas a determinadas ações, poderá usar algo como o seguinte:

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

```
   "Statement": [ {
      "Effect" : "Allow",
      "Action" : "swf:StartWorkflowExecution",
      "Resource" : "arn:aws:swf:*:123456789012:/domain/department2"
   } ]
}
```

Se você anexar essa política a um atuador, ele poderá chamar `StartWorkflowExecution` para iniciar fluxos de trabalho no domínio `department2`. Ele não poderá usar outras ações ou iniciar fluxos de trabalho em outros domínios.

Você pode restringir ainda mais quais fluxos de trabalho um ator pode iniciar, restringindo um ou mais dos valores de parâmetros `StartWorkflowExecution`, da seguinte forma:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect" : "Allow",
         "Action" : "swf:StartWorkflowExecution",
         "Resource" : "arn:aws:swf:*:123456789012:/domain/department1",
         "Condition" : {
            "StringEquals" : {
               "swf:workflowType.name" : "workflow1",
               "swf:workflowType.version" : "version2"
            }
         }
      }
   ]
}
```

Essa política restringe os parâmetros `name` e `version` da ação `StartWorkflowExecution`. Se você anexar a política a um atuador, ele poderá executar apenas `version2` de `workflow1` no domínio `department1`, e ambos os parâmetros deverão ser incluídos na solicitação.

Você pode restringir um parâmetro sem exigir que ele seja incluído em uma solicitação, usando um operador `StringEqualsIfExists`, da seguinte maneira:

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

```
      "Effect" : "Allow",
      "Action" : "swf:StartWorkflowExecution",
      "Resource" : "arn:aws:swf:*:123456789012:/domain/some_domain",
      "Condition" : {
         "StringEqualsIfExists" : { "swf:taskList.name" : "task_list_name" }
      }
   } ]
}
```

Essa política permite que um atuador especifique opcionalmente uma lista de tarefas ao iniciar uma execução de fluxo de trabalho.

É possível restringir uma lista de tags para algumas ações. Nesse caso, cada tag tem uma chave separada e, portanto, você usa `swf:tagList.member.0` para restringir a primeira tag na lista, `swf:tagList.member.1` para restringir a segunda tag na lista e assim por diante, até um máximo de 5. No entanto, você deve ter cuidado em termos de como restringir listas de tags. Veja a seguir um exemplo de uma política que não é recomendada:

```
{
   "Version": "2012-10-17",
   "Statement" : [ {
      "Effect" : "Allow",
      "Action" : "swf:StartWorkflowExecution",
      "Resource" : "arn:aws:swf:*:123456789012:/domain/some_domain",
      "Condition" : {
         "StringEqualsIfExists" : {
            "swf:tagList.member.0" : "some_ok_tag", "another_ok_tag"
         }
      }
   } ]
}
```

Essa política permite que você especifique opcionalmente `some_ok_tag` ou `another_ok_tag`. No entanto, ela restringe apenas o primeiro elemento da lista de tags. A lista pode ter elementos adicionais com valores arbitrários que seriam todos permitidos, já que essa política não aplica condições a `swf:tagList.member.1`, `swf:tagList.member.2` e assim por diante.

Uma maneira de resolver esse problema é desativar o uso de listas de tags. A seguinte política garante que somente `some_ok_tag` ou `another_ok_tag` seja permitido, exigindo que a lista tenha apenas um elemento.

```
{
   "Version": "2012-10-17",
   "Statement" : [ {
      "Effect" : "Allow",
      "Action" : "swf:StartWorkflowExecution",
      "Resource" : "arn:aws:swf:*:123456789012:/domain/some_domain",
      "Condition" : {
         "StringEqualsIfExists" : {
            "swf:tagList.member.0" : "some_ok_tag", "another_ok_tag"
         },
         "Null" : { "swf:tagList.member.1" : "true" }
      }
   } ]
}
```

Permissões e restrições da pseudo API

Se você quiser restringir as decisões disponíveis ao `RespondDecisionTaskCompleted`, primeiro deverá permitir que o atuador chame `RespondDecisionTaskCompleted`. Em seguida, você poderá expressar permissões para os membros apropriados da pseudo API usando a mesma sintaxe da API normal, da seguinte maneira:

```
{
   "Version": "2012-10-17",
   "Statement" : [
      {
         "Resource" : "arn:aws:swf:*:123456789012:/domain/*",
         "Action" : "swf:RespondDecisionTaskCompleted",
         "Effect" : "Allow"
      }, {
         "Resource" : "*",
         "Action" : "swf:ScheduleActivityTask",
         "Effect" : "Allow",
         "Condition" : {
            "StringEquals" : { "swf:activityType.name" : "SomeActivityType" }
         }
      }
   ]
}
```

Se você anexar essa política a um atuador, o primeiro elemento `Statement` permitirá que o atuador chame `RespondDecisionTaskCompleted`. O segundo elemento permite que o ator use a decisão

`ScheduleActivityTask` para direcionar o Amazon SWF para agendar uma tarefa de atividade. Para permitir todas as decisões, substitua “swf:ScheduleActivityTask" por “swf: *”.

É possível operadores de condição para restringir parâmetros, como é o caso com a API regular. O operador `StringEquals` nesse `Condition` permite que `RespondDecisionTaskCompleted` agende uma tarefa de atividade para a atividade `SomeActivityType`, e ele deve agendar essa tarefa. Se você quiser permitir que `RespondDecisionTaskCompleted` use um valor de parâmetro, mas não exige que ele faça isso, poderá usar o operador `StringEqualsIfExists`.

AWS política gerenciada: SimpleWorkflowFullAccess

É possível anexar a política `SimpleWorkflowFullAccess` às identidades do IAM.

Essa política fornece acesso total ao serviço de configuração do Amazon SWF.

Detalhes da permissão

Esta política inclui as seguintes permissões.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "swf:*"
         ],
         "Resource": "*"
      }
   ]
}
```

Limitações do modelo de serviço nas políticas do IAM

Você deve considerar as restrições do modelo de serviço ao criar políticas do IAM. É possível criar uma política do IAM sintaticamente válida que represente uma solicitação inválida do Amazon SWF; uma solicitação permitida em termos de controle de acesso ainda pode falhar por ser uma solicitação inválida.

Por exemplo, a seguinte política para `ListOpenWorkflowExecutions` não é recomendada:

```
{
   "Version": "2012-10-17",
   "Statement" : [ {
      "Effect" : "Allow",
      "Action" : "swf:ListOpenWorkflowExecutions",
      "Resource" : "arn:aws:swf:*:123456789012:/domain/domain_name",
      "Condition" : {
         "StringEquals" : {
            "swf:typeFilter.name" : "workflow_name",
            "swf:typeFilter.version" : "workflow_version",
            "swf:tagFilter.tag" : "some_tag"
         }
      }
   } ]
}
```

O modelo de serviço do Amazon SWF não permite que os parâmetros `typeFilter` e `tagFilter` sejam usados na mesma solicitação `ListOpenWorkflowExecutions`. Portanto, a política permite chamadas que o serviço rejeitará, lançando `ValidationException` como uma solicitação inválida.

Resumo de APIs

Esta seção descreve brevemente como você pode usar as políticas do IAM para controlar como um ator pode usar cada API e pseudo API para acessar os recursos do Amazon SWF.

- Para todas as ações, exceto `RegisterDomain` e `ListDomains`, você pode permitir ou negar acesso a qualquer ou todos os domínios de uma conta, expressando permissões para o recurso de domínio.
- Você pode permitir ou negar permissão para qualquer membro da API regular e, se conceder permissão para chamar `RespondDecisionTaskCompleted`, para qualquer membro da pseudo API.
- É possível usar uma Condição para restringir os valores permitidos de alguns parâmetros.

As seções a seguir listam os parâmetros que podem ser restritos para cada membro da API regular e da pseudo API e fornecem a chave associada, além de mencionarem todas as limitações sobre como você pode controlar o acesso ao domínio.

API regular

Esta seção lista os membros da API regular e descreve brevemente os parâmetros que podem ser restritos e as chaves associadas. Ela também menciona quaisquer limitações sobre como você pode controlar o acesso ao domínio.

`CountClosedWorkflowExecutions`

- `tagFilter.tag`: restrição de string. A chave é `swf:tagFilter.tag`
- `typeFilter.name`: restrição de string. A chave é `swf:typeFilter.name`.
- `typeFilter.version`: restrição de string. A chave é `swf:typeFilter.version`.

> **Note**
>
> `CountClosedWorkflowExecutions` requer que `typeFilter` e `tagFilter` sejam mutuamente exclusivos.

`CountOpenWorkflowExecutions`

- `tagFilter.tag`: restrição de string. A chave é `swf:tagFilter.tag`
- `typeFilter.name`: restrição de string. A chave é `swf:typeFilter.name`.
- `typeFilter.version`: restrição de string. A chave é `swf:typeFilter.version`.

> **Note**
>
> `CountOpenWorkflowExecutions` requer que `typeFilter` e `tagFilter` sejam mutuamente exclusivos.

`CountPendingActivityTasks`

- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.

`CountPendingDecisionTasks`

- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.

DeleteActivityType

- `activityType.name`: restrição de string. A chave é `swf:activityType.name`.
- `activityType.version`: restrição de string. A chave é `swf:activityType.version`.

DeprecateActivityType

- `activityType.name`: restrição de string. A chave é `swf:activityType.name`.
- `activityType.version`: restrição de string. A chave é `swf:activityType.version`.

DeprecateDomain

- Não é possível restringir os parâmetros dessa ação.

DeleteWorkflowType

- `workflowType.name`: restrição de string. A chave é `swf:workflowType.name`.
- `workflowType.version`: restrição de string. A chave é `swf:workflowType.version`.

DeprecateWorkflowType

- `workflowType.name`: restrição de string. A chave é `swf:workflowType.name`.
- `workflowType.version`: restrição de string. A chave é `swf:workflowType.version`.

DescribeActivityType

- `activityType.name`: restrição de string. A chave é `swf:activityType.name`.
- `activityType.version`: restrição de string. A chave é `swf:activityType.version`.

DescribeDomain

- Não é possível restringir os parâmetros dessa ação.

DescribeWorkflowExecution

- Não é possível restringir os parâmetros dessa ação.

DescribeWorkflowType

- `workflowType.name`: restrição de string. A chave é `swf:workflowType.name`.
- `workflowType.version`: restrição de string. A chave é `swf:workflowType.version`.

GetWorkflowExecutionHistory

- Não é possível restringir os parâmetros dessa ação.

ListActivityTypes

- Não é possível restringir os parâmetros dessa ação.

ListClosedWorkflowExecutions

- `tagFilter.tag`: restrição de string. A chave é `swf:tagFilter.tag`
- `typeFilter.name`: restrição de string. A chave é `swf:typeFilter.name`.
- `typeFilter.version`: restrição de string. A chave é `swf:typeFilter.version`.

> **Note**
>
> `ListClosedWorkflowExecutions` requer que `typeFilter` e `tagFilter` sejam mutuamente exclusivos.

ListDomains

- Não é possível restringir os parâmetros dessa ação.

ListOpenWorkflowExecutions

- `tagFilter.tag`: restrição de string. A chave é `swf:tagFilter.tag`
- `typeFilter.name`: restrição de string. A chave é `swf:typeFilter.name`.
- `typeFilter.version`: restrição de string. A chave é `swf:typeFilter.version`.

> **Note**
>
> `ListOpenWorkflowExecutions` requer que `typeFilter` e `tagFilter` sejam mutuamente exclusivos.

ListWorkflowTypes

- Não é possível restringir os parâmetros dessa ação.

PollForActivityTask

- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.

PollForDecisionTask

- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.

RecordActivityTaskHeartbeat

- Não é possível restringir os parâmetros dessa ação.

RegisterActivityType

- `defaultTaskList.name`: restrição de string. A chave é `swf:defaultTaskList.name`.
- `name`: restrição de string. A chave é `swf:name`.
- `version`: restrição de string. A chave é `swf:version`.

RegisterDomain

- `name`: o nome do domínio que está sendo registrado está disponível como o recurso dessa ação.

RegisterWorkflowType

- `defaultTaskList.name`: restrição de string. A chave é `swf:defaultTaskList.name`.
- `name`: restrição de string. A chave é `swf:name`.

- `version`: restrição de string. A chave é `swf:version`.

RequestCancelWorkflowExecution

- Não é possível restringir os parâmetros dessa ação.

RespondActivityTaskCanceled

- Não é possível restringir os parâmetros dessa ação.

RespondActivityTaskCompleted

- Não é possível restringir os parâmetros dessa ação.

RespondActivityTaskFailed

- Não é possível restringir os parâmetros dessa ação.

RespondDecisionTaskCompleted

- `decisions.member.N`: restrito indiretamente por meio de permissões de pseudo API. Para obter detalhes, consulte Pseudo API.

SignalWorkflowExecution

- Não é possível restringir os parâmetros dessa ação.

StartWorkflowExecution

- `tagList.member.0`: restrição de string. A chave é `swf:tagList.member.0`
- `tagList.member.1`: restrição de string. A chave é `swf:tagList.member.1`
- `tagList.member.2`: restrição de string. A chave é `swf:tagList.member.2`
- `tagList.member.3`: restrição de string. A chave é `swf:tagList.member.3`
- `tagList.member.4`: restrição de string. A chave é `swf:tagList.member.4`
- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.

- `workflowType.name`: restrição de string. A chave é `swf:workflowType.name`.
- `workflowType.version`: restrição de string. A chave é `swf:workflowType.version`.

> **Note**
>
> Não é possível restringir mais de cinco tags.

`TerminateWorkflowExecution`

- Não é possível restringir os parâmetros dessa ação.

Pseudo API

Esta seção lista os membros da pseudo API, que representam as decisões incluídas em `RespondDecisionTaskCompleted`. Se você tiver concedido permissão para usar `RespondDecisionTaskCompleted`, sua política poderá expressar permissões para os membros dessa API da mesma maneira que a API regular. Você pode restringir ainda mais alguns membros da pseudo API definindo condições em um ou mais parâmetros. Esta seção lista os membros da pseudo API e descreve brevemente os parâmetros que podem ser restritos e as chaves associadas.

> **Note**
>
> As chaves `aws:SourceIP`, `aws:UserAgent` e `aws:SecureTransport` não estão disponíveis para a pseudo API. Se a sua política de segurança pretendida exigir essas chaves para controlar o acesso à pseudo API, você poderá usá-las com a ação `RespondDecisionTaskCompleted`.

`CancelTimer`

- Não é possível restringir os parâmetros dessa ação.

`CancelWorkflowExecution`

- Não é possível restringir os parâmetros dessa ação.

`CompleteWorkflowExecution`

- Não é possível restringir os parâmetros dessa ação.

`ContinueAsNewWorkflowExecution`

- `tagList.member.0`: restrição de string. A chave é `swf:tagList.member.0`
- `tagList.member.1`: restrição de string. A chave é `swf:tagList.member.1`
- `tagList.member.2`: restrição de string. A chave é `swf:tagList.member.2`
- `tagList.member.3`: restrição de string. A chave é `swf:tagList.member.3`
- `tagList.member.4`: restrição de string. A chave é `swf:tagList.member.4`
- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.
- `workflowTypeVersion`: restrição de string. A chave é `swf:workflowTypeVersion`.

> **Note**
>
> Não é possível restringir mais de cinco tags.

`FailWorkflowExecution`

- Não é possível restringir os parâmetros dessa ação.

`RecordMarker`

- Não é possível restringir os parâmetros dessa ação.

`RequestCancelActivityTask`

- Não é possível restringir os parâmetros dessa ação.

`RequestCancelExternalWorkflowExecution`

- Não é possível restringir os parâmetros dessa ação.

`ScheduleActivityTask`

- `activityType.name`: restrição de string. A chave é `swf:activityType.name`.
- `activityType.version`: restrição de string. A chave é `swf:activityType.version`.
- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.

`SignalExternalWorkflowExecution`

- Não é possível restringir os parâmetros dessa ação.

`StartChildWorkflowExecution`

- `tagList.member.0`: restrição de string. A chave é `swf:tagList.member.0`
- `tagList.member.1`: restrição de string. A chave é `swf:tagList.member.1`
- `tagList.member.2`: restrição de string. A chave é `swf:tagList.member.2`
- `tagList.member.3`: restrição de string. A chave é `swf:tagList.member.3`
- `tagList.member.4`: restrição de string. A chave é `swf:tagList.member.4`
- `taskList.name`: restrição de string. A chave é `swf:taskList.name`.
- `workflowType.name`: restrição de string. A chave é `swf:workflowType.name`.
- `workflowType.version`: restrição de string. A chave é `swf:workflowType.version`.

> **Note**
>
> Não é possível restringir mais de cinco tags.

`StartTimer`

- Não é possível restringir os parâmetros dessa ação.

Políticas baseadas em tag

O Amazon SWF oferece suporte a políticas baseadas em tags. Por exemplo, é possível restringir domínios do Amazon SWF que incluem uma tag com a chave `environment` e o valor `production`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "swf:*",
            "Resource": "arn:aws:swf:*:123456789012:/domain/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

Esta política `Deny` o acesso a todos os domínios que foram marcados como `environment/production`.

Para obter mais informações sobre marcação, consulte:

- Tags no Amazon SWF
- Controlar o acesso com tags do IAM

Endpoints da Amazon VPC para o Amazon SWF

> **Note**
>
> AWS PrivateLink atualmente, o suporte está disponível somente nas regiões AWS Top Secret - Leste, AWS Secret Region e China.

Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus AWS recursos, você pode estabelecer uma conexão entre os fluxos de trabalho da Amazon VPC e do Amazon Simple Workflow Service. Você pode usar essa conexão com seus fluxos de trabalho do Amazon SWF sem cruzar a Internet pública.

A Amazon VPC permite que você lance AWS recursos em uma rede virtual personalizada. Você pode usar uma VPC para controlar as configurações de rede, como o intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Para obter mais informações sobre VPCs, consulte o Guia do usuário da Amazon VPC.

Para conectar seu Amazon VPC ao Amazon SWF, você deve primeiro definir um endpoint da VPC de interface, que permite conectar a VPC a outros Serviços da AWS. O endpoint fornece conectividade confiável e escalável sem a necessidade de um gateway da internet, da instância de conversão de endereço de rede (NAT) ou de uma conexão VPN. Para obter mais informações, consulte Endpoints da VPC da interface (AWS PrivateLink) no Guia do usuário da Amazon VPC.

Criação do endpoint

Você pode criar um endpoint Amazon SWF em sua VPC usando o AWS Management Console, the AWS Command Line Interface (AWS CLI), um AWS SDK, a API Amazon SWF ou. AWS CloudFormation

Para obter informações sobre como criar e configurar um endpoint usando o console da Amazon VPC ou a AWS CLI, consulte Creating an Interface Endpoint (“Criar um endpoint da interface”) no Guia do usuário da Amazon VPC.

> **Note**
>
> Ao criar um endpoint, especifique o Amazon SWF como o serviço ao qual você deseja que sua VPC se conecte. No console da Amazon VPC, os nomes dos serviços variam de acordo com a região da AWS . Por exemplo, na região AWS Top Secret - Leste, o nome do serviço do Amazon SWF é com.amazonaws. us-iso-east-1.swf.

Para obter informações sobre como criar e configurar um endpoint usando AWS CloudFormation, consulte o VPCEndpoint recurso AWS::EC2:: no Guia do AWS CloudFormation usuário.

Políticas de endpoint da Amazon VPC

Para controlar o acesso à conectividade ao Amazon SWF, você pode anexar uma política de endpoint AWS Identity and Access Management (IAM) ao criar um endpoint Amazon VPC. É possível criar regras complexas do IAM anexando várias políticas de endpoint. Para obter mais informações, consulte:

- Políticas de endpoint da Amazon Virtual Private Cloud para o Amazon SWF
- Controle do acesso a serviços com VPC endpoints

Políticas de endpoint da Amazon Virtual Private Cloud para o Amazon SWF

Você pode criar uma política de endpoint da Amazon VPC para o Amazon SWF, na qual você especifica o seguinte:

- A entidade principal que pode realizar ações.
- As ações que podem ser realizadas.
- Os recursos nos quais as ações podem ser executadas.

O exemplo a seguir mostra uma política de endpoint da Amazon VPC que permite todas as operações do Amazon SWF em um único domínio para um perfil do IAM específico.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "swf:*",
      "Resource": "arn:aws:swf:*:123456789012:/domain/myDomain",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/MyRole"
      }
    }
  ]
}
```

- Para obter mais informações sobre a criação de políticas de endpoint, consulte Controlling Access to Services with VPC Endpoints.
- Para obter informações sobre como você pode usar o IAM para controlar o acesso aos seus recursos AWS e aos do Amazon SWF, consulte. Gerenciamento de identidade e acesso no Amazon Simple Workflow Service

Solução de problemas de identidade e acesso do Amazon Simple Workflow Service

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com o Amazon SWF e o IAM.

Tópicos

- Não estou autorizado a executar uma ação no Amazon SWF
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos do Amazon SWF

Não estou autorizado a executar uma ação no Amazon SWF

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um recurso do *`my-example-widget`* fictício, mas não tem as permissões fictícias do `swf:`*`GetWidget`*.

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

Nesse caso, a política de Mateo deve ser atualizada para permitir que ele tenha acesso ao recurso *`my-example-widget`* usando a ação `swf:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a executar a ação `iam:PassRole`, suas políticas deverão ser atualizadas para permitir que você passe uma função para o Amazon SWF.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para executar uma ação no Amazon SWF. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos do Amazon SWF

É possível criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon SWF oferece suporte a esses recursos, consulte Como o Amazon Simple Workflow Service funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Registro e Monitoramento

Esta seção fornece informações sobre o registro em log e o monitoramento do Amazon SWF.

Tópicos

- Métricas do Amazon SWF para CloudWatch
- Visualizando as métricas do Amazon SWF para usar o CloudWatch AWS Management Console
- Gravando chamadas de API com AWS CloudTrail
- EventBridge para alterações no status de execução do Amazon SWF
- Usando Notificações de Usuários da AWS com o Amazon Simple Workflow Service

Métricas do Amazon SWF para CloudWatch

O Amazon SWF agora fornece métricas CloudWatch que você pode usar para rastrear seus fluxos de trabalho e atividades e definir alarmes nos valores limite que você escolher. Você pode visualizar as métricas usando AWS Management Console o. Para obter mais informações, consulte Visualizando as métricas do Amazon SWF para usar o CloudWatch AWS Management Console.

Tópicos

- Relatar unidades para métricas do Amazon SWF
- Métricas de eventos de API e decisão
- Métricas do Amazon SWF
- Nomes e dimensões de recursos não ASCII do Amazon SWF CloudWatch

Relatar unidades para métricas do Amazon SWF

Métricas que relatam um intervalo de tempo

Algumas das métricas do Amazon SWF CloudWatch são intervalos de tempo, sempre medidos em milissegundos. A CloudWatch unidade é relatada como`Time`. Essas métricas geralmente correspondem a estágios da execução de seu fluxo de trabalho para os quais você pode definir limites de tempo de atividade e de fluxo de trabalho e têm nomes semelhantes.

Por exemplo, a métrica `DecisionTaskStartToCloseTime` mede o tempo necessário para a conclusão da tarefa de decisão após o início da sua execução, que é o mesmo período de tempo para o qual você pode definir um valor de `DecisionTaskStartToCloseTimeout`.

Para um diagrama de cada um desses estágios de fluxo de trabalho e para saber quando eles ocorrem ao longo dos ciclos de vida de fluxo de trabalho e atividades, consulte Tipos de tempo limite do Amazon SWF .

Métricas que relatam uma contagem

Algumas das métricas do Amazon SWF para resultados de CloudWatch relatórios como contagem. Por exemplo, `WorkflowsCanceled` registra um resultado como um ou zero, indicando se o fluxo de trabalho foi ou não cancelado. Um valor de zero não indica que a métrica não foi relatada, apenas que a condição descrita pela métrica não ocorreu.

Algumas das métricas do Amazon SWF para CloudWatch esse relatório a `Count` in CloudWatch são uma contagem por segundo. Por exemplo,`ProvisionedRefillRate`, que é relatado como um `Count` in CloudWatch, representa uma taxa `Count` de solicitações por segundo.

Para métricas de contagem, os valores mínimo e máximo sempre serão zero ou um, mas a média será um valor entre zero e um.

Métricas de eventos de API e decisão

Você pode monitorar os eventos de API e de decisão CloudWatch para fornecer informações sobre seu uso e capacidade. Consulte os agentes de decisão na seção Conceitos básicos de fluxo de trabalho no Amazon SWF e o tópico Decision em Amazon Simple Workflow Service API Reference.

Você também pode monitorar esses limites para emitir um alarme quando estiver se aproximando dos limites de controle de utilização do Amazon SWF. Consulte Cotas de controle de utilização do Amazon SWF para obter uma descrição desses limites e suas configurações padrão. Esses limites foram projetados para evitar que fluxos de trabalho incorretos consumam recursos excessivos do sistema. Para solicitar um aumento nos seus limites, consulte: ???.

Como prática recomendada, você deve configurar CloudWatch alarmes em cerca de 60% da sua API ou capacidade de eventos de decisão. Isso permitirá que você ajuste seu fluxo de trabalho ou solicite um aumento do limite de serviço antes que o controle de utilização do Amazon SWF seja habilitado. Dependendo da capacidade de explosão das suas chamadas, você pode configurar diferentes alarmes para notificar quando estiver se aproximando dos seus limites de serviço:

- Se o seu tráfego tem picos significativos, defina um alarme em 60% dos seus limites de `ProvisionedBucketSize`.
- Se as suas chamadas tiverem uma taxa relativamente estável, defina um alarme em 60% de seu limite de `ProvisionedRefillRate` para seus eventos de API e decisão relacionados.

Métricas do Amazon SWF

As métricas a seguir estão disponíveis para o Amazon SWF:

Métrica	Descrição
`DecisionTaskScheduleToStartTime`	O intervalo de tempo, em milissegundos, entre o momento em que a tarefa de decisão foi agendada e o momento em que ela foi selecionada por um operador e iniciada. CloudWatch Unidades: `Time` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Average, Minimum, Maximum`
`DecisionTaskStartToCloseTime`	O intervalo de tempo, em milissegundos, entre o momento em que a tarefa de decisão foi agendada e o momento em que ela foi encerrada. CloudWatch Unidades: `Time` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Average, Minimum, Maximum`
`DecisionTasksCompleted`	A contagem de tarefas de decisão que foram concluídas. CloudWatch Unidades: `Count` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`
`PendingTasks`	A contagem de tarefas pendentes em um intervalo de 1 minuto para uma Lista de tarefas determinada. CloudWatch Unidades: `Count` Dimensões: `Domain, TaskListName` Estatísticas válidas: `Sum`

Métrica	Descrição
`StartedDecisionTasksTimedOutOnClose`	A contagem de tarefas de decisão que foram iniciadas, mas atingiram o tempo limite no fechamento. CloudWatch Unidades: `Count` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`
`WorkflowStartToCloseTime`	O intervalo de tempo, em milissegundos, entre o momento em que o fluxo de trabalho foi iniciado e o momento em que ele foi encerrado. CloudWatch Unidades: `Time` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Average, Minimum, Maximum`
`WorkflowsCanceled`	A contagem de fluxos de trabalho que foram cancelados. CloudWatch Unidades: `Count` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`
`WorkflowsCompleted`	A contagem de fluxos de trabalho que foram concluídos. CloudWatch Unidades: `Count` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`

Métrica	Descrição
`WorkflowsContinuedAsNew`	A contagem de fluxos de trabalho que continuaram como novos. CloudWatch Unidades: `Count` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`
`WorkflowsFailed`	A contagem dos fluxos de trabalho que falharam. CloudWatch Unidades: `Count` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`
`WorkflowsTerminated`	A contagem dos fluxos de trabalho que foram finalizados. CloudWatch Unidades: `Count` Dimensões: `Cause, Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`
`WorkflowsTimedOut`	A contagem de fluxos de trabalho que atingiram o tempo limite, por qualquer motivo. CloudWatch Unidades: `Count` Dimensões: `Domain, WorkflowTypeName, WorkflowTypeVersion` Estatísticas válidas: `Sum`

Métrica	Descrição
`ActivityTaskScheduleToCloseTime`	O intervalo de tempo, em milissegundos, entre o momento em que a atividade foi agendada e o momento em que ela foi encerrada. CloudWatch Unidades: `Time` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Average, Minimum, Maximum`
`ActivityTaskScheduleToStartTime`	O intervalo de tempo, em milissegundos, entre o momento em que a tarefa de atividade foi programada até quando ela foi iniciada. CloudWatch Unidades: `Time` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Average, Minimum, Maximum`
`ActivityTaskStartToCloseTime`	O intervalo de tempo, em milissegundos, entre o momento em que a tarefa de atividade foi iniciada e o momento em que ela foi encerrada. CloudWatch Unidades: `Time` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Average, Minimum, Maximum`

Métrica	Descrição
`ActivityTasksCanceled`	A contagem de tarefas de atividades que foram canceladas. CloudWatch Unidades: `Count` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Sum`
`ActivityTasksCompleted`	A contagem de tarefas de atividade que foram concluídas. CloudWatch Unidades: `Count` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Sum`
`ActivityTasksFailed`	A contagem de tarefas de atividade que apresentaram falha. CloudWatch Unidades: `Count` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Sum`
`ScheduledActivityTasksTimedOutOnClose`	A contagem de tarefas de atividade que foram programadas, mas atingiram o tempo limite no fechamento. CloudWatch Unidades: `Count` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Sum`

Métrica	Descrição
`ScheduledActivityTasksTimedOutOnStart`	A contagem de tarefas de atividade que foram programadas, mas atingiram o tempo limite no início. CloudWatch Unidades: `Count` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Sum`
`StartedActivityTasksTimedOutOnClose`	A contagem de tarefas de atividade que foram iniciadas, mas atingiram o tempo limite no fechamento. CloudWatch Unidades: `Count` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Sum`
`StartedActivityTasksTimedOutOnHeartbeat`	A contagem de tarefas de atividade que foram iniciadas, mas atingiram o tempo limite de heartbeat. CloudWatch Unidades: `Count` Dimensões: `Domain, ActivityTypeName, ActivityTypeVersion` Estatísticas válidas: `Sum`
`ThrottledEvents`	A contagem de solicitações que foram limitadas. CloudWatch Unidades: `Count` Dimensões: `APIName, DecisionName, ThrottlingScope` Estatísticas válidas: `Sum`

Métrica	Descrição
`ProvisionedBucketSize`	A contagem de solicitações disponíveis por segundo. Dimensões: `APIName, DecisionName` Estatísticas válidas: `Minimum`
`ConsumedCapacity`	A contagem de solicitações por segundo. CloudWatch Unidades: `Count` Dimensões: `APIName, DecisionName` Estatísticas válidas: `Sum`
`ConsumedLimit`	A quantidade de limite geral que foi consumida. Dimensões: `GeneralLimitType`
`ProvisionedRefillRate`	A contagem de solicitações por segundo que são permitidas no bucket. Dimensões: `APIName, DecisionName` Estatísticas válidas: `Minimum`
`ProvisionedLimit`	O valor do limite geral que é provisionado para a conta. Dimensões: `GeneralLimitType`

Dimensão	Descrição
`Domain`	Filtra dados para o domínio do Amazon SWF no qual o fluxo de trabalho ou a atividade está sendo executada.
`ActivityTypeName`	Filtra dados para o nome do tipo de atividade.
`ActivityTypeVersion`	Filtra dados para a versão do tipo de atividade.

Dimensão	Descrição
`WorkflowTypeName`	Filtra dados para o nome do tipo de fluxo de trabalho dessa execução de fluxo de trabalho.
`WorkflowTypeVersion`	Filtra dados para a versão do tipo de fluxo de trabalho dessa execução de fluxo de trabalho.
`APIName`	Filtra dados para uma API do nome de API especificado.
`DecisionName`	Filtra dados para o nome da Decisão especificado.
`TaskListName`	Filtra dados para o nome da Lista de tarefas especificado.
`TaskListClassification`	Filtra os dados de acordo com a classificação da lista de tarefas. O valor é “D” para listas de tarefas de decisão e “A” para listas de tarefas de atividade.
`ThrottlingScope`	Filtra os dados para o escopo de limitação especificado. O valor é “Conta” ao exceder a cota no nível da conta ou “Fluxo de trabalho” ao exceder a cota no nível do fluxo de trabalho.

Nomes e dimensões de recursos não ASCII do Amazon SWF CloudWatch

O Amazon SWF permite caracteres não ASCII em nomes de recursos, como e. TaskList DomainName No entanto, os valores de dimensão das CloudWatch métricas só podem conter caracteres ASCII imprimíveis. Para garantir que o Amazon SWF use valores de dimensão compatíveis com os CloudWatch requisitos, os nomes de recursos do Amazon SWF que não atendem a esses requisitos são convertidos e terão uma soma de verificação anexada da seguinte forma:

- Qualquer caractere não ASCII é substituído por ?.
- A string de entrada ou a string convertida será, se necessário, truncada. Isso garante que, quando a soma de verificação for anexada, o novo comprimento da string não exceda o CloudWatch máximo.
- Como todos os caracteres não ASCII são convertidos em?, alguns valores de dimensão CloudWatch métrica que eram diferentes antes da conversão podem parecer iguais após a

conversão. Para ajudar a diferenciá-los, um sublinhado (_) seguido pelos primeiros 16 caracteres da SHA256 soma de verificação do nome do recurso original é anexado ao nome do recurso.

Exemplos de conversão:

- `test àpple` seria convertido em `test ?pple_82cc5b8e3a771d12`
- `àòà` seria convertido em `???_2fec5edbb2c05c22`.
- Os TaskList nomes `àpplé` e ambos `âpplè` seriam convertidos `?ppl?` em e seriam idênticos. A adição da soma de verificação retorna valores distintos, `?ppl?_f39a36df9d85a69d` e `?ppl?_da3efb4f11dd0f7f`.

> **Tip**
>
> Você pode gerar sua própria SHA256 soma de verificação. Por exemplo, para usar a ferramenta de linha de comando `shasum`:
> echo -n "<the original resource name>" | shasum -a 256 | cut -c1-16

Visualizando as métricas do Amazon SWF para usar o CloudWatch AWS Management Console

CloudWatch A Amazon fornece várias métricas visíveis para fluxos de trabalho e atividades do Amazon SWF. Você pode visualizar as métricas e definir alarmes para as execuções do fluxo de trabalho do Amazon SWF usando o AWS Management Console. Você deve estar conectado ao console para continuar.

Para obter uma descrição de cada uma das métricas disponíveis, consulte Métricas do Amazon SWF para CloudWatch.

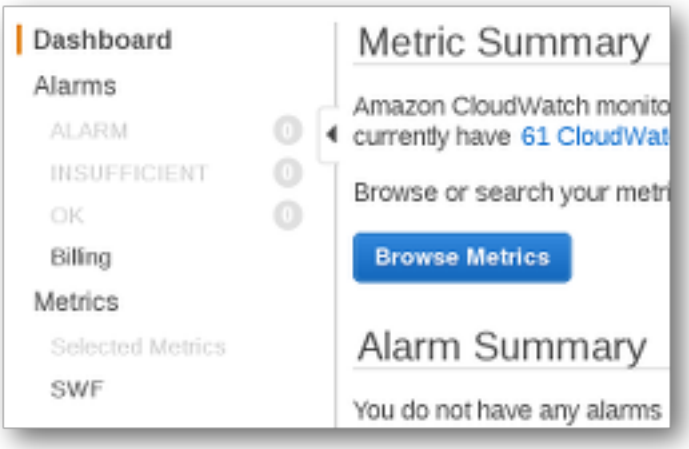
Tópicos

- Visualizar métricas
- Configurar alarmes

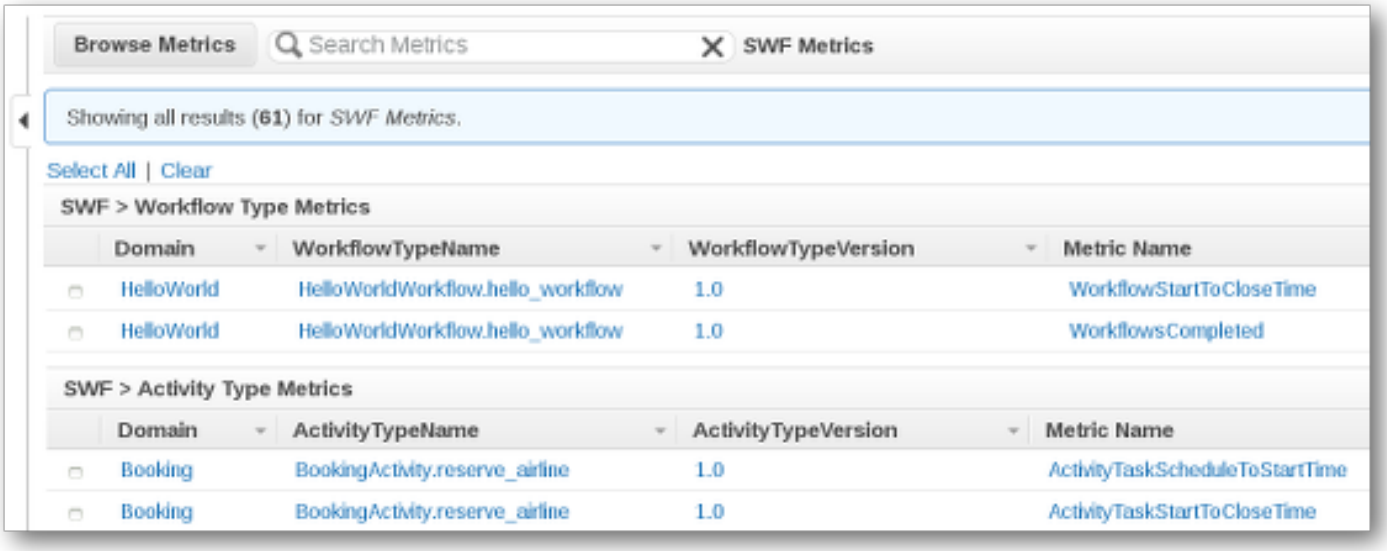
Visualizar métricas

Para visualizar suas métricas para o Amazon SWF

1. Faça login no AWS Management Console e abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, em Métricas, escolha SWF.



Se você tiver processado qualquer execução de fluxo de trabalho recentemente, verá duas listas de métricas apresentadas: Métricas de tipo de fluxo de trabalho e Métricas de tipo de atividade.



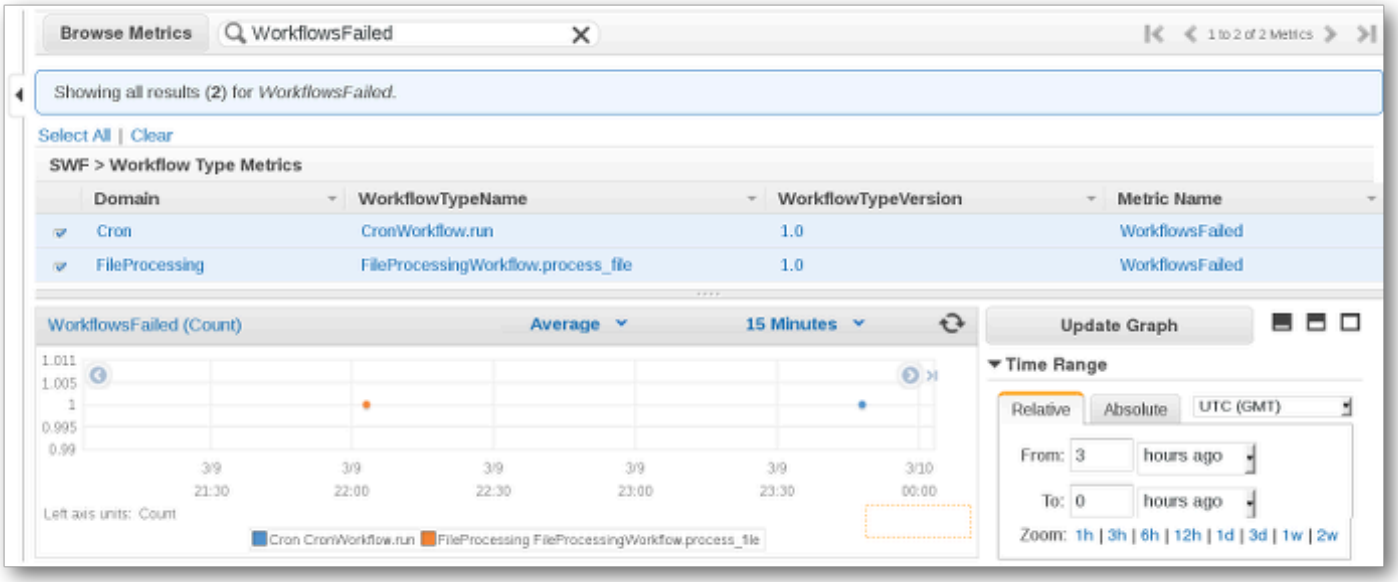
> **Note**
>
> Inicialmente, talvez você só possa ver as Métricas de tipo de fluxo de trabalho. As Métricas de tipo de atividade são apresentadas na mesma exibição, mas talvez seja necessário rolar para baixo para vê-las.

Até 50 das métricas mais recentes serão exibidas de cada vez, divididas entre métricas de fluxo de trabalho e métricas de atividade.

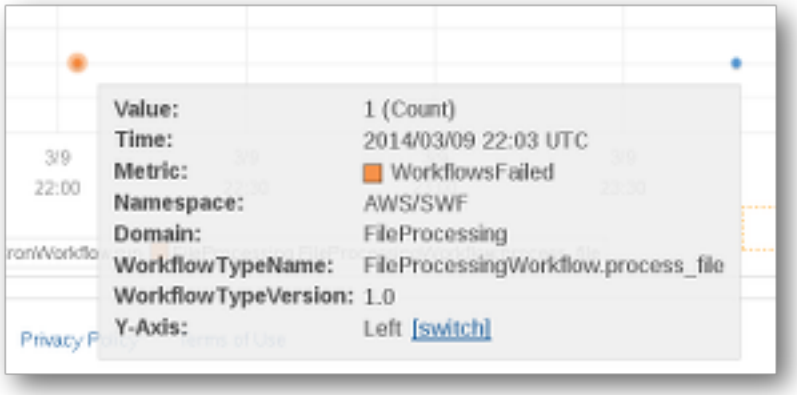
Você pode usar os cabeçalhos interativos acima de cada coluna na lista para classificar suas métricas usando qualquer uma das dimensões fornecidas. Para fluxos de trabalho, as dimensões são Domain, WorkflowTypeNameWorkflowTypeVersion, e Metric Name. Para atividades, as dimensões são Domínio, ActivityTypeNameActivityTypeVersion, e Nome da métrica.

Os vários tipos de métricas estão descritos em Métricas do Amazon SWF para CloudWatch.

Você pode visualizar gráficos de métricas escolhendo as caixas ao lado da linha de métrica na lista e alterando os parâmetros de gráficos usando os controles de Intervalo de tempo à direita da exibição de gráfico.



Para obter detalhes sobre qualquer ponto no gráfico, coloque o cursor sobre o ponto no gráfico. Um detalhe das dimensões do ponto será mostrado.



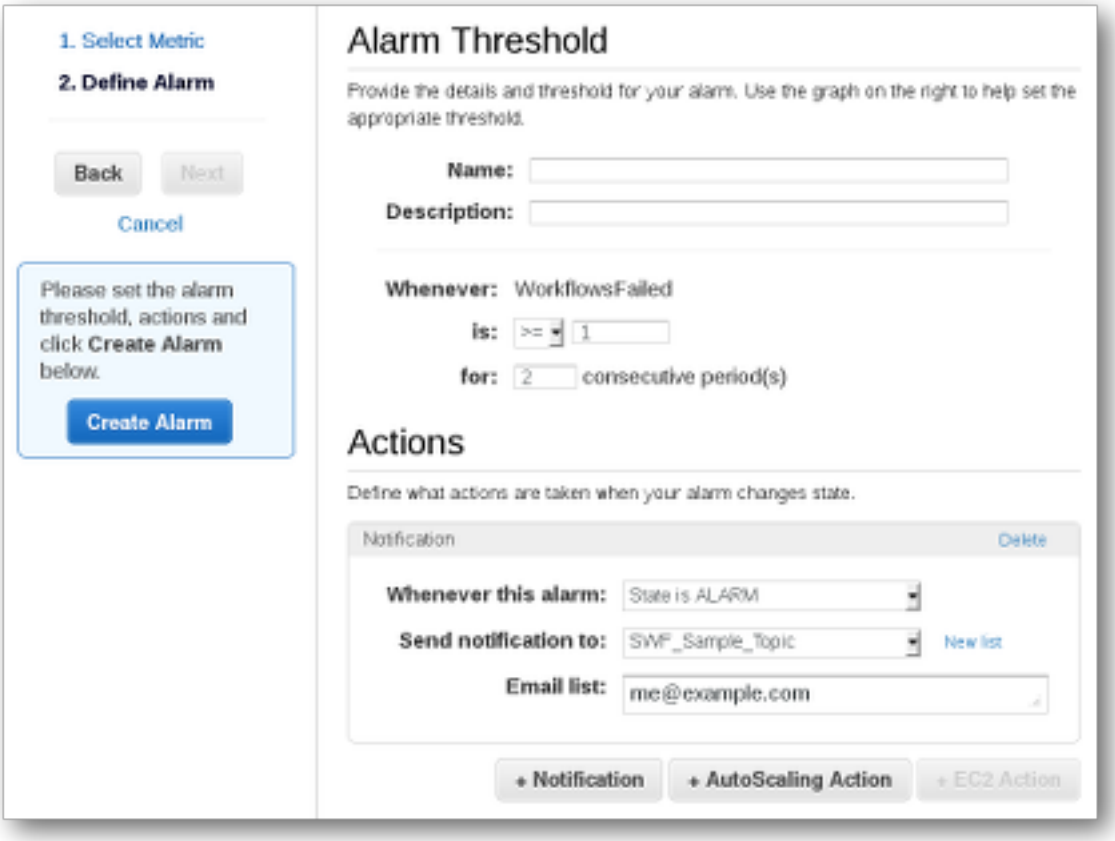
Para obter mais informações sobre como trabalhar com CloudWatch métricas, consulte Visualização, representação gráfica e publicação de métricas no Guia do CloudWatch usuário da Amazon.

Configurar alarmes

Você pode usar CloudWatch alarmes para realizar ações como notificá-lo quando um limite de alarme é atingido. Por exemplo, você pode configurar um alarme para enviar uma notificação para um tópico do SNS ou para enviar um e-mail quando a métrica `WorkflowsFailed` subir além de um determinado limite.

Para definir um alarme em qualquer uma das suas métricas

1. Escolha uma única métrica marcando sua caixa.
2. À direita do gráfico, nos controles de Tools, escolha Create Alarm.
3. Na tela Define Alarm, insira o valor do limite do alarme, os parâmetros do período e as ações a serem realizadas.



Para obter mais informações sobre a configuração e o uso de CloudWatch alarmes, consulte Criação de CloudWatch alarmes da Amazon no Guia CloudWatch do usuário da Amazon.

Gravando chamadas de API com AWS CloudTrail

O Amazon Simple Workflow Service é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou um AWS service (Serviço da AWS). CloudTrail captura todas as chamadas de API para o Amazon SWF como eventos. As chamadas capturadas incluem chamadas do console do Amazon SWF e chamadas de código para as operações da API do Amazon SWF. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon SWF, o endereço IP a partir do qual a solicitação foi feita, quando foi feita e detalhes adicionais.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário raiz ou credenciais de usuário.

- Se a solicitação foi feita em nome de um usuário do Centro de Identidade do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS service (Serviço da AWS).

CloudTrail está ativo Conta da AWS quando você cria a conta e você tem acesso automático ao histórico de CloudTrail eventos. O histórico de CloudTrail eventos fornece um registro visível, pesquisável, baixável e imutável dos últimos 90 dias de eventos de gerenciamento registrados em um. Região da AWS Para obter mais informações, consulte Trabalhando com o histórico de CloudTrail eventos no Guia AWS CloudTrail do usuário. Não há CloudTrail cobrança pela visualização do histórico de eventos.

Para um registro contínuo dos eventos dos Conta da AWS últimos 90 dias, crie uma trilha ou um armazenamento de dados de eventos do CloudTrailLake.

CloudTrail trilhas

Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Todas as trilhas criadas usando o AWS Management Console são multirregionais. Só é possível criar uma trilha de região única ou de várias regiões usando a AWS CLI. É recomendável criar uma trilha multirregional porque você captura todas as atividades Regiões da AWS em sua conta. Ao criar uma trilha de região única, é possível visualizar somente os eventos registrados na Região da AWS da trilha. Para obter mais informações sobre trilhas, consulte Criar uma trilha para a Conta da AWS e Criar uma trilha para uma organização no Guia do usuário do AWS CloudTrail .

Você pode entregar uma cópia dos seus eventos de gerenciamento contínuos para o bucket do Amazon S3 sem nenhum custo CloudTrail criando uma trilha. No entanto, há cobranças de armazenamento do Amazon S3. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços. Para receber informações sobre a definição de preços do Amazon S3, consulte Definição de preços do Amazon S3.

CloudTrail Armazenamentos de dados de eventos em Lake

CloudTrail O Lake permite que você execute consultas baseadas em SQL em seus eventos. CloudTrail O Lake converte eventos existentes no formato JSON baseado em linhas para o formato Apache ORC. O ORC é um formato colunar de armazenamento otimizado para recuperação rápida de dados. Os eventos são agregados em armazenamentos de dados de eventos, que são coleções imutáveis de eventos baseados nos critérios selecionados com a aplicação de seletores de eventos avançados. Os seletores que aplicados a um armazenamento

de dados de eventos controlam quais eventos persistem e estão disponíveis para consulta. Para obter mais informações sobre o CloudTrail Lake, consulte Trabalhando com o AWS CloudTrail Lake no Guia AWS CloudTrail do Usuário.

CloudTrail Os armazenamentos e consultas de dados de eventos em Lake incorrem em custos. Ao criar um armazenamento de dados de eventos, você escolhe a opção de preço que deseja usar para ele. A opção de preço determina o custo para a ingestão e para o armazenamento de eventos, e o período de retenção padrão e máximo para o armazenamento de dados de eventos. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços.

Eventos de dados em CloudTrail

Os Eventos de dados fornecem informações sobre as operações de recursos realizadas em um recurso (por exemplo, leitura ou gravação em um objeto do Amazon S3). Também são conhecidas como operações de plano de dados. Eventos de dados geralmente são atividades de alto volume. Por padrão, CloudTrail não registra eventos de dados. O histórico de CloudTrail eventos não registra eventos de dados.

Há cobranças adicionais para eventos de dados. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços.

Você pode registrar eventos de dados para os tipos de recursos do Amazon SWF usando o CloudTrail console ou as operações AWS CLI de CloudTrail API. Para obter mais informações sobre como registrar eventos de dados em log, consulte Registrar eventos de dados com o AWS Management Console e Registrar eventos de dados com a AWS Command Line Interface no Guia do usuário do AWS CloudTrail .

A tabela a seguir lista os tipos de recursos do Amazon SWF para os quais você pode registrar eventos de dados. A coluna Tipo de evento de dados mostra o valor a ser escolhido na lista Tipo de evento de dados no CloudTrail console. A coluna de valor resources.type mostra o `resources.type` valor, que você especificaria ao configurar seletores de eventos avançados usando o ou. AWS CLI CloudTrail APIs A CloudTrail coluna Dados APIs registrados em mostra as chamadas de API registradas CloudTrail para o tipo de recurso.

É possível configurar seletores de eventos avançados para filtrar os campos `eventName`, `readOnly` e `resources.ARN` para registrar em log somente os eventos que são importantes para você. Para obter mais informações sobre esses campos, consulte AdvancedFieldSelector, na Referência de APIs do AWS CloudTrail .

Tipo de evento de dados	valor resources.type	Dados APIs registrados em CloudTrail
Domínio SWF	`AWS::SWF::Domain`	Eventos de fluxo de trabalho • CountClosedWorkflowExecutions • CountOpenWorkflowExecutions • DescribeWorkflowExecution • ListClosedWorkflowExecutions • ListOpenWorkflowExecutions • GetWorkflowExecutionHistory • RequestCancelWorkflowExecution • SignalWorkflowExecution • StartWorkflowExecution • TerminateWorkflowExecution Eventos de tarefas • CountPendingActivityTasks • PollForDecisionTask • PollForActivityTask • RecordActivityTaskHeartbeat • RespondActivityTaskCanceled • RespondActivityTaskCompleted

Tipo de evento de dados	valor resources.type	Dados APIs registrados em CloudTrail
		• RespondActivityTaskFailed • RespondDecisionTaskCompleted Eventos de decisão • CancelTimer • CancelWorkflowExecution • CompleteWorkflowExecution • ContinueAsNewWorkflowExecution • FailWorkflowExecution • RecordMarker • RequestCancelActivityTask • RequestCancelExternalWorkflowExecution • ScheduleActivityTask • ScheduleLambdaFunction • SignalExternalWorkflowExecution • StartChildWorkflowExecution • StartTimer

> **CloudTrail eventos e RespondDecisionTaskCompleted**
>
> A RespondDecisionTaskCompletedaction obtém uma lista de decisões na carga útil da solicitação. Uma chamada concluída emitirá eventos de CloudTrail dados N+1, um para cada

decisão mais um para a própria chamada de API. Os eventos de dados e o evento da API terão todos o mesmo ID de solicitação.

Eventos de gestão em CloudTrail

Os eventos de gerenciamento fornecem informações sobre as operações de gerenciamento que são realizadas nos recursos do seu Conta da AWS. Também são conhecidas como operações de ambiente de gerenciamento. Por padrão, CloudTrail registra eventos de gerenciamento.

O Amazon Simple Workflow Service registra as seguintes operações do plano de controle CloudTrail como eventos de gerenciamento.

Eventos de domínio

- RegisterDomain
- DescribeDomain
- ListDomains
- DeprecateDomain
- UndeprecateDomain

Eventos de atividades

- RegisterActivityType
- DescribeActivityType
- ListActivityTypes
- DeprecateActivityType
- UndeprecateActivityType
- DeleteActivityType

WorkflowType Eventos

- RegisterWorkflowType
- DescribeWorkflowType
- ListWorkflowTypes

- DeprecateWorkflowType
- UndeprecateWorkflowType
- DeleteWorkflowType

Tag: Eventos

- TagResource
- UntagResource
- ListTagsforResource

Exemplo de evento do

Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a operação de API solicitada, a data e a hora da operação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, os eventos não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra um CloudTrail evento que demonstra a `CountClosedWorkflowExecutions` operação.

```
{
    "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-EXAMPLEbbbbb",
    "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 obter informações sobre o conteúdo do CloudTrail registro, consulte o conteúdo do CloudTrail registro no Guia AWS CloudTrail do usuário.

EventBridge para alterações no status de execução do Amazon SWF

Você usa EventBridge a Amazon para responder a mudanças de estado ou eventos em um AWS recurso. Quando o Amazon SWF emite um evento, ele sempre vai para o barramento

de EventBridge eventos padrão da sua conta. Você pode criar uma regra para eventos, associá-la ao barramento de eventos padrão e especificar uma ação de destino a ser tomada ao EventBridge receber um evento que corresponda à regra. Dessa forma, você pode monitorar seus fluxos de trabalho sem precisar fazer pesquisas constantes usando a API `GetWorkflowExecutionHistory`. Com base nas mudanças nas execuções do fluxo de trabalho, você pode usar um EventBridge destino para chamar AWS Lambda funções, publicar mensagens em tópicos do Amazon Simple Notification Service (Amazon SNS) e muito mais.

Você pode ver o conteúdo completo de um evento de alteração do status de execução usando `DescribeWorkflowExecution`.

Para obter mais informações, consulte o Guia EventBridge do usuário da Amazon.

EventBridge eventos

Os tipos de eventos do histórico contêm as alterações do estado de execução. A `detail` seção de cada evento contém pelo menos os seguintes parâmetros:

- `eventId`: o ID do evento mostrado por GetWorkflowExecutionHistory.
- `workflowExecutionDetail`: o estado do fluxo de trabalho quando o evento foi emitido.
- `eventType`: o tipo de evento histórico, um dos seguintes:
  - `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`

Exemplos de eventos do Amazon SWF

Veja a seguir exemplos do Amazon SWF enviando eventos para: EventBridge

Tópicos

- Execução iniciada
- Execução concluída
- Falha na execução
- A execução atingiu o tempo limite
- Execução encerrada

Em cada caso, a seção `detail` em dados de evento fornece as mesmas informações da API do `DescribeWorkflowExecution`. O campo `executionStatus` indica o status da execução no momento em que o evento foi enviado, seja `OPEN` ou `CLOSED`.

Execução iniciada


```
{
  "version": "0",
  "id": "444444444444",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "444444444444",
  "time": "2020-05-08T15:57:38Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:444444444444:/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": "444444444444"
        },
        "taskPriority": null,
        "childPolicy": "ABANDON",
        "lambdaRole": "arn:aws:iam::444444444444:role/BasicSWFLambdaExecution"
```

```
      },
      "openCounts": {
        "openActivityTasks": 0,
        "openDecisionTasks": 1,
        "openTimers": 0,
        "openChildWorkflowExecutions": 0,
        "openLambdaFunctions": 0
      },
      "latestActivityTaskTimestamp": null,
    }
  }
}
```

Execução concluída

```
{
  "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,
    }
  }
}
```

Falha na execução

```
{
  "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,
    }
  }
}
```

A execução atingiu o tempo limite


```
{
  "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,
    }
  }
}
```

Execução encerrada

```
{
  "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,
    }
  }
}
```

Usando Notificações de Usuários da AWS com o Amazon Simple Workflow Service

Você pode usar o Notificações de Usuários da AWS para configurar canais de entrega para receber notificações sobre eventos do Amazon Simple Workflow Service. Você recebe uma notificação quando um evento corresponde a uma regra especificada. Você pode receber notificações de eventos por meio de vários canais, incluindo e-mail, Amazon Q Developer em aplicativos de bate-papo, notificações por chat ou notificações AWS Console Mobile Applicationpush. Você também pode visualizar notificações na Central de Notificações do Console. O Notificações de Usuários oferece é compatível com agregação, o que pode reduzir o número de notificações que você recebe durante eventos específicos.

Validação de conformidade para o Amazon Simple Workflow Service

Os auditores terceirizados avaliam a segurança e a conformidade do Amazon Simple Workflow Service como parte de vários programas de conformidade da AWS . Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista de AWS serviços no escopo de programas de conformidade específicos, consulte AWS Serviços no escopo do programa de conformidade AWS . Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixando relatórios no AWS Artifact .

Sua responsabilidade de conformidade ao usar o Amazon SWF é determinada pela sensibilidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. A AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido sobre sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos com foco em segurança e conformidade em. AWS
- Documento técnico sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper descreve como as empresas podem usar para criar aplicativos compatíveis com a HIPAA. AWS
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência no Amazon Simple Workflow Service

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas,

conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Além da infraestrutura AWS global, o Amazon SWF oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

Segurança de infraestrutura no Amazon Simple Workflow Service

Como serviço gerenciado, é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Você pode chamar essas operações de API de qualquer local da rede, mas o Amazon SWF oferece suporte a políticas de acesso baseadas em recursos, que podem incluir restrições com base no endereço IP de origem. Você também pode usar as políticas do Amazon SWF para controlar o acesso de endpoints específicos da Amazon Virtual Private Cloud (Amazon VPC) ou específicos.

VPCs Efetivamente, isso isola o acesso à rede a um determinado recurso Amazon SWF somente da VPC específica dentro da rede. AWS

Configuração e análise de vulnerabilidade no Amazon Simple Workflow Service

A configuração e os controles de TI são uma responsabilidade compartilhada entre você AWS e você, nosso cliente. Para obter mais informações, consulte o modelo de responsabilidade AWS compartilhada.

Usando o AWS CLI com o Amazon Simple Workflow Service

Muitos dos recursos do Amazon Simple Workflow Service podem ser acessados a partir da AWS CLI. O AWS CLI fornece uma alternativa ao uso do Amazon SWF com o AWS Management Console ou, em alguns casos, à programação com a API do Amazon SWF e o. AWS Flow Framework

Por exemplo, você pode usar o AWS CLI para registrar um novo tipo de fluxo de trabalho:

```
aws swf register-workflow-type --domain MyDomain --name "MySimpleWorkflow" --workflow-
version "v1"
```

Você também pode listar seus tipos de fluxo de trabalho registrados:

```
aws swf list-workflow-types --domain MyDomain --registration-status REGISTERED
```

Veja a seguir um exemplo da saída padrão em 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"
            }
        }
    ]
}
```

Os comandos do Amazon SWF AWS CLI fornecem a capacidade de iniciar e gerenciar execuções de fluxo de trabalho, pesquisar tarefas de atividade, registrar batimentos cardíacos de tarefas e

muito mais! Para obter uma lista completa dos comandos do Amazon SWF, com descrições dos argumentos disponíveis e exemplos que mostram seu uso, consulte Comandos do Amazon SWF na Referência de comandos da AWS CLI .

Os AWS CLI comandos seguem de perto a API do Amazon SWF, então você pode usá-los AWS CLI para aprender sobre a API subjacente do Amazon SWF. Você também pode usar seu conhecimento de API existente para criar um protótipo de código ou executar ações do Amazon SWF na linha de comando.

Para saber mais sobre o AWS CLI, consulte o Guia AWS Command Line Interface do usuário.

Trabalhando com o Amazon SWF APIs

Além de usar AWS SDKs os descritos emDesenvolva com AWS SDKs, você pode usar a API HTTP diretamente.

Para usar a API, você pode enviar solicitações HTTP ao endpoint do SWF que corresponde à região que você deseja usar para seus domínios, fluxos de trabalho e atividades. Para obter mais informações sobre como fazer solicitações HTTP para o Amazon SWF, consulte Solicitações HTTP no Amazon SWF.

Esta seção fornece informações básicas sobre o uso da API HTTP para desenvolver seus fluxos de trabalho com o Amazon SWF. Recursos mais avançados, como usar cronômetros, registrar CloudTrail e marcar seus fluxos de trabalho, são fornecidos na seção,. Conceitos básicos de fluxo de trabalho no Amazon SWF

Tópicos

- Solicitações HTTP no Amazon SWF
- Lista de ações do Amazon SWF por categoria
- Registrando um domínio com o Amazon SWF
- Definindo valores de tempo limite no Amazon SWF
- Registro de um tipo de fluxo de trabalho com o Amazon SWF
- Registro de um tipo de atividade com o Amazon SWF
- AWS Lambda tarefas no Amazon SWF
- Desenvolver um operador de atividade no Amazon SWF
- Desenvolvendo decisores no Amazon SWF
- Iniciando fluxos de trabalho no Amazon SWF
- Definindo a prioridade da tarefa no Amazon SWF
- Tratamento de erros no Amazon SWF

Solicitações HTTP no Amazon SWF

Se você não usar um deles AWS SDKs, poderá realizar operações do Amazon Simple Workflow Service (Amazon SWF) via HTTP usando o método de solicitação POST. O método POST exige a

especificação da operação no cabeçalho da solicitação e o fornecimento de dados para operação no formato JSON no corpo da solicitação.

Conteúdo de cabeçalho HTTP

O Amazon SWF requer as seguintes informações no cabeçalho de uma solicitação HTTP:

- `host` O endpoint do Amazon SWF.
- `x-amz-date`Você deve fornecer o carimbo de data/hora no `Date` cabeçalho HTTP ou no AWS `x-amz-date header` (algumas bibliotecas de cliente HTTP não permitem que você defina o `Date` cabeçalho). Quando existe um cabeçalho `x-amz-date`, o sistema ignora qualquer cabeçalho `Date` ao autenticar a solicitação.

  A data precisa ser especificada em um destes três formatos, conforme especificado em HTTP/1.1 RFC:

  - Domingo, 06-Nov-1994 08:49:37 GMT (RFC 822, atualizada pela RFC 1123)
  - Domingo, 06-Nov-94 08:49:37 GMT (RFC 850, substituído por RFC 1036)
  - Dom Nov 6 08:49:37 1994 (formato ANSI C asctime())
- `x-amzn-authorization` Os parâmetros de solicitação assinados no formato:

  ```
  AWS3 AWSAccessKeyId=####,Algorithm=HmacSHA256, [,SignedHeaders=Header1;Header2;...]
  Signature=S(StringToSign)
  ```

  `AWS3`— Essa é uma tag AWS específica da implementação que indica a versão de autenticação usada para assinar a solicitação (atualmente, para o Amazon SWF, esse valor é sempre). `AWS3`

  `AWSAccessKeyId`— Sua ID da chave de AWS acesso.

  `Algorithm`— O algoritmo usado para criar o valor HMAC-SHA do string-to-sign, como ou. `HmacSHA256` `HmacSHA1`

  `Signature`— Base64 (Algoritmo (StringToSign, SigningKey)). Para obter mais detalhes, consulte Cálculo da assinatura HMAC-SHA para o Amazon SWF

  `SignedHeaders`— (Opcional) Se presente, deve conter uma lista de todos os cabeçalhos HTTP usados no cálculo canônico HttpHeaders . Um caractere de ponto e vírgula (;) (caractere ASCII 59) deve ser usado como delimitador para valores de lista.
- `x-amz-target`: o serviço de destino da solicitação e a operação para os dados, no formato

`com.amazonaws.swf.service.model.SimpleWorkflowService. + <action>`

Por exemplo, `com.amazonaws.swf.service.model.SimpleWorkflowService.RegisterDomain`

- `content-type`: o tipo precisa especificar JSON e o conjunto de caracteres, como `application/json; charset=UTF-8`

O seguinte é um cabeçalho de exemplo para uma solicitação HTTP para criar um domínio.

```
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=tzjkF55lxAxPhzp/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"}
```

Aqui está um exemplo da resposta HTTP correspondente.

```
HTTP/1.1 200 OK
Content-Length: 0
```

```
Content-Type: application/json
x-amzn-RequestId: 4ec4ac3f-3e16-11e1-9b11-7182192d0b57
```

Conteúdo do corpo HTTP

O corpo de uma solicitação HTTP apresenta os dados da operação especificada no cabeçalho da solicitação HTTP. Use o formato de dados JSON para transmitir valores e estrutura de dados, simultaneamente. Os elementos podem ser aninhados dentro de outros elementos usando a notação de colchetes. Por exemplo, o seguinte mostra uma solicitação para listar todas as execuções de fluxo de trabalho iniciadas entre dois pontos especificados no tempo, usando a notação Unix Time.

```
{
 "domain": "867530901",
 "startTimeFilter":
 {
   "oldestDate": 1325376070,
  "latestDate": 1356998399
 },
 "tagFilter":
 {
   "tag": "music purchase"
 }
}
```

Exemplo de solicitação e resposta JSON do Amazon SWF

O exemplo a seguir mostra uma solicitação ao Amazon SWF para obter uma descrição do domínio que criamos anteriormente. Em seguida, ele mostra a resposta do Amazon SWF.

Solicitação 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
 AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Algorithm=HmacSHA256,SignedHeaders=Host;X-Amz-
Date;X-Amz-Target;Content-
Encoding,Signature=IFJtq3M366CHqMlTpyqYqd9z0ChCoKDC5SCJBsLifu4=
Referer: http://swf.us-east-1.amazonaws.com/explorer/index.html
Content-Length: 21
Pragma: no-cache
Cache-Control: no-cache

{"name": "867530901"}
```

Resposta do 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 o protocolo (`HTTP/1.1`) é seguido por um código de status (`200`). Um valor de código de `200` indica uma operação bem-sucedida.

O Amazon SWF não serializa valores nulos. Se o analisador JSON estiver configurado para serializar valores nulos para solicitações, o Amazon SWF os ignorará.

Cálculo da assinatura HMAC-SHA para o Amazon SWF

Toda solicitação ao Amazon SWF deve ser autenticada. Eles assinam AWS SDKs automaticamente suas solicitações e gerenciam sua autenticação baseada em tokens. No entanto, se você quiser

escrever suas próprias solicitações HTTP `POST`, precisará criar um valor `x-amzn-authorization` para o conteúdo HTTP `POST` `Header` como parte da autenticação da sua solicitação.

Para obter mais informações sobre cabeçalhos de formatação, consulte Conteúdo de cabeçalho HTTP. Para a AWS SDK para Java implementação da assinatura da AWS versão 3, consulte a classe AWSSigner.java.

Criação de uma assinatura de solicitação

Antes de criar uma assinatura de solicitação HMAC-SHA, você deve obter suas credenciais da AWS (o ID de chave de acesso e a chave secreta).

> **Important**
>
> Você pode usar um SHA1 ou SHA256 para assinar suas solicitações. No entanto, certifique-se de usar o mesmo método por todo o processo de assinatura. O método escolhido deve corresponder ao valor do nome `Algorithm` no cabeçalho HTTP.

Para criar a assinatura de solicitação

1. Crie uma forma canônica dos cabeçalhos de solicitação HTTP. A forma canônica do cabeçalho HTTP inclui o seguinte:

   - `host`
   - Qualquer elemento de cabeçalho que comece com `x-amz-`

   Para obter mais informações sobre os cabeçalhos incluídos, consulte Conteúdo de cabeçalho HTTP.

   a. Para cada par de nome/valor de cabeçalho, converta o nome do cabeçalho (mas não o valor do cabeçalho) em letras minúsculas.

   b. Construa um mapa do nome do cabeçalho para valores de cabeçalho separados por vírgula.

   ```
   x-amz-example: value1
   x-amz-example: value2  =>  x-amz-example:value1,value2
   ```

   Para obter mais informações, consulte a Seção 4.2 da RFC 2616.

c. Converta cada par de nome/valor de cabeçalho em uma string no formato `headerName:headerValue`. Remova todos os espaços em branco do início e do final de `headerName` e `headerValue`, sem espaços em cada lado dos dois pontos.

```
x-amz-example1:value1,value2
x-amz-example2:value3
```

d. Insira uma nova linha (`U+000A`) após cada string convertida, incluindo a última string.

e. Classifique a coleção de strings convertidas em ordem alfabética, por nome de cabeçalho.

2. Crie um string-to-sign valor que inclua os seguintes itens:

- Linha 1: o método HTTP (`POST`), seguido por uma nova linha.
- Linha 2: o URI da solicitação (`/`), seguido por uma nova linha.
- Linha 3: uma string vazia seguida por uma nova linha.

> **Note**
>
> Normalmente, a string de consulta aparece aqui, mas o Amazon SWF não usa uma string de consulta.

- Linhas `4-n`: a string que representa os cabeçalhos de solicitação canônicos que você calculou na Etapa 1, seguida por uma nova linha. Essa nova linha cria uma linha em branco entre os cabeçalhos e o corpo da solicitação HTTP. Para obter mais informações, consulte RFC 2616.
- O corpo da solicitação, não seguido por uma nova linha.

3. Calcule o SHA1 resumo SHA256 ou resumo do string-to-sign valor. Use o mesmo método SHA por todo o processo.

4. Compute e codifique em Base64 o HMAC-SHA usando um SHA256 ou um SHA1 resumo (dependendo do método usado) do valor resultante da etapa anterior e da chave de acesso secreta temporária do Security Token Service usando a ação da AWS API. `GetSessionToken`

> **Note**
>
> O Amazon SWF espera um sinal de igual (=) no final do valor HMAC-SHA codificado em Base64. Se a sua rotina de codificação em Base64 não incluir o sinal de igual acrescentado, acrescente um ao final do valor.

Para obter mais informações sobre o uso de credenciais de segurança temporárias com o Amazon SWF e AWS outros serviços, AWS consulte Serviços que funcionam com o IAM no Guia do usuário do IAM.

5. Coloque o valor resultante como o valor do nome `Signature` no cabeçalho `x-amzn-authorization` da solicitação HTTP para o Amazon SWF.
6. O Amazon SWF verifica a solicitação e executa a operação especificada.

Lista de ações do Amazon SWF por categoria

Esta seção lista os tópicos de referência para ações do Amazon SWF na interface de programação de aplicações (API) do Amazon SWF. Eles são listados por categoria funcional.

Para obter uma lista alfabética de ações, consulte a Amazon Simple Workflow Service API Reference.

Tópicos

- Ações relacionadas a atividades
- Ações relacionadas a agentes de decisão
- Ações relacionadas a execuções de fluxo de trabalho
- Ações relacionadas à administração
- Ações de visibilidade

Ações relacionadas a atividades

Operadores de atividade usam `PollForActivityTask` para receber novas tarefas de atividade. Depois que um operador recebe uma tarefa de atividade do Amazon SWF, ele executa essa tarefa e responde usando `RespondActivityTaskCompleted` ser ela for bem-sucedida ou `RespondActivityTaskFailed` se ela falhar.

As ações a seguir são realizadas por operadores de atividade.

- `PollForActivityTask`
- `RespondActivityTaskCompleted`
- `RespondActivityTaskFailed`

- `RespondActivityTaskCanceled`
- `RecordActivityTaskHeartbeat`

Ações relacionadas a agentes de decisão

Agentes de decisão usam `PollForDecisionTask` para obter tarefas de decisão. Depois que um decisor recebe uma tarefa de administração do Amazon SWF, ele examina o histórico de execuções de fluxo de trabalho e decide o que fazer a seguir. Ele chama `RespondDecisionTaskCompleted` para concluir a tarefa de decisão e fornece zero ou mais próxima decisões.

As ações a seguir são realizadas por agentes de decisão.

- `PollForDecisionTask`
- `RespondDecisionTaskCompleted`

Ações relacionadas a execuções de fluxo de trabalho

As ações a seguir operam em uma execução de fluxo de trabalho.

- `RequestCancelWorkflowExecution`
- `StartWorkflowExecution`
- `SignalWorkflowExecution`
- `TerminateWorkflowExecution`

Ações relacionadas à administração

Embora seja possível executar tarefas administrativas no console do Amazon SWF, você pode usar as ações desta seção para automatizar funções ou criar suas próprias ferramentas administrativas.

Gerenciamento de atividade

- `RegisterActivityType`
- `DeprecateActivityType`
- `UndeprecateActivityType`
- `DeleteActivityType`

Gerenciamento de fluxo de trabalho

- `RegisterWorkflowType`
- `DeprecateWorkflowType`
- `UndeprecateWorkflowType`
- `DeleteWorkflowType`

Gerenciamento de domínio

Essas ações permitem registrar e descontinuar os domínios do Amazon SWF.

- `RegisterDomain`
- `DeprecateDomain`
- `UndeprecateDomain`

Para obter mais informações e exemplos dessas ações de gerenciamento de domínios, consulte Registrando um domínio com o Amazon SWF.

Gerenciamento de execução de fluxo de trabalho

- `RequestCancelWorkflowExecution`
- `TerminateWorkflowExecution`

Ações de visibilidade

Embora seja possível executar ações de visibilidade no console do Amazon SWF, você pode usar as ações desta seção para criar seu próprio console ou ferramentas administrativas.

Visibilidade de Atividade

- `ListActivityTypes`
- `DescribeActivityType`

Visibilidade do fluxo de trabalho

- ListWorkflowTypes

- DescribeWorkflowType

Visibilidade de execução de fluxo de trabalho

- DescribeWorkflowExecution
- ListOpenWorkflowExecutions
- ListClosedWorkflowExecutions
- CountOpenWorkflowExecutions
- CountClosedWorkflowExecutions
- GetWorkflowExecutionHistory

Visibilidade de domínio

- ListDomains
- DescribeDomain

Visibilidade de lista de tarefas

- CountPendingActivityTasks
- CountPendingDecisionTasks

Registrando um domínio com o Amazon SWF

Seus tipos de fluxo de trabalho e atividade e a execução de fluxo de trabalho em si têm escopo definido em um domínio. Domínios isolam um conjunto de tipos, execuções e listas de tarefas de outros na mesma conta.

Você pode registrar um domínio usando o AWS Management Console ou usando a `RegisterDomain` ação na API Amazon SWF. O exemplo a seguir usa a API.

```
https://swf.us-east-1.amazonaws.com
RegisterDomain
{
  "name" : "867530901",
  "description" : "music",
```

```
  "workflowExecutionRetentionPeriodInDays" : "60"
}
```

Os parâmetros são especificados no formato JavaScript Object Notation (JSON). Aqui, o período de retenção está definido como 60 dias. Durante o período de retenção, todas as informações sobre a execução do fluxo de trabalho estão disponíveis por meio de operações de visibilidade usando a API AWS Management Console ou a API do Amazon SWF.

Depois de registrar o domínio, você deverá registrar o tipo de fluxo de trabalho e os tipos de atividade usados pelo fluxo de trabalho. Você precisa registrar o domínio primeiro, pois um nome de domínio registrado faz parte das informações necessárias para o registro de tipos de fluxo de trabalho e atividade.

Consulte também

`RegisterDomain` na Amazon Simple Workflow Service API Reference

Definindo valores de tempo limite no Amazon SWF

Tópicos

- Cotas em valores de tempo limite
- Tempos limite de execução de fluxo de trabalho e tarefas de decisão
- Tempo limite de tarefas de atividade
- Consulte também

Cotas em valores de tempo limite

Os valores de tempo limite são sempre declarados em segundos e podem ser definidos para qualquer número de segundos até um ano (31.536.000 segundos): o limite máximo de execução de qualquer fluxo de trabalho ou atividade. O valor especial `NONE` é usado para definir um parâmetro de tempo limite como "sem tempo limite", ou infinito, mas o limite máximo de um ano ainda é aplicável.

Tempos limite de execução de fluxo de trabalho e tarefas de decisão

Você pode definir valores de tempo limite para suas tarefas de fluxo de trabalho e decisão ao registrar o tipo de fluxo de trabalho. Por exemplo:

```
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"
}
```

Esse registo de tipo de fluxo de trabalho define `defaultTaskStartToCloseTimeout` como 600 segundos (10 minutos) e `defaultExecutionStartToCloseTimeout` como 3600 segundos (1 hora).

Para obter mais informações sobre o registro do tipo de fluxo de trabalho, consulte Registro de um tipo de fluxo de trabalho com o Amazon SWF e `RegisterWorkflowType` a Amazon Simple Workflow Service API Reference.

É possível substituir o valor definido por `defaultExecutionStartToCloseTimeout`, especificando `executionStartToCloseTimeout` .

Tempo limite de tarefas de atividade

Você pode definir valores de tempo limite para suas tarefas de atividade ao registrar o tipo de atividade. Por exemplo:

```
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"
```

```
}
```

Esse registro de tipo de atividade define `defaultTaskStartToCloseTimeout` como 600 segundos (10 minutos), o `defaultTaskHeartbeatTimeout` como 120 segundos (2 minutos), `defaultTaskScheduleToStartTimeout` como 1800 segundos (30 minutos) e `defaultTaskScheduleToCloseTimeout` como 5400 segundos (1,5 horas).

Para obter mais informações sobre o registro do tipo de atividade, consulte Registro de um tipo de atividade com o Amazon SWF e `RegisterActivityType` a Amazon Simple Workflow Service API Reference.

Você pode substituir o valor definido para `defaultTaskStartToCloseTimeout` especificando `taskStartToCloseTimeout` ao programar a tarefa de atividade.

Consulte também

Tipos de tempo limite do Amazon SWF

Registro de um tipo de fluxo de trabalho com o Amazon SWF

O exemplo discutido nesta seção registra um tipo de fluxo de trabalho usando a Amazon SWF API. O nome e a versão que você especificou durante o registro formam um identificador exclusivo para o tipo de fluxo de trabalho. O domínio especificado já deve ter sido registrado usando a ação de API `RegisterDomain`.

Os parâmetros de tempo limite no exemplo a seguir são valores de duração especificados em segundos. Para o parâmetro `defaultTaskStartToCloseTimeout`, você pode usar o especificador de duração `NONE` para indicar nenhum tempo limite. No entanto, não pode especificar um valor de `NONE` para `defaultExecutionStartToCloseTimeout`. Há um limite máximo de um ano para o tempo em que uma execução de fluxo de trabalho pode ser executada. Exceder este limite sempre faz com que a execução de fluxo de trabalho atinja o tempo limite. Se você especificar um valor para `defaultExecutionStartToCloseTimeout` que seja maior que um ano, o registro falhará.

```
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"
}
```

Consulte também

RegisterWorkflowType na Amazon Simple Workflow Service API Reference

Registro de um tipo de atividade com o Amazon SWF

O exemplo a seguir registra um tipo de atividade usando a Amazon SWF API. O nome e a versão que você especificar durante o registro formam um identificador exclusivo para o tipo de atividade dentro do domínio. O domínio especificado já deve ter sido registrado usando a ação `RegisterDomain`.

Os parâmetros de tempo limite neste exemplo são valores de duração especificados em segundos. Você pode usar o especificador de duração `NONE` para indicar nenhum tempo limite.

```
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"
}
```

Consulte também

RegisterActivityType na Amazon Simple Workflow Service API Reference

AWS Lambda tarefas no Amazon SWF

Tópicos

- Sobre AWS Lambda
- Benefícios e limitações do uso de tarefas Lambda
- Usar tarefas do Lambda em seus fluxos de trabalho

Sobre AWS Lambda

AWS Lambda é um serviço computacional totalmente gerenciado que executa seu código em resposta a eventos gerados por código personalizado ou de vários AWS serviços, como Amazon S3, DynamoDB, Amazon Kinesis, Amazon SNS e Amazon Cognito. Para obter mais informações sobre o Lambda, consulte o Manual do desenvolvedor do AWS Lambda.

O Amazon Simple Workflow Service fornece uma tarefa Lambda para que você possa executar funções do Lambda no lugar ou ao lado das atividades tradicionais do Amazon SWF.

> **Important**
>
> Sua AWS conta será cobrada pelas execuções (solicitações) do Lambda executadas pelo Amazon SWF em seu nome. Para obter detalhes sobre os preços do Lambda, consulte https://aws.amazon.com/lambda/ preços/.

Benefícios e limitações do uso de tarefas Lambda

Há vários benefícios no uso de tarefas Lambda em vez de uma atividade tradicional do Amazon SWF:

- As tarefas Lambda não precisam ser registradas ou versionadas como os tipos de atividade do Amazon SWF.
- Você pode usar quaisquer funções do Lambda existentes que já tenha definido em seus fluxos de trabalho.
- As funções do Lambda são chamadas diretamente pelo Amazon SWF; não há necessidade de implementar um programa de trabalho para executá-las, como deve ser feito com as atividades tradicionais.

- O Lambda fornece métricas e logs para rastrear e analisar as execuções das funções.

Há também uma série de limitações em relação a tarefas Lambda sobre as quais você deve estar ciente:

- As tarefas do Lambda só podem ser executadas em AWS regiões que fornecem suporte para o Lambda. Consulte Regiões e endpoints do Lambda na Referência geral da Amazon Web Services para obter detalhes sobre as regiões atualmente compatíveis com o Lambda.
- Atualmente, as tarefas Lambda são suportadas somente pela API HTTP SWF básica e pelo AWS Flow Framework for Java. Atualmente, não há suporte para tarefas Lambda no AWS Flow Framework for Ruby.

Usar tarefas do Lambda em seus fluxos de trabalho

Para usar as tarefas Lambda em seus fluxos de trabalho do Amazon SWF, você precisará:

1. Configure os perfis do IAM para fornecer ao Amazon SWF permissão para invocar funções do Lambda.
2. Anexar as funções do IAM aos seus fluxos de trabalho.
3. Chamar sua função Lambda durante uma execução de fluxo de trabalho.

Configurar uma função do IAM

Antes de poder invocar as funções do Lambda a partir do Amazon SWF, você deve fornecer um perfil do IAM que forneça acesso ao Lambda a partir do Amazon SWF. Você também pode:

- escolha uma função predefinida, Role, AWSLambda para dar permissão aos seus fluxos de trabalho para invocar qualquer função Lambda associada à sua conta.
- defina sua própria política e função associada para dar permissão aos fluxos de trabalho para invocar funções específicas do Lambda, especificadas por seus Amazon Resource Names (). ARNs

Limite de permissões de um perfil do IAM

Você pode limitar as permissões em um perfil do IAM fornecido ao Amazon SWF usando as chaves de contexto `SourceArn` e `SourceAccount` em sua política de confiança de recursos. Essas chaves

limitam o uso de uma política do IAM para que ela seja usada somente em execuções do Amazon Simple Workflow Service que pertençam ao domínio ARN especificado. Se você usar ambas as chaves de contexto de condição global, o valor `aws:SourceAccount` e a conta referenciada no valor `aws:SourceArn` deverão usar o mesmo ID de conta quando usados na mesma declaração de política.

No exemplo de política de confiança a seguir, usamos a chave de contexto `SourceArn` para restringir o perfil de serviço do IAM a ser usado somente nas execuções do Amazon Simple Workflow Service que pertençam a `someDomain` na conta `123456789012`.

```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Sid": "",
       "Effect": "Allow",
       "Principal": {
         "Service": "swf.amazonaws.com"
       },
       "Action": "sts:AssumeRole",
       "Condition": {
         "ArnLike": {
           "aws:SourceArn": "arn:aws:swf:*:123456789012:/domain/someDomain"
       }
      }
    }
  ]
}
```

No exemplo de política de confiança a seguir, usamos a chave de contexto `SourceAccount` para restringir o perfil de serviço do IAM a ser usado somente em execuções do Amazon Simple Workflow Service na conta `123456789012`.

```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Sid": "",
       "Effect": "Allow",
       "Principal": {
         "Service": "swf.amazonaws.com"
       },
```

```
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringLike": {
            "aws:SourceAccount": "123456789012"
        }
       }
     }
   ]
}
```

Fornecer ao Amazon SWF acesso para invocar qualquer função do Lambda

Você pode usar a função predefinida, Role, para dar aos fluxos de trabalho do Amazon SWF a capacidade de invocar qualquer AWSLambdafunção Lambda associada à sua conta.

Para usar a AWSLambda função para dar ao Amazon SWF acesso para invocar funções do Lambda

1. Abra o console do Amazon IAM.
2. Escolha Roles e depois Create New Role.
3. Dê um nome para a sua função, como `swf-lambda` e escolha Next Step.
4. Em Perfis de serviço da AWS , escolha Amazon SWF e selecione Próxima etapa.
5. Na tela Anexar política, escolha AWSLambdaFunção na lista.
6. Escolha Next Step e depois Create Role depois de analisar a função.

Definir um perfil do IAM para fornecer acesso para invocar uma função do Lambda específica

Se quiser fornecer acesso para invocar uma função do Lambda específica do seu fluxo de trabalho, você precisará definir sua própria política do IAM.

Para criar uma política do IAM para fornecer acesso a uma função do Lambda específica

1. Abra o console do Amazon IAM.
2. Escolha Policies e depois Create Policy.
3. Escolha Copiar uma política AWS gerenciada e selecione AWSLambdaFunção na lista. Uma política será gerada para você. Opcionalmente, edite seu nome e descrição para atender às suas necessidades.
4. No campo Recurso do Documento de política, adicione o ARN de uma ou mais funções do Lambda. Por exemplo:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:us-east-1:111111000000:function:hello_lambda_function"
      ]
    }
  ]
}
```

> **Note**
>
> Para obter uma descrição completa de como especificar recursos em um perfil do IAM, consulte Visão geral das políticas do IAM em Uso do IAM.

5. Escolha Create Policy para concluir a criação da sua política.

Em seguida, você pode selecionar essa política ao criar um perfil do IAM e usá-lo para conceder acesso de invocação aos seus fluxos de trabalho do Amazon SWF. Esse procedimento é muito semelhante à criação de uma função com a política de AWSLambdafunção. Em vez disso, escolha sua própria política ao criar a função.

Para criar uma função do Amazon SWF usando sua política Lambda

1. Abra o console do Amazon IAM.
2. Escolha Roles e depois Create New Role.
3. Dê um nome para a sua função, como `swf-lambda-function` e escolha Next Step.
4. Em Perfis de serviço da AWS , escolha Amazon SWF e selecione Próxima etapa.
5. Em Anexar política, escolha a política específica da função do Lambda na lista.
6. Escolha Next Step e depois Create Role depois de analisar a função.

Anexar a função do IAM ao seu fluxo de trabalho

Depois de definir seu perfil do IAM, você precisará anexá-lo ao fluxo de trabalho que a usará para chamar as funções do Lambda às quais você forneceu acesso ao Amazon SWF.

Existem dois lugares em que você pode anexar a função ao seu fluxo de trabalho:

- Durante o registro do tipo de fluxo de trabalho. Essa função pode ser usada como a função Lambda padrão para cada execução desse tipo de fluxo de trabalho.
- Ao iniciar uma execução de fluxo de trabalho. Essa função será usada somente durante a execução desse fluxo de trabalho (e durante toda a execução).

Para fornecer uma função Lambda padrão para um tipo de fluxo de trabalho

- Ao chamar RegisterWorkflowType, defina o defaultLambdaRole campo como o ARN da função que você definiu.

Para fornecer uma função Lambda a ser usada durante uma execução de fluxo de trabalho

- Ao chamar StartWorkflowExecution, defina o campo LambdaRole como o ARN da função que você definiu.

> **Note**
>
> se a conta estiver ligando RegisterWorkflowType ou StartWorkflowExecution não tiver permissão para usar a função especificada, a chamada falhará com um OperationNotPermittedFault.

Chame a função do Lambda por meio de um fluxo de trabalho do Amazon SWF

Você pode usar o tipo de ScheduleLambdaFunctionDecisionAttributes dados para identificar a função Lambda a ser chamada durante a execução de um fluxo de trabalho.

Durante uma ligação para RespondDecisionTaskCompleted, forneça uma ScheduleLambdaFunctionDecisionAttributes à sua lista de decisões. Por exemplo:

```
{
```

```
  "decisions": [{
    "ScheduleLambdaFunctionDecisionAttributes": {
      "id": "lambdaTaskId",
      "name": "myLambdaFunctionName",
      "input": "inputToLambdaFunction",
      "startToCloseTimeout": "30"
    },
  }],
}
```

Defina os seguintes parâmetros:

- id com um identificador para a tarefa Lambda. Deve ser uma string de 1 a 256 caracteres e não deve conter os caracteres : (dois pontos), / (barra), | (barra vertical) nem caracteres de controle (\u0000 - \u001f and \u007f - \u009f) ou a string literal `arn`.
- name com o nome da sua função Lambda. Seu fluxo de trabalho do Amazon SWF deve ser fornecido com um perfil do IAM que lhe dê acesso para chamar a função do Lambda. O nome fornecido deve seguir as restrições do FunctionName parâmetro, como na ação Lambda Invoke.
- input com dados de entrada opcionais para a função. Se definido, isso deve seguir as restrições do ClientContext parâmetro, como na ação Lambda Invoke.
- startToCloseTempo limite com um período máximo opcional, em segundos, que a função pode levar para ser executada antes que a tarefa falhe com uma exceção de tempo limite. O valor NONE pode ser usado para especificar duração ilimitada.

Para obter mais informações, consulte Implementando AWS Lambda tarefas

Desenvolver um operador de atividade no Amazon SWF

Um operador de atividade fornece a implementação de um ou mais tipos de atividades. Um operador de atividade se comunica com o Amazon SWF para receber tarefas de atividade e executá-las. Você pode ter uma frota de vários operadores de atividade realizando tarefas de atividade do mesmo tipo de atividade.

O Amazon SWF disponibiliza uma tarefa de atividade para os operadores de atividade quando o agente de decisão agenda a tarefa de atividade. Quando um agente de decisão agenda uma tarefa de atividade, ele fornece os dados (que você determina) que o operador de atividade precisa para executar a tarefa de atividade. O Amazon SWF insere esses dados na tarefa da atividade antes de enviá-los ao funcionário da atividade.

Os operadores de atividade são gerenciados por você. Eles podem ser escritos em qualquer linguagem. Um operador pode ser executado em qualquer lugar, desde que possa se comunicar com o Amazon SWF por meio da API. Como o Amazon SWF fornece todas as informações necessárias para executar uma tarefa de atividade, todos os operadores de atividade podem ser stateless. O estado stateless permite que seus fluxos de trabalho sejam altamente escaláveis. Para lidar com os maiores requisitos de capacidade, basta adicionar mais operadores de atividade.

Esta seção explica como implementar um operador de atividade. Os operadores de atividade devem fazer repetidamente o seguinte.

1. Pesquise o Amazon SWF para obter uma tarefa de atividade.
2. Iniciar a execução da tarefa.
3. Relatar periodicamente um heartbeat ao Amazon SWF se a tarefa tiver vida longa.
4. Informar que a tarefa foi concluída ou falhou e retornar os resultados para o Amazon SWF.

Tópicos

- Sondar tarefas de atividade
- Realizar a tarefa de atividade
- Informar heartbeats de tarefas de atividade
- Concluir ou falhar uma tarefa de atividade
- Iniciar operadores de atividade

Sondar tarefas de atividade

Para executar tarefas de atividade, cada operador de atividade deve sondar o Amazon SWF chamando periodicamente a ação `PollForActivityTask`.

No exemplo a seguir, o operador de atividade `ChargeCreditCardWorker01` faz uma sondagem em busca de uma tarefa na lista de tarefas, `ChargeCreditCard-v0.1`. Se nenhuma tarefa de atividade estiver disponível, após 60 segundos, o Amazon SWF enviará de volta uma resposta vazia. Uma resposta vazia é uma estrutura `Task` na qual o valor do `taskToken` é uma string vazia.

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

```
  "domain" : "867530901",
  "taskList" : { "name": "ChargeCreditCard-v0.1" },
  "identity" : "ChargeCreditCardWorker01"
}
```

Se uma tarefa de atividade ficar disponível, o Amazon SWF a devolverá ao operador de atividade. A tarefa contém os dados que o agente de decisão especifica quando ele agenda a atividade.

Depois que um operador de atividade receber uma tarefa de atividade, ele estará pronto para realizar o trabalho. A próxima seção fornece informações sobre como realizar uma tarefa de atividade.

Realizar a tarefa de atividade

Depois de receber uma tarefa de atividade, o operador da atividade está pronto para realizá-la.

Para realizar uma tarefa de atividade

1. Programe seu operador de atividade para interpretar o conteúdo no campo de entrada da tarefa. Esse campo contém os dados especificados pelo agente de decisão quando a tarefa foi agendada.
2. Programe o operador de atividade para começar a processar os dados e executar sua lógica.

A próxima seção descreve como programar seus operadores de atividade para fornecer atualizações de status ao Amazon SWF para atividades de longa duração.

Informar heartbeats de tarefas de atividade

Se um tempo limite de heartbeat tiver sido registrado com o tipo de atividade, o operador de atividade deverá registrar um heartbeat antes que esse tempo limite seja excedido. Se uma tarefa de atividade não fornecer um batimento cardíaco dentro do tempo limite, a tarefa atingirá o tempo limite, o Amazon SWF a fechará e agendará uma nova tarefa de decisão para informar um agente de decisão sobre o tempo limite. Em seguida, o agente de decisão poderá reprogramar a tarefa de atividade ou realizar outra ação.

Se, após o tempo limite, o operador de atividade tentar entrar em contato com o Amazon SWF, por exemplo, chamando `RespondActivityTaskCompleted`, o Amazon SWF retornará uma falha `UnknownResource`.

Esta seção descreve como fornecer um heartbeat de atividade.

Para registrar um heartbeat de tarefa de atividade, programe seu operador de atividade para chamar a ação `RecordActivityTaskHeartbeat`. Essa ação também fornece um campo de string que você pode usar para armazenar dados de forma livre para quantificar o progresso de qualquer maneira que funcione para o seu aplicativo.

Neste exemplo, o operador de atividade informa o batimento cardíaco para o Amazon SWF e usa o campo de detalhes para informar que a tarefa da atividade está 40% concluída. Para informar o heartbeat, o operador de atividade deve especificar o token da tarefa de atividade.

```
https://swf.us-east-1.amazonaws.com
RecordActivityTaskHeartbeat
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "details" : "40"
}
```

Por si só, essa ação não cria um evento no histórico de execução de fluxo de trabalho. Porém, se a tarefa atingir o tempo limite, o histórico de execução de fluxo de trabalho conterá um evento `ActivityTaskTimedOut` com as informações do último heartbeat gerado pelo operador de atividade.

Concluir ou falhar uma tarefa de atividade

Depois de executar uma tarefa, o operador de atividade deve informar se a tarefa de atividade foi concluída ou falhou.

Concluir uma tarefa de atividade

Para concluir uma tarefa de atividade, programe o operador de atividade para chamar a ação `RespondActivityTaskCompleted` depois que ele concluir com êxito uma tarefa de atividade, especificando o token de tarefa.

Neste exemplo, o operador de atividade indica que a tarefa foi concluída com êxito.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "results": "40"
}
```

Quando a atividade é concluída, o Amazon SWF agenda uma nova tarefa de decisão para a execução do fluxo de trabalho ao qual a atividade está associada.

Programe o operador de atividade para sondar outra tarefa de atividade depois que ele concluir a tarefa em questão. Isso cria um loop em que o operador de atividade sonda tarefas e as conclui continuamente.

Se a atividade não responder dentro do StartToCloseTimeoutperíodo ou se ScheduleToCloseTimeouttiver sido excedida, o Amazon SWF expira o tempo limite da tarefa da atividade e agenda uma tarefa de decisão. Isso permite que um agente de decisão tome uma ação apropriada, como reagendar a tarefa.

Por exemplo, se uma EC2 instância da Amazon estiver executando uma tarefa de atividade e a instância falhar antes que a tarefa seja concluída, o decisor receberá um evento de tempo limite no histórico de execução do fluxo de trabalho. Se a tarefa da atividade estiver usando uma pulsação, o decisor receberá o evento quando a tarefa falhar em entregar a próxima pulsação após a falha da instância da Amazon EC2 . Caso contrário, o agente de decisão acabará recebendo o evento quando a tarefa de atividade não for concluída antes de atingir um de seus valores de tempo limite gerais. Em seguida, caberá ao agente de decisão reatribuir a tarefa ou tomar alguma outra medida.

Falhar uma tarefa de atividade

Se um operador de atividade não puder executar uma tarefa de atividade por algum motivo, mas ainda puder se comunicar com o Amazon SWF, você poderá programá-lo para falhar na tarefa.

Para programar um operador de atividade para marcar uma tarefa de atividade como falha, programe-o para chamar a ação `RespondActivityTaskFailed`, que especifica o token dessa tarefa.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskFailed
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "reason" : "CC-Invalid",
  "details" : "Credit Card Number Checksum Failed"
}
```

Como desenvolvedor, você define os valores que são armazenados nos campos de motivo e detalhes. Essas são strings de forma livre; você pode usar qualquer convenção de código de erro

que sirva à aplicação. O Amazon SWF não processa esses valores. No entanto, o Amazon SWF pode exibir esses valores no console.

Quando uma tarefa de atividade falha, o Amazon SWF agenda uma tarefa de decisão para a execução do fluxo de trabalho ao qual a tarefa de atividade está associada para informar o agente de decisão sobre a falha. Programe seu agente de decisão para lidar com atividades com falha, por exemplo, reagendando uma atividade ou marcando a execução de fluxo de trabalho como falha, dependendo da natureza da falha.

Iniciar operadores de atividade

Para iniciar operadores de atividade, empacote sua lógica em um executável que você possa usar na sua plataforma de operadores de atividade. Por exemplo, você pode empacotar o código da atividade como um executável Java que pode ser executado em servidores Linux e Windows.

Uma vez iniciados, seus operadores começarão a fazer sondagens em busca de tarefas. Porém, até que o agente de decisão agende tarefas de atividade, essas sondagens atingirão o tempo limite sem tarefas, e seus operadores apenas continuarão a realizá-las.

Como as sondagens são solicitações de saída, o operador de atividade pode ser executado em qualquer rede que tenha acesso ao endpoint do Amazon SWF.

Você pode iniciar quantos operadores de atividade quiser. À medida que o agente de decisão agenda as tarefas de atividade, o Amazon SWF distribui automaticamente as tarefas de atividade para os operadores de atividade de sondagem.

Desenvolvendo decisores no Amazon SWF

Um agente de decisão é uma implementação da lógica de coordenação do seu tipo de fluxo de trabalho que é processada durante a execução do seu fluxo de trabalho. É possível executar vários agentes de decisão para um único tipo de fluxo de trabalho.

Como o estado de execução de um fluxo de trabalho é armazenado em seu histórico de fluxo de trabalho, os agentes de decisão podem ficar sem estado. O Amazon SWF mantém o histórico de execução do fluxo de trabalho e o fornece a um agente de decisão em cada tarefa de decisão. Isso permite que você adicione e remova agentes de decisão dinamicamente conforme necessário, o que torna o processamento dos seus fluxos de trabalho altamente escalável. À medida que a carga no seu sistema crescer, basta adicionar mais agentes de decisão para lidar com a capacidade

aumentada. Porém, observe que pode haver apenas uma tarefa de decisão aberta a qualquer momento para uma determinada execução de fluxo de trabalho.

Toda vez que ocorre uma mudança de estado para a execução de um fluxo de trabalho, o Amazon SWF agenda uma tarefa de decisão. Sempre que um agente de decisão recebe uma tarefa de decisão, ele faz o seguinte:

- Interpreta o histórico de execução de fluxo de trabalho fornecido com a tarefa de decisão
- Aplica a lógica de coordenação com base no histórico de execução de fluxo de trabalho e toma decisões sobre o que fazer em seguida. Cada decisão é representada por uma estrutura de decisão
- Conclui a tarefa de decisão e fornece uma lista de decisões ao Amazon SWF.

Esta seção descreve como desenvolver um agente de decisão, o que envolve:

- Programar o agente de decisão para sondar tarefas de decisão
- Programar o agente de decisão para interpretar o histórico de execução de fluxo de trabalho e tomar decisões
- Programar o agente de decisão para responder a uma tarefa de decisão.

Os exemplos desta seção mostram como é possível programar um agente de decisão para o exemplo de fluxo de trabalho de comércio eletrônico.

Você pode implementar o agente de decisão em qualquer linguagem que desejar e executá-lo em qualquer lugar, desde que ele possa se comunicar com o Amazon SWF por meio de sua API de serviço.

Tópicos

- Definir a lógica de coordenação
- Sondar tarefas de decisão
- Aplicar a lógica de coordenação
- Responder com decisões
- Fechar uma execução de fluxo de trabalho
- Iniciar agentes de decisão

Definir a lógica de coordenação

A primeira coisa a fazer ao desenvolver um agente de decisão é definir a lógica de coordenação. No exemplo de comércio eletrônico, a lógica de coordenação que agenda cada atividade após a conclusão da atividade anterior pode parecer semelhante à seguinte:

```
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
```

O agente de decisão aplica a lógica de coordenação ao histórico de execução de fluxo de trabalho e cria uma lista de decisões ao concluir a tarefa de decisão usando a ação `RespondDecisionTaskCompleted`.

Sondar tarefas de decisão

Cada agente de decisão faz uma sondagem em busca de tarefas de decisão. As tarefas de decisão contêm as informações que o agente de decisão utiliza para gerar decisões, como agendar tarefas de atividade. Para sondar tarefas de decisão, o agente de decisão usa a ação `PollForDecisionTask`.

Neste exemplo, o agente de decisão faz uma sondagem em busca de uma tarefa de decisão, especificando a lista de tarefas `customerOrderWorkflow-0.1`.

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

```
  "domain": "867530901",
  "taskList": {"name": "customerOrderWorkflow-v0.1"},
  "identity": "Decider01",
  "maximumPageSize": 50,
  "reverseOrder": true
}
```

Se uma tarefa de decisão estiver disponível na lista de tarefas especificada, o Amazon SWF a retornará imediatamente. Se nenhuma tarefa de decisão estiver disponível, o Amazon SWF manterá a conexão aberta por até 60 segundos e retornará uma tarefa assim que ela estiver disponível. Se nenhuma tarefa estiver disponível, o Amazon SWF retornará uma resposta vazia. Uma resposta vazia é uma estrutura `Task` na qual o valor de `taskToken` é uma string vazia. Certifique-se de programar seu agente de decisão para sondar outra tarefa se ele receber uma resposta vazia.

Se uma tarefa de decisão estiver disponível, o Amazon SWF retornará uma resposta que contém a tarefa de decisão, bem como uma exibição paginada do histórico de execução do fluxo de trabalho.

Neste exemplo, o tipo de evento mais recente indica que a execução de fluxo de trabalho foi iniciada e que o elemento de entrada contém as informações necessárias para realizar a primeira tarefa.

```
{
  "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"
        }
      }
    }
  ],
  ...
}
```

Depois de receber o histórico de execução de fluxo de trabalho, o agente de decisão interpreta esse histórico e toma decisões com base em sua lógica de coordenação.

Como o número de eventos do histórico de fluxo de trabalho para uma única execução de fluxo de trabalho pode ser grande, o resultado retornado pode ser dividido em várias páginas. Para recuperar as páginas subsequentes, faça chamadas adicionais `PollForDecisionTask` usando o nextPageTokenretornado pela chamada inicial. Observe que você não liga `GetWorkflowExecutionHistory` com isso nextPageToken. Em vez disso, chame `PollForDecisionTask` novamente.

Aplicar a lógica de coordenação

Depois que o agente de decisão receber uma tarefa de decisão, programe-a para interpretar o histórico de execução de fluxo de trabalho e determinar o que aconteceu até agora. Com base nisso, ele deverá gerar uma lista de decisões.

No exemplo de comércio eletrônico, estamos preocupados apenas com o último evento no histórico de fluxo de trabalho e, portanto, definimos a seguinte 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
```

Se lastEvent for `CompleteVerifyOrderActivity`, você adicionará a atividade `ScheduleChargeCreditCardActivity` à lista de decisões.

Depois que o agente de decisão determina uma ou mais decisões a serem tomadas, ele pode responder ao Amazon SWF com decisões apropriadas.

Responder com decisões

Depois de interpretar o histórico do fluxo de trabalho e gerar uma lista de decisões, o agente de decisão está pronto para responder ao Amazon SWF com essas decisões.

Programe seu agente de decisão para extrair os dados necessários do histórico de execução de fluxo de trabalho e, em seguida, criar decisões que especificam as próximas ações apropriadas para o fluxo de trabalho. O agente de decisão transmite essa decisão de volta ao Amazon SWF usando a ação `RespondDecisionTaskCompleted`. Consulte a Amazon Simple Workflow Service API Reference para obter uma lista dos tipos de decisão disponíveis.

No exemplo de comércio eletrônico, quando o agente de decisão responde com o conjunto de decisões que ele gerou, ele também inclui a entrada de cartão de crédito do histórico de execução de fluxo de trabalho. Dessa forma, o operador de atividade tem as informações necessárias para realizar a tarefa de atividade.

Quando todas as atividades na execução de fluxo de trabalho estiverem concluídas, o agente de decisão encerrará a execução de fluxo de trabalho.

```
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"
      }
    }
  ]
}
```

Fechar uma execução de fluxo de trabalho

Quando o agente de decisão determinar que o processo de negócios está concluído, ou seja, que não há mais atividades para realizar, ele gerará uma decisão de encerrar a execução de fluxo de trabalho.

Para encerrar uma execução de fluxo de trabalho, programe seu agente de decisão para interpretar os eventos no histórico de fluxo de trabalho de modo a determinar o que aconteceu na execução até agora e verificar se ela deve ser encerrada.

Se o fluxo de trabalho tiver sido concluído com sucesso, encerre a execução de fluxo de trabalho chamando `RespondDecisionTaskCompleted` com a decisão `CompleteWorkflowExecution`. Como alternativa, você pode marcar uma execução incorreta como falha usando a decisão `FailWorkflowExecution`.

No exemplo de comércio eletrônico, o agente de decisão analisa o histórico e, com base na lógica de coordenação, adiciona uma decisão para encerrar a execução de fluxo de trabalho à sua lista de decisões e inicia uma ação `RespondDecisionTaskCompleted` com uma decisão de encerramento de fluxo de trabalho.

> **Note**
>
> Existem alguns casos em que o encerramento de uma execução de fluxo de trabalho falha. Por exemplo, se um sinal for recebido enquanto o agente de decisão estiver encerrando a execução de fluxo de trabalho, a decisão de encerramento falhará. Para lidar com essa possibilidade, certifique-se de que o agente de decisão continue a sondar tarefas de decisão. Além disso, garanta que o agente de decisão que recebe a próxima tarefa de decisão responda ao evento (nesse caso, um sinal) que impediu o encerramento da execução.

Você também pode oferecer suporte ao cancelamento de execuções de fluxo de trabalho. Isso pode ser especialmente útil para fluxos de trabalho de longa duração. Para oferecer suporte ao cancelamento, o agente de decisão deve lidar com o evento `WorkflowExecutionCancelRequested` na histórico. Esse evento indica que o cancelamento da execução foi solicitado. O agente de decisão deve realizar as ações de limpeza apropriadas, como cancelar tarefas de atividades contínuas e encerrar o fluxo de trabalho, chamando a ação `RespondDecisionTaskCompleted` com a decisão `CancelWorkflowExecution`.

O exemplo a seguir chama `RespondDecisionTaskCompleted` para especificar que a execução atual do fluxo de trabalho está cancelada.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions" : [
    {
      "decisionType":"CancelWorkflowExecution",
      "CancelWorkflowExecutionAttributes":{
        "Details": "Customer canceled order"
      }
    }
  ]
}
```

O Amazon SWF verifica para garantir que a decisão de fechar ou cancelar a execução do fluxo de trabalho seja a última decisão enviada pelo agente de decisão. Ou seja, não é válido ter um conjunto de decisões em que há decisões após aquela que encerra o fluxo de trabalho.

Iniciar agentes de decisão

Depois de concluir o desenvolvimento de agentes de decisão, você está pronto para iniciar um ou mais deles.

Para iniciar agentes de decisão, empacote sua lógica de coordenação em um executável que você possa usar na sua plataforma de agentes de decisão. Por exemplo, você pode empacotar o código do agente de decisão como um executável Java que pode ser executado em computadores Linux e Windows.

Uma vez iniciado, seus agente de decisão devem começar a sondar o Amazon SWF em busca de tarefas. Até que você inicie as execuções do fluxo de trabalho e o Amazon SWF programe as tarefas de decisão, essas enquetes terão um tempo limite e receberão respostas vazias. Uma resposta vazia é uma estrutura `Task` na qual o valor de `taskToken` é uma string vazia. Seus agentes de decisão devem simplesmente continuar a sondagem.

O Amazon SWF garante que apenas uma tarefa de decisão possa estar ativa para a execução de um fluxo de trabalho a qualquer momento. Isso evita problemas, como decisões conflitantes. Além disso, o Amazon SWF garante que uma única tarefa de decisão seja atribuída a um único agente de decisão, independentemente do número de agentes de decisão que estejam em execução.

Se ocorrer algo que gere uma tarefa de decisão enquanto um agente de decisão estiver processando outra tarefa de decisão, o Amazon SWF enfileira a nova tarefa até que a tarefa atual seja concluída. Após a conclusão da tarefa atual, o Amazon SWF disponibiliza a nova tarefa de decisão. Além disso, as tarefas de decisão são agrupadas, no sentido de que, se várias atividades forem concluídas enquanto um agente de decisão estiver processando uma tarefa de decisão, o Amazon SWF criará apenas uma única tarefa de decisão para contabilizar as várias conclusões de tarefas. No entanto, cada conclusão de tarefa receberá um evento individual no histórico de execução de fluxo de trabalho.

Como as pesquisas são solicitações de saída, os agentes de decisão podem ser executados em qualquer rede que tenha acesso ao endpoint do Amazon SWF.

Para que as execuções de fluxo de trabalho avancem, um ou mais agentes de decisão devem estar em execução. Inicie quantos agentes de decisão quiser. O Amazon SWF comporta a sondagem de vários agentes de decisão na mesma lista de tarefas.

Iniciando fluxos de trabalho no Amazon SWF

Você pode iniciar uma execução de fluxo de trabalho de um tipo de fluxo de trabalho registrado a partir de qualquer aplicativo usando a ação `StartWorkflowExecution`. Ao iniciar a execução, você associa a ela um identificador, chamado de `workflowId`. O `workflowId` pode ser qualquer string apropriada para o seu aplicativo, como o número do pedido em um aplicativo de processamento de pedidos. Você não pode usar o mesmo `workflowId` para várias execuções de fluxo de trabalho abertas no mesmo domínio. Por exemplo, se você iniciar duas execuções de fluxo de trabalho com o `workflowId` `Customer Order 01`, a segunda execução de fluxo de trabalho não será iniciada, e a solicitação falhará. No entanto, você pode reutilizar o `workflowId` de uma execução fechada. O Amazon SWF também associa um identificador exclusivo gerado pelo sistema, chamado `runId`, a cada execução de fluxo de trabalho.

Depois que os tipos de fluxo de trabalho e atividade forem registrados, inicie o fluxo de trabalho chamando a ação `StartWorkflowExecution`. O valor do parâmetro `input` pode ser qualquer string especificada pelo aplicativo que está iniciando o fluxo de trabalho. O `executionStartToCloseTimeout` é o tempo em segundos que a execução de fluxo de trabalho pode consumir desde o início até o encerramento. Exceder esse limite fará com que a execução de fluxo de trabalho atinja o tempo limite. Ao contrário de alguns dos outros parâmetros de tempo limite no Amazon SWF, você não pode especificar um valor `NONE` para esse tempo limite; há um limite máximo de um ano no tempo em que uma execução de fluxo de trabalho pode ser executada. Da mesma forma, taskStartToCloseTimeouté o tempo em segundos que uma tarefa de decisão associada à execução desse fluxo de trabalho pode levar antes de atingir o tempo limite.

```
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"
}
```

Se a ação `StartWorkflowExecution` for bem-sucedida, o Amazon SWF retornará o `runId` para a execução do fluxo de trabalho. O `runId` para uma execução de fluxo de trabalho é exclusivo dentro de uma região específica. Salve o `runId` caso precise especificar posteriormente a execução desse fluxo de trabalho em uma chamada para o Amazon SWF. Por exemplo, você usaria o `runId` se, mais tarde, precisasse enviar um sinal à execução de fluxo de trabalho.

```
{"runId": "9ba33198-4b18-4792-9c15-7181fb3a8852"}
```

Definindo a prioridade da tarefa no Amazon SWF

Por padrão, as tarefas em uma lista de tarefas são entregues de acordo com sua hora de chegada: as primeiras tarefas a serem agendadas são geralmente executadas antes, na medida do possível. Ao definir uma prioridade de tarefa opcional, você pode dar prioridade a determinadas tarefas: O Amazon SWF tentará entregar as tarefas de maior prioridade em uma lista de tarefas antes daquelas com menor prioridade.

> **Note**
>
> As tarefas agendadas primeiro geralmente são executadas antes, mas isso não é garantido.

Você pode definir prioridades de tarefas para fluxos de trabalho e atividades. A prioridade de tarefas de um fluxo de trabalho não afeta a prioridade de nenhuma das tarefas de atividades que ele agenda, nem afeta nenhum dos fluxos de trabalho secundários que ele inicia. A prioridade padrão de uma atividade ou fluxo de trabalho é definida (por você ou pelo Amazon SWF) durante o registro, e a prioridade da tarefa registrada é sempre usada, a menos que seja substituída durante o agendamento da atividade ou o início da execução de um fluxo de trabalho.

Valores de prioridade de tarefas podem variar de "-2147483648" até "2147483647", com números mais altos indicando maior prioridade. Se você não definir a prioridade de tarefas para uma atividade ou um fluxo de trabalho, este(a) último(a) receberá uma prioridade de zero ("0").

Tópicos

- Definindo a prioridade de tarefas para fluxos de trabalho
- Definindo a prioridade de tarefas para atividades
- Ações que retornam informações de prioridade de tarefa

Definindo a prioridade de tarefas para fluxos de trabalho

Você pode definir a prioridade de tarefa para um fluxo de trabalho ao registrá-lo ou iniciá-lo. A prioridade de tarefa definida quando o tipo de fluxo de trabalho é registrado é utilizada como padrão para qualquer execução de fluxo de trabalho desse tipo, a menos que ela seja substituída no momento de iniciar a execução de fluxo de trabalho.

Para registrar um tipo de fluxo de trabalho com uma prioridade de tarefa padrão, defina a defaultTaskPriorityopção ao usar a RegisterWorkflowTypeação:

```
{
  "domain": "867530901",
  "name": "expeditedOrderWorkflow",
  "version": "1.0",
  "description": "Expedited customer orders workflow",
  "defaultTaskStartToCloseTimeout": "600",
  "defaultExecutionStartToCloseTimeout": "3600",
  "defaultTaskList": {"name": "mainTaskList"},
  "defaultTaskPriority": "10",
  "defaultChildPolicy": "TERMINATE"
}
```

Você pode substituir a prioridade de tarefa registrada de um tipo de fluxo de trabalho ao iniciar a execução de um fluxo de trabalho com 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"},
}
```

Você também pode substituir a prioridade da tarefa registrada ao iniciar um fluxo de trabalho secundário ou ao continuar um fluxo de trabalho como novo, como ao responder a uma decisão com. RespondDecisionTaskCompleted

Para definir a prioridade de tarefa de um fluxo de trabalho secundário, forneça o valor em `startChildWorkflowExecutionDecisionAttributes`:

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAA...",
  "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"
        }
      }
    }
  ]
}
```

Ao continuar um fluxo de trabalho como novo, defina a prioridade de tarefa em `continueAsNewWorkflowExecutionDecisionAttributes`:

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAA...",
  "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"
      }
```

```
    }
  ]
}
```

Definindo a prioridade de tarefas para atividades

Você pode definir a prioridade de tarefa para uma atividade ao registrá-la ou agendá-la. A prioridade de tarefa definida ao registrar um tipo de atividade é usada como a prioridade padrão quando essa atividade é executada, a menos que ela seja substituída no momento de agendar a atividade.

Para definir a prioridade da tarefa ao registrar um tipo de atividade, defina a defaultTaskPriorityopção ao usar a RegisterActivityTypeação:

```
{
  "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 agendar uma tarefa com prioridade de tarefa, use a opção taskPriority ao agendar a atividade com a ação: RespondDecisionTaskCompleted

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "ScheduleActivityTask",
      "scheduleActivityTaskDecisionAttributes": {
        "activityId": "verify-account",
        "activityType": {
            "name": "activityVerify",
            "version": "1.0"
        },
        "control": "digital music",
```

```
        "input": "abab-101",
        "taskList": {"name": "mainTaskList"},
        "taskPriority": "15"
      }
    }
  ]
}
```

Ações que retornam informações de prioridade de tarefa

Você pode obter informações sobre a prioridade da tarefa definida (ou definir a prioridade da tarefa padrão) nas seguintes ações do Amazon SWF:

- DescribeActivityTyperetorna o defaultTaskPrioritydo tipo de atividade na `configuration` seção da resposta.
- DescribeWorkflowExecutionretorna o TaskPriority da execução do fluxo de trabalho na seção `executionConfiguration` da resposta.
- DescribeWorkflowTyperetorna o defaultTaskPrioritydo tipo de fluxo de trabalho na `configuration` seção da resposta.
- GetWorkflowExecutionHistorye PollForDecisionTaskforneça informações sobre a prioridade da tarefa `workflowExecutionStartedEventAttributes` nas seções `activityTaskScheduledEventAttributes` `decisionTaskScheduledEventAttributesworkflowExecutionContinuedAsNewEventAttribu` e da resposta.

Tratamento de erros no Amazon SWF

Existem vários tipos de erros que podem ocorrer durante a execução de um fluxo de trabalho.

Tópicos

- Erros de validação
- Erros na execução de ações ou decisões
- Tempos limite
- Erros gerados pelo código do usuário
- Erros relacionados ao encerramento de uma execução de fluxo de trabalho

Erros de validação

Os erros de validação ocorrem quando uma solicitação para o Amazon SWF falha porque não está formada corretamente ou contém dados inválidos. Nesse contexto, uma solicitação poderia ser uma ação, como `DescribeDomain`, ou uma decisão, como `StartTimer`. Se a solicitação for uma ação, o Amazon SWF retornará um código de erro na resposta. Verifique esse código de erro, pois ele pode fornecer informações sobre qual aspecto da solicitação causou a falha. Por exemplo, um ou mais dos argumentos transmitidos com a solicitação podem ser inválidos. Para obter uma lista de códigos de erro comuns, acesse o tópico da ação em Amazon Simple Workflow Service API Reference.

Se a solicitação que falhou for uma decisão, um evento apropriado será listado no histórico de execução de fluxo de trabalho. Por exemplo, se a decisão `StartTimer` falhasse, você veria um evento `StartTimerFailed` no histórico. O agente de decisão deve verificar esses eventos ao receber o histórico em resposta a `PollForDecisionTask` ou `GetWorkflowExecutionHistory`. Veja a seguir uma lista de possíveis eventos de falha de decisão que podem ocorrer quando a decisão não está formada corretamente ou contém dados inválidos.

Erros na execução de ações ou decisões

Mesmo que a solicitação seja formada corretamente, podem ocorrer erros quando o Amazon SWF tentar executar a solicitação. Nesses casos, um dos seguintes eventos no histórico indicará que ocorreu um erro. Observe o campo `reason` do evento para determinar a causa da falha.

- `CancelTimerFailed`
- `RequestCancelActivityTaskFailed`
- `RequestCancelExternalWorkflowExecutionFailed`
- `ScheduleActivityTaskFailed`
- `SignalExternalWorkflowExecutionFailed`
- `StartChildWorkflowExecutionFailed`
- `StartTimerFailed`

Tempos limite

Agentes de decisão, operadores de atividade e execuções de fluxo de trabalho operam todos dentro das restrições de períodos de tempo limite. Nesse tipo de erro, uma tarefa ou um fluxo de trabalho

secundário atinge o tempo limite. Um evento aparecerá no histórico que descreve o tempo limite. O agente de decisão deve lidar com esse evento, por exemplo, reagendando a tarefa ou reiniciando o fluxo de trabalho secundário. Para obter mais informações sobre tempos limite, consulte Tipos de tempo limite do Amazon SWF

- `ActivityTaskTimedOut`
- `ChildWorkflowExecutionTimedOut`
- `DecisionTaskTimedOut`
- `WorkflowExecutionTimedOut`

Erros gerados pelo código do usuário

Exemplos desse tipo de condição de erro são falhas de tarefa de atividade e falhas de fluxos de trabalho secundários. Assim como acontece com os erros de tempo limite, o Amazon SWF adiciona um evento apropriado ao histórico de execução do fluxo de trabalho. O agente de decisão deve lidar com esse evento, possivelmente reagendando a tarefa ou reiniciando o fluxo de trabalho secundário.

- `ActivityTaskFailed`
- `ChildWorkflowExecutionFailed`

Erros relacionados ao encerramento de uma execução de fluxo de trabalho

Agentes de decisão também podem ver os seguintes eventos se tentarem encerrar um fluxo de trabalho que tenha uma tarefa de decisão pendente.

- `FailWorkflowExecutionFailed`
- `CompleteWorkFlowExecutionFailed`
- `ContinueAsNewWorkflowExecutionFailed`
- `CancelWorkflowExecutionFailed`

Para obter mais informações sobre qualquer um dos eventos listados acima, consulte Evento de histórico na Referência da API do Amazon SWF.

Cotas do Amazon SWF

O Amazon SWF coloca cotas nos tamanhos de determinados parâmetros de fluxo de trabalho, como no número de domínios por conta e no tamanho do histórico de execução do fluxo de trabalho. Essas cotas são projetadas para evitar que fluxos de trabalho errôneos consumam todos os recursos do sistema, mas não são limites rígidos. Se você achar que sua aplicação está excedendo essas cotas com frequência, poderá solicitar um aumento da cota de serviço.

Conteúdo

- Cotas gerais de contas para o Amazon SWF
- Cotas na execução do fluxo de trabalho
- Cotas em execuções de tarefas
- Cotas de controle de utilização do Amazon SWF
  - Controle de utilização de cotas para todas as regiões
  - cotas de decisão para todas as regiões
  - Cotas em nível de fluxo de trabalho
- Solicitar um aumento de cota

Cotas gerais de contas para o Amazon SWF

- Máximo de domínios registrados: 100

  Essa cota inclui domínios registrados e obsoletos.
- Máximo de tipos de atividade e fluxo de trabalho: 10.000 por domínio

  Essa cota inclui tipos registrados e obsoletos.
- Cota de chamadas de API: além de picos infrequentes, as aplicações podem ter controle de utilização se fizerem um grande número de chamadas de API em um período muito curto.
- Tamanho máximo da solicitação: 1 MB por solicitação

  Esse é o tamanho total dos dados por solicitação da API do Amazon SWF, incluindo o cabeçalho da solicitação e todos os outros dados de solicitação associados.
- Respostas truncadas para contagem APIs — indica que uma cota interna foi atingida e que a resposta não é a contagem completa.

Algumas consultas atingirão internamente a cota de 1 MB mencionada acima antes de retornar uma resposta completa. As APIs a seguir podem retornar uma resposta truncada, em vez de uma contagem total.

- CountClosedWorkflowExecutions
- CountOpenWorkflowExecutions
- CountPendingActivityTasks
- CountPendingDecisionTasks

Para cada uma dessas, se a resposta `truncated` for definida como verdadeira, a contagem será menor que a quantidade total. Essa cota interna não pode ser aumentada.

- Número máximo de tags: 50 tags por recurso.

  A tentativa de adicionar mais de 50 tags resultará em um erro 400 `TooManyTagsFault`.

Cotas na execução do fluxo de trabalho

- Máximo de execuções de fluxo de trabalho abertas: 100.000 por domínio

  Essa contagem inclui execuções de fluxo de trabalho secundárias.
- Tempo máximo de execução do fluxo de trabalho: 1 ano. Essa é uma cota rígida que não pode ser alterada.
- Tamanho máximo do histórico de execução do fluxo de trabalho: 25.000 eventos. Essa é uma cota rígida que não pode ser alterada.

  A melhor prática é estruturar cada fluxo de trabalho de tal forma que seu histórico não ultrapasse 10.000 eventos. Como o decisor precisa buscar o histórico do fluxo de trabalho, um histórico menor permite que o decisor conclua mais rapidamente. Se estiver usando o Flow Framework, você pode usá-lo ContinueAsNew para continuar um fluxo de trabalho com um novo histórico.
- Máximo de execuções de fluxo de trabalho secundárias abertas:1.000 por execução de fluxo de trabalho

Se o seu caso de uso exigir que você vá além dessas cotas, você poderá usar os recursos que o Amazon SWF oferece para continuar as execuções e estruturar suas aplicações usando execuções de fluxo de trabalho filho. Se você achar que ainda precisa de um aumento de cota, consulte Solicitar um aumento de cota.

Cotas em execuções de tarefas

- Máximo de sondagens por lista de tarefas: 1.000 por lista de tarefas

  Você pode ter um máximo de 1.000 agentes de sondagem sondando simultaneamente uma lista de tarefas específica. Se você ultrapassar 1.000, receberá um `LimitExceededException`.

  > **Note**
  >
  > Embora o máximo seja 1.000, você pode encontrar erros de `LimitExceededException` bem antes dessa cota. Esse erro não significa que suas tarefas estejam atrasadas. Em vez disso, isso significa que você tem a quantidade máxima de pesquisas inativas em uma lista de tarefas. O Amazon SWF define esse limite para economizar recursos no lado do cliente e do servidor. Definir o limite evita que um número excessivo de pesquisadores espere desnecessariamente. Você pode reduzir os erros `LimitExceededException` usando várias listas de tarefas para distribuir a sondagem.

- Máximo de tarefas agendadas por segundo: 2.000 por lista de tarefas

  Você pode agendar no máximo 2.000 tarefas por segundo em uma lista de tarefas específica. Se você exceder 2.000, suas decisões `ScheduleActivityTask` falharão com erro `ACTIVITY_CREATION_RATE_EXCEEDED`.

  > **Note**
  >
  > Embora o máximo seja 2.000, você pode encontrar erros `ACTIVITY_CREATION_RATE_EXCEEDED` bem antes dessa cota. Para reduzir esses erros, use várias listas de tarefas para distribuir a carga.

- Tempo máximo de execução da tarefa: 1 ano (limitado pelo tempo máximo de execução do fluxo de trabalho)

  Você pode configurar tempos limite de atividade para fazer com que um evento de tempo limite ocorra se um estágio específico da execução da tarefa de atividade demorar muito.

- Tempo máximo em que o SWF manterá uma tarefa na fila: 1 ano (limitado pela cota de tempo de execução do fluxo de trabalho)

Você pode configurar tempos limite de atividade padrão durante o registro de uma atividade, o que fará com que um evento de tempo limite ocorra se um estágio específico da execução da tarefa de atividade demorar muito. Você também pode substituir os tempos limites de atividade padrão ao agendar uma tarefa de atividade no código do agente de decisão.

- Máximo de tarefas de atividade abertas: 1.000 por execução de fluxo de trabalho.

  Essa cota inclui tanto as tarefas de atividade que foram agendadas quanto as que estão sendo processadas pelos funcionários.

- Máximo de temporizadores abertos: 1.000 por execução de fluxo de trabalho
- Tamanho máximo dos dados de entrada/resultado: 32.768 caracteres

  Essa cota afeta os dados de resultado da execução da atividade ou do fluxo de trabalho, os dados de entrada ao programar tarefas de atividade ou execuções de fluxo de trabalho e a entrada enviada com um sinal de execução de fluxo de trabalho.

- Decisões máximas em uma resposta de tarefa de decisão: varia

  Devido à cota de 1 MB no tamanho máximo da solicitação de API, o número de decisões retornadas em uma única chamada para `RespondDecisionTaskCompleted` será limitado de acordo com o tamanho dos dados usados por cada decisão, incluindo o tamanho de quaisquer dados de entrada fornecidos para tarefas de atividade agendadas ou para execuções de fluxo de trabalho.

Cotas de controle de utilização do Amazon SWF

Além das cotas de serviço descritas anteriormente, determinadas chamadas de API do Amazon SWF e eventos de decisão têm controle de utilização para manter a largura de banda do serviço, usando um esquema de token de bucket. Se a sua taxa de solicitações exceder consistentemente as taxas listadas aqui, você poderá solicitar um aumento da cota do acelerador.

As cotas de controle de utilização e decisão são as mesmas em todas as regiões.

Controle de utilização de cotas para todas as regiões

As cotas a seguir são aplicáveis em níveis de contas individuais. Não é possível solicitar um aumento das seguintes cotas. Para obter informações sobre como fazer isso, consulte Solicitar um aumento de cota.

Nome da API	Tamanho do bucket	Taxa de reabastecimento 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
`ListClosedWorkflowExecutions`	200	6
`ListDomains`	100	6
`ListOpenWorkflowExecutions`	200	48
`ListTagsForResource`	50	30

Nome da API	Tamanho do bucket	Taxa de reabastecimento por segundo
`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
`TerminateWorkflowExecution`	2000	60
`UndeprecateActivityType`	200	6
`UndeprecateDomain`	100	6
`UndeprecateWorkflowType`	200	6
`UntagResource`	50	30

cotas de decisão para todas as regiões

As cotas a seguir são aplicáveis em níveis de contas individuais. Não é possível solicitar um aumento das seguintes cotas. Para obter informações sobre como fazer isso, consulte Solicitar um aumento de cota.

Nome da API	Tamanho do bucket	Taxa de reabastecimento por segundo
`RequestCancelExternalWorkflowExecution`	1200	120
`ScheduleActivityTask`	1000	200
`SignalExternalWorkflowExecution`	1200	120
`StartChildWorkflowExecution`	500	12
`StartTimer`	2000	200

Cotas em nível de fluxo de trabalho

As cotas a seguir são aplicáveis em níveis de fluxo de trabalho e não podem ser aumentadas.

Nome da API	Tamanho do bucket	Taxa de reabastecimento por segundo
`GetWorkflowExecutionHistory`	400	200
`SignalWorkflowExecution`	1000	1000
`RecordActivityTaskHeartbeat`	1000	1000
`RequestCancelWorkflowExecution`	200	200

Solicitar um aumento de cota

Para obter mais informações, consulte Service Quotas do AWS em Referência geral da AWS.

Recursos adicionais e informações de referência para o Amazon SWF

Este capítulo fornece recursos adicionais e informações de referência que são úteis ao desenvolver fluxos de trabalho com o Amazon SWF.

Tópicos

- Tipos de tempo limite do Amazon SWF
- Endpoints do Amazon Simple Workflow Service
- Documentação adicional sobre o Amazon Simple Workflow Service
- Recursos da web para o Amazon Simple Workflow Service
- Opções de migração para Ruby Flow

Tipos de tempo limite do Amazon SWF

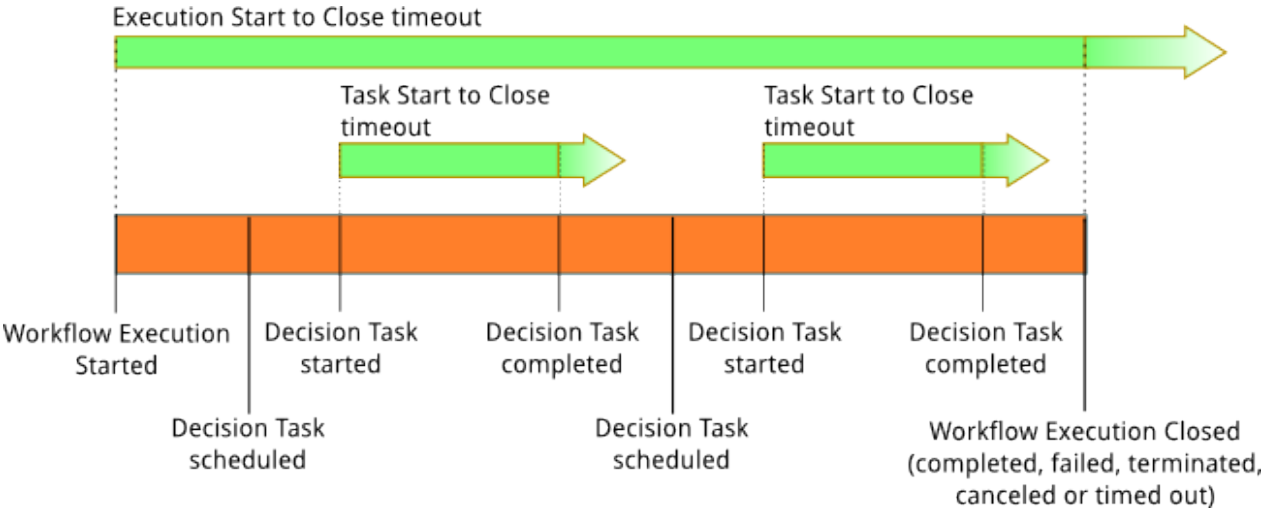
Para garantir que as execuções do fluxo de trabalho sejam executadas corretamente, você pode definir diferentes tipos de tempos limite com o Amazon SWF. Alguns tempos limite especificam por quanto tempo o fluxo de trabalho pode ser executado na sua totalidade. Outros tempos limites especificam quanto tempo atividades de tarefa podem demorar antes de serem atribuídas a um operador e quanto tempo elas podem levar para serem concluídas a partir do momento em que são agendadas. Todos os tempos limite na API do Amazon SWF são especificados em segundos. O Amazon SWF também aceita a string `NONE` como um valor de tempo limite, o que indica que não há tempo limite.

Para tempos limite relacionados a tarefas de decisão e tarefas de atividade, o Amazon SWF adiciona um evento ao histórico de execução do fluxo de trabalho. Os atributos do evento fornecem informações sobre o tipo de tempo limite ocorrido e qual tarefa de decisão ou tarefa de atividade foi afetada. O Amazon SWF também agenda uma tarefa de decisão. Quando o tomador de decisão receber a nova tarefa de decisão, ele verá o evento de tempo limite no histórico e tomará uma ação apropriada chamando a RespondDecisionTaskCompletedação.

Uma tarefa é considerada aberta a partir do momento em que ela é agendada até ser encerrada. Portanto, uma tarefa é informada como aberta enquanto um operador a processa. Uma tarefa é encerrada quando um operador a informa como concluída, cancelada ou falha. Uma tarefa também pode ser fechada pelo Amazon SWF como resultado de um tempo limite.

Tempos limites em tarefas de fluxo de trabalho e decisão

O diagrama a seguir mostra como tempos limites de fluxo de trabalho e decisão estão relacionado ao ciclo de vida de um fluxo de trabalho:



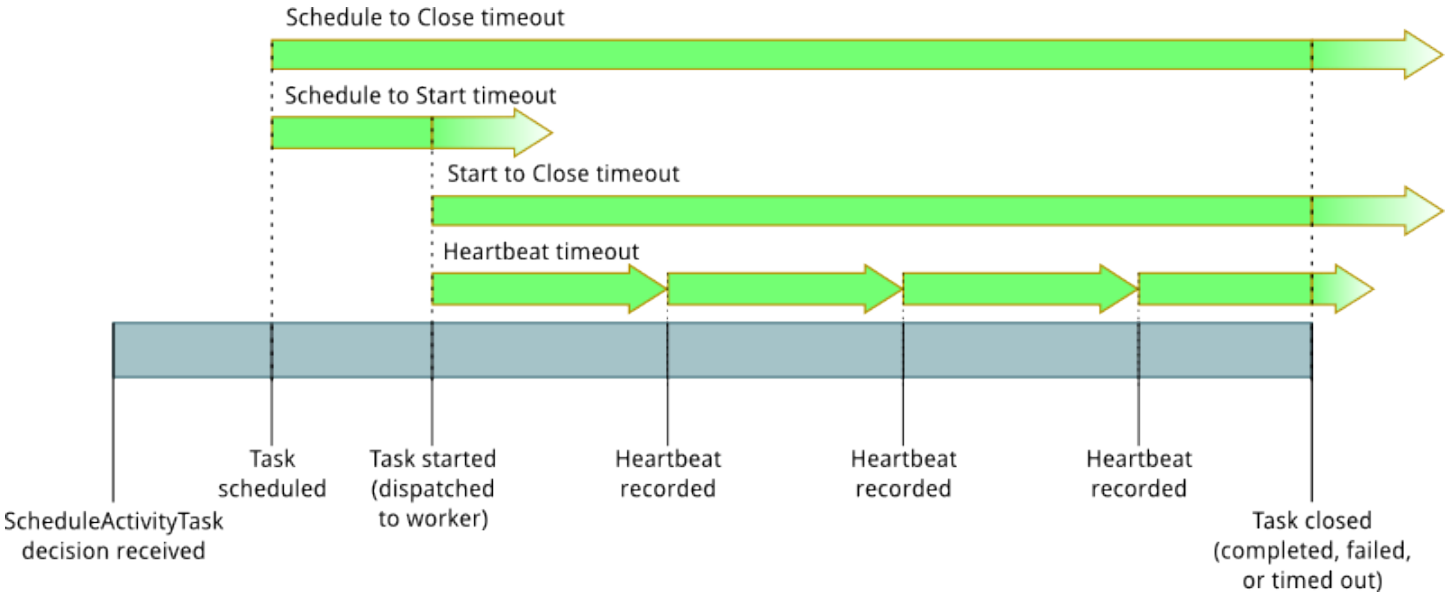
Existem dois tipos de tempo limite que são relevantes para tarefas de decisão e fluxo de trabalho:

- Início do fluxo de trabalho até o encerramento (**`timeoutType: START_TO_CLOSE`**): esse tempo limite especifica o tempo máximo que uma execução de fluxo de trabalho pode levar para ser concluída. Ele é definido como um padrão durante o registro do fluxo de trabalho, mas pode ser substituído por um valor diferente quando o fluxo de trabalho é iniciado. Se esse tempo limite for excedido, o Amazon SWF fecha a execução do fluxo de trabalho e adiciona um evento do WorkflowExecutionTimedOuttipo ao histórico de execução do fluxo de trabalho. Além do `timeoutType`, os atributos de evento especificam o `childPolicy` que está em vigor para essa execução de fluxo de trabalho. A política de elementos secundários especifica como as execuções de fluxo de trabalho secundárias serão tratadas se a execução de fluxo de trabalho principal atingir o tempo limite ou for encerrada de outro modo. Por exemplo `childPolicy`, se ela estiver definida como TERMINATE, as execuções de fluxo de trabalho secundárias serão finalizadas. Quando uma execução de fluxo de trabalho tiver atingido o tempo limite, não será possível realizar nenhuma ação nela além de chamadas de visibilidade.
- Início da tarefa de decisão até o encerramento (**`timeoutType: START_TO_CLOSE`**): esse tempo limite especifica o tempo máximo que o agente de decisão correspondente pode levar para concluir uma tarefa de decisão. Ele é definido durante o registro do tipo de fluxo de trabalho. Se esse tempo limite for excedido, a tarefa será marcada como expirada no histórico de execução do fluxo de trabalho e o Amazon SWF adicionará um evento do tipo DecisionTaskTimedOutao histórico do fluxo de trabalho. Os atributos do evento incluirão IDs os eventos que correspondem

a quando essa tarefa de decisão foi agendada (`scheduledEventId`) e quando foi iniciada (`startedEventId`). Além de adicionar o evento, o Amazon SWF também agenda uma nova tarefa de decisão para alertar o agente de decisão de que essa tarefa de decisão atingiu o tempo limite. Após esse tempo limite, uma tentativa de concluir a tarefa de decisão expirada usando `RespondDecisionTaskCompleted` falhará.

Tempos limites em tarefas de atividade

O diagrama a seguir mostra como tempos limite estão relacionados ao ciclo de vida de uma tarefa de atividade:



Existem quatro tipos de tempo limite que são relevantes para tarefas de atividade:

- Início da tarefa de atividade até o encerramento (**`timeoutType: START_TO_CLOSE`**): esse tempo limite especifica o tempo máximo que um operador de atividade pode levar para processar uma tarefa depois que ele recebeu a tarefa. As tentativas de fechar uma tarefa de atividade com tempo limite expirado usando RespondActivityTaskCanceled, RespondActivityTaskCompleted, RespondActivityTaskFailedfalharão.
- Heartbeat de tarefa de atividade (**`timeoutType: HEARTBEAT`**): esse tempo limite especifica o tempo máximo que uma tarefa pode ser executada antes de fornecer seu andamento por meio da ação `RecordActivityTaskHeartbeat`.
- Programação de início da tarefa de atividade(**`timeoutType: SCHEDULE_TO_START`**): esse tempo limite especifica quanto tempo o Amazon SWF aguarda antes de encerrar a tarefa de

atividade se não houver operadores disponíveis para executar a tarefa. Após o tempo limite, a tarefa expirada não será atribuída a outro operador.

- Programação de encerramento da tarefa de atividade (**`timeoutType: SCHEDULE_TO_CLOSE`**): esse tempo limite especifica quanto tempo a tarefa pode levar desde o momento em que é agendada até o momento em que é concluída. Como prática recomendada, esse valor não deve ser maior que a soma do tempo limite da tarefa e do schedule-to-start tempo limite da tarefa start-to-close.

> **Note**
>
> Cada um dos tipos de tempo limite tem um valor padrão, que geralmente está definido como `NONE` (infinito). No entanto, o tempo máximo para qualquer execução de atividade é limitado a um ano.

Você define valores padrão para eles durante o registro do tipo de atividade, mas pode substituí-los por novos valores ao agendar a tarefa de atividade. Quando um desses tempos limite ocorrer, o Amazon SWF adicionará um evento do ActivityTaskTimedOuttipo ao histórico do fluxo de trabalho. O atributo de valor `timeoutType` desse evento especificará quais desses tempos limites ocorreu. Para cada um dos tempos limites, o valor de `timeoutType` é mostrado entre parênteses. Os atributos do evento também incluirão IDs os eventos que correspondem a quando a tarefa da atividade foi agendada (`scheduledEventId`) e quando foi iniciada (`startedEventId`). Além de adicionar o evento, o Amazon SWF também agenda uma nova tarefa de decisão para alertar o agente de decisão de que o tempo limite ocorreu.

Endpoints do Amazon Simple Workflow Service

Uma lista das atuais regiões e endpoints do Amazon SWF é fornecida na Referência geral da Amazon Web Services, com os endpoints de outros serviços.

Os domínios do Amazon SWF e todos os fluxos de trabalho e atividades relacionados devem existir na mesma região para se comunicarem entre si. Além disso, domínios, fluxos de trabalho e atividades registrados dentro de uma região não existem em outras regiões. Por exemplo, se você criar um domínio chamado "MySampleDomain" em us-east-1 e em us-west-2, eles existirão como domínios separados: nenhum dos fluxos de trabalho, listas de tarefas, atividades ou dados associados aos seus domínios é compartilhado entre regiões.

Se você usa outros AWS recursos em seus fluxos de trabalho, como EC2 instâncias da Amazon, eles também devem existir na mesma região que seus recursos do Amazon SWF. As únicas exceções a isso são os serviços que abrangem regiões, como o Amazon S3 e o IAM. É possível acessar esses serviços a partir de fluxos de trabalho que existem em qualquer região com suporte.

Documentação adicional sobre o Amazon Simple Workflow Service

Além deste Guia do Desenvolvedor, a seguinte documentação útil pode ser útil para você.

Amazon Simple Workflow Service API Reference

A Amazon Simple Workflow Service API Reference fornece informações detalhadas sobre a API HTTP do Amazon SWF, incluindo ações, estruturas de solicitação e resposta e códigos de erro.

AWS Flow Framework Documentação

O AWS Flow Framework é um framework de programação que simplifica o processo de implementação de aplicações assíncronas distribuídas que usam o Amazon SWF para gerenciar fluxos de trabalho e atividades, para que você possa se concentrar na implementação da lógica do fluxo de trabalho.

Cada um foi AWS Flow Framework projetado para funcionar idiomaticamente no idioma para o qual foi projetado, para que você possa trabalhar naturalmente com a linguagem de sua escolha para implementar fluxos de trabalho com todos os benefícios do Amazon SWF.

Existe uma estrutura AWS de fluxo para Java. O Guia do Desenvolvedor AWS Flow Framework for Java fornece informações sobre como obter, configurar e usar o AWS Flow Framework for Java.

AWS Documentação do SDK

Os kits AWS de desenvolvimento de software (SDKs) fornecem acesso ao Amazon SWF em várias linguagens de programação diferentes. Eles SDKs acompanham de perto a API HTTP, mas também fornecem interfaces de programação específicas de linguagem para alguns recursos do Amazon SWF. Você pode encontrar mais informações sobre cada SDK visitando os seguintes links.

> **Note**
>
> Somente aqueles SDKs que têm suporte para o Amazon SWF no momento em que este artigo foi escrito estão listados aqui. Para obter uma lista completa dos disponíveis AWS SDKs, visite a página Tools for Amazon Web Services.

Java

O AWS SDK para Java fornece uma API Java para serviços de AWS infraestrutura.

Para visualizar a documentação disponível, consulte a página de Documentação do AWS SDK para Java. Você também pode ir diretamente para as seções do Amazon SWF na referência do SDK seguindo estes links:

- `Class: AmazonSimpleWorkflowClient`
- `Class: AmazonSimpleWorkflowAsyncClient`
- `Interface: AmazonSimpleWorkflow`
- `Interface: AmazonSimpleWorkflowAsync`

JavaScript

AWS SDK para JavaScript Isso permite que os desenvolvedores criem bibliotecas ou aplicativos que fazem uso de AWS serviços usando uma easy-to-use API simples e disponível no navegador ou dentro dos aplicativos Node.js no servidor.

Para visualizar a documentação disponível, consulte a página de Documentação do AWS SDK para JavaScript. Você também pode ir diretamente para a seção Amazon SWF na referência do SDK, seguindo este link:

- `Class: AWS.SimpleWorkflow`

.NET

AWS SDK para .NET É um pacote único e disponível para download que inclui modelos de projeto do Visual Studio, a AWS biblioteca.NET, exemplos de código C# e documentação. AWS SDK para .NET Isso torna mais fácil para os desenvolvedores do Windows criarem aplicativos.NET para o Amazon SWF e outros serviços.

Para visualizar a documentação disponível, consulte a página de Documentação do AWS SDK para .NET. Você também pode ir diretamente para as seções do Amazon SWF na referência do SDK seguindo estes links:

- `Namespace: Amazon.SimpleWorkflow`
- `Namespace: Amazon.SimpleWorkflow.Model`

PHP

O AWS SDK for PHP fornece uma interface de programação PHP para o Amazon SWF.

Para visualizar a documentação disponível, consulte a página de Documentação do AWS SDK for PHP. Você também pode ir diretamente para a seção Amazon SWF na referência do SDK, seguindo este link:

- `Class: SwfClient`

Python

O AWS SDK for Python (Boto) fornece uma interface de programação em Python para o Amazon SWF.

Para ver a documentação disponível, consulte boto: Uma interface do Python para a Amazon Web Services. Você também pode ir diretamente para as seções do Amazon SWF na documentação seguindo estes links:

- Tutorial do Amazon SWF
- Referência do Amazon SWF

Ruby

O AWS SDK para Ruby fornece uma interface de programação Ruby para o Amazon SWF.

Para visualizar a documentação disponível, consulte a página de Documentação do AWS SDK para Ruby. Você também pode ir diretamente para a seção Amazon SWF na referência do SDK, seguindo este link:

- Classe: AWS::Simple Fluxo de trabalho

AWS CLI Documentação

O AWS Command Line Interface (AWS CLI) é uma ferramenta unificada para gerenciar seus AWS serviços. Com apenas uma ferramenta para baixar e configurar, você pode controlar vários AWS serviços a partir da linha de comando e automatizá-los por meio de scripts.

Para obter mais informações sobre o AWS CLI, consulte a AWS Command Line Interfacepágina.

Para obter uma visão geral dos comandos disponíveis para o Amazon SWF, consulte swf na Referência de comandos da AWS CLI .

Recursos da web para o Amazon Simple Workflow Service

Existe vários recursos da web que você pode usar para saber mais sobre o Amazon SWF ou para obter ajuda para usar o serviço e desenvolver fluxos de trabalho.

Fórum do Amazon SWF

O fórum do Amazon SWF oferece um local para você se comunicar com outros desenvolvedores do Amazon SWF e membros da equipe de desenvolvimento do Amazon SWF na Amazon para fazer perguntas e obter respostas.

Você pode visitar o fórum em: Fórum: Serviço de fluxo de trabalho simples da Amazon.

Perguntas frequentes sobre o Amazon SWF

As perguntas frequentes sobre o Amazon SWF fornecem respostas a perguntas frequentes sobre o Amazon SWF, incluindo uma visão geral de casos de uso comuns, diferenças entre o Amazon SWF e outros serviços e muito mais.

Você pode acessar as perguntas frequentes aqui: Perguntas frequentes sobre o Amazon SWF.

Vídeos Amazon SWF

O canal Amazon Web Services YouTube oferece treinamento em vídeo para todos os serviços web da Amazon, incluindo o Amazon SWF. Para obter uma lista completa de vídeos relacionados ao Amazon SWF, use a seguinte consulta: Fluxo de trabalho simples na Amazon Web Services

Opções de migração para Ruby Flow

O AWS Flow Framework for Ruby não está mais em desenvolvimento ativo. Embora o código existente continue funcionando indefinidamente, não haverá novos recursos nem versões. Este tópico abordará as opções de uso e migração para continuar trabalhando com o Amazon SWF, além de informações sobre como migrar para o Step Functions.

Opção	Descrição
Continuar usando o Ruby Flow Framework	Por ora, o Ruby Flow Framework continuará funcionando. Se você não fizer nada, o código continuará funcionando como está. Planeje migrar do AWS Flow Framework para Ruby em um futuro próximo.
Migrar para o Java Flow Framework	O Java Flow Framework permanece em desenvolvimento ativo e continuará recebendo recursos e atualizações novos.
Migrar para o Step Functions	O Step Functions oferece uma maneira de coordenar os componentes de aplicativos distribuídos usando fluxos de trabalho visuais controlad os por uma máquina de estado.
Usar a API do SWF diretamente, sem o Flow Framework	Você pode continuar trabalhando no Ruby e usar a API do SWF diretamente em vez do Ruby Flow Framework.

A vantagem que o Flow Framework oferece, para o Ruby ou o Java, é que permite a você se concentrar na lógica do fluxo de trabalho. A estrutura processa muitos dos detalhes da comunicação e da coordenação, e parte da complexidade é abstraída. Você pode continuar a ter o mesmo nível de abstração migrando para o Java Flow Framework ou pode interagir diretamente com o SDK do Amazon SWF.

Continuar usando o Ruby Flow Framework

O AWS Flow Framework for Ruby continuará funcionando como agora no curto prazo. Se você tiver fluxos de trabalho escritos em AWS Flow Framework para Ruby, eles continuarão funcionando. Sem atualizações, suporte ou correções de segurança, é melhor ter um plano firme para migrar do AWS Flow Framework para o Ruby em um futuro próximo.

Migrar para o Java Flow Framework

O AWS Flow Framework for Java permanecerá em desenvolvimento ativo. Conceitualmente, o AWS Flow Framework para Java é semelhante ao AWS Flow Framework para Ruby: você ainda pode se concentrar na lógica do fluxo de trabalho, e a estrutura ajudará a gerenciar sua lógica de decisão e facilitará o gerenciamento de outros aspectos do Amazon SWF.

- AWS Flow Framework para Java

- AWS Flow Framework para referência da API Java

Migrar para o Step Functions

AWS Step Functions fornece um serviço semelhante ao Amazon SWF, mas em que a lógica do fluxo de trabalho é controlada por uma máquina de estado. O Step Functions permite que você coordene os componentes de aplicações e microsserviços distribuídos usando fluxos de trabalho visuais. Os aplicativos são desenvolvidos a partir de componentes individuais que executam uma função ou uma tarefa distinta e permitem dimensionar e alterar rapidamente os aplicativos. O Step Functions oferece uma maneira confiável de coordenar componentes e percorrer as funções da aplicação. Um console gráfico oferece uma maneira de visualizar os componentes do aplicativo como uma série de etapas. Ele aciona e rastreia e monitora automaticamente todas as etapas e tenta executar novamente etapas que apresentaram falha para que a aplicação sempre execute na ordem e da forma esperada. O Step Functions registra o estado de cada etapa, de modo que, quando algo dá errado, é possível diagnosticar e depurar problemas rapidamente.

No Step Functions, você gerencia a coordenação de suas tarefas usando uma máquina de estado, escrita em JSON declarativo, que é definida usando a Amazon States Language. Ao usar uma máquina de estado, você não precisa escrever e manter um programa do agente de decisão para controlar a lógica da aplicação. O Step Functions oferece uma abordagem intuitiva, produtiva e ágil para coordenar componentes de aplicações usando fluxos de trabalho visuais. Você deve considerar o uso AWS Step Functions para todos os seus novos aplicativos, e o Step Functions fornece uma excelente plataforma para migrar para os fluxos de trabalho que você implementou atualmente no AWS Flow Framework para Ruby.

Para ajudar a migrar suas tarefas para o Step Functions e, ao mesmo tempo, continuar aproveitando suas habilidades na linguagem Ruby, o Step Functions fornece um exemplo de operador de atividade Ruby. Esse exemplo usa as práticas recomendadas para implementar um operador de atividade e pode ser usado como modelo para migrar sua lógica de tarefa para Step Functions. Para obter mais informações, consulte o tópico Example Activity Worker in Ruby no Guia do desenvolvedor AWS Step Functions.

> **Note**
>
> Para muitos clientes, migrar do para o Ruby AWS Flow Framework para Step Functions é a melhor opção. Mas, se você precisar que os sinais interfiram em seus processos ou se

precisar iniciar processos secundários que retornem um resultado para um pai, considere usar a API Amazon SWF diretamente ou migrar para a AWS Flow Framework for Java.

Para obter mais informações sobre AWS Step Functions, consulte:

- AWS Step Functions Guia do desenvolvedor
- AWS Step Functions API Reference
- AWS Step Functions Referência de linha de comando

Use a API do Amazon SWF diretamente

Embora o AWS Flow Framework for Ruby gerencie parte da complexidade do Amazon SWF, você também pode usar a API do Amazon SWF diretamente. Usar a API diretamente permite criar fluxos de trabalho nos quais você tenha controle total sobre a implementação de tarefas e a coordenação delas sem se preocupar com complexidades subjacentes, como acompanhar o progresso e a manter o estado.

- Guia do desenvolvedor do Amazon Simple Workflow Service
- Amazon Simple Workflow Service API Reference

Histórico do documento

A tabela a seguir descreve as alterações importantes na documentação desde a última versão do Guia do Desenvolvedor do Amazon Simple Workflow Service.

Alteração	Descrição	Alterado em
Atualização somente da documentação.	O Amazon SWF agora inclui uma seção sobre notificaç ões AWS do usuário, AWS service (Serviço da AWS) que atua como um local central para suas AWS notificações no. AWS Management Console Para obter mais informações, consulte Usando Notificações de Usuários da AWS com o Amazon Simple Workflow Service.	4 de maio de 2023
Atualizar	O Amazon SWF agora oferece uma nova experiência de console para gerenciar fluxos de trabalho SWF e suas ações relacionadas à execução. Para obter mais informaçõ es, consulte os tutoriais do console do Amazon SWF.	12 de setembro de 2022
Atualizar	A Cotas em execuções de tarefas seção foi atualizada para incluir `Maximum tasks scheduled per second` e a Métricas do Amazon SWF para CloudWatch página para incluir informações sobre o uso de nomes de recursos não ASCII com. CloudWatch	12 de maio de 2021
Novo atributo	O Amazon Simple Workflow Service agora oferece suporte à Amazon EventBridge. Para obter mais informações, consulte: • EventBridge para Amazon SWF • EventBridge Guia do usuário	18 de dezembro de 2020
Novo atributo	O Amazon Simple Workflow Service aceita permissões do IAM usando tags. Para obter mais informações, consulte. • Tags no Amazon SWF • Gerenciar tags	20 de junho de 2019

Alteração	Descrição	Alterado em
	• Marcar execuções de fluxo de trabalho • Controle o acesso aos domínios com tags • `TagResource` • `UntagResource` • `ListTagsForResource` • `RegisterDomain`	
Novo atributo	O Amazon Simple Workflow Service já está disponível na região da Europa (Estocolmo).	12 de dezembro de 2018
Atualizar	O tópico sobre CloudTrail integração do Amazon Simple Workflow Service foi aprimorado. Consulte Gravando chamadas de API com AWS CloudTrail.	7 de agosto de 2018
Atualizar	Foram adicionadas informações sobre a nova `PendingTasks` métrica para CloudWatch. Para obter mais informações, consulte Métricas do Amazon SWF.	18 de junho de 2018
Atualizar	Destaque aprimorado de sintaxe em exemplos de código.	29 de março de 2018
Atualizar	Adicionado um tópico descrevendo opções para usuários do Ruby Flow deixarem a plataforma. Para obter mais informações, consulte Opções de migração para Ruby Flow.	9 de março de 2018
Atualizar	Melhorada a navegação em tópico de conceitos avançados. Consulte Conceitos avançados de fluxo de trabalho no Amazon SWF.	19 de fevereiro de 2018
Atualizar	Documentação de CloudWatch métricas aprimorada com a adição de informações estatísticas válidas. Consulte Métricas do Amazon SWF para CloudWatch.	4 de dezembro de 2017

Alteração	Descrição	Alterado em
Atualizar	Alteração do Índice para melhorar a estrutura do documento. Adição de novas informações em Métricas de eventos de API e decisão.	9 de novembro de 2017
Atualizar	Atualização da seção Cotas do Amazon SWF para incluir limites de controle de fluxo para todas as regiões.	18 de outubro de 2017
Atualizar	Alteração de `task_list` para `workflowId` no Começando a usar o Amazon SWF para evitar confusão com `activity_list` .	25 de julho de 2017
Atualizar	Limpeza dos exemplos de código neste guia.	5 de junho de 2017
Atualizar	Simplificação e melhoria da organização e do conteúdo deste guia.	19 de maio de 2017
Atualizar	Atualizações e correções de links.	16 de maio de 2017
Atualizar	Atualizações e correções de links.	1 de outubro de 2016
Suporte a tarefas do Lambda	Você pode especificar tarefas Lambda além das tarefas de atividade tradicionais em seus fluxos de trabalho. Para obter mais informações, consulte AWS Lambda tarefas no Amazon SWF.	21 de julho de 2015
Suporte para definir a prioridade de tarefas	O Amazon SWF agora inclui suporte para definir a prioridade das tarefas em uma lista de tarefas e tentará entregar aquelas com prioridade mais alta antes das tarefas com prioridade mais baixa. Informações sobre como definir a prioridade de tarefas para fluxos de trabalho e atividades são fornecidas em Definindo a prioridade da tarefa no Amazon SWF.	17 de dezembro de 2014

Alteração	Descrição	Alterado em
Atualizar	Foi adicionado um novo tópico que descreve como registrar chamadas da API Amazon SWF usando CloudTrail:. Gravando chamadas de API com AWS CloudTrail	8 de maio de 2014
Atualizar	Dois novos tópicos relacionados às CloudWatch métricas do Amazon SWF foram adicionados:Métricas do Amazon SWF para CloudWatch, que fornece uma lista e descrições das métricas suportadas eVisualizando as métricas do Amazon SWF para usar o CloudWatch AWS Management Console, que fornece informações sobre como visualizar métricas e definir alarmes com o. AWS Management Console	28 de abril de 2014
Atualizar	Adição de uma nova seção: Recursos adicionais e informações de referência para o Amazon SWF. Esta seção fornece algumas informações de referência de serviço e fornece informações sobre documentação adicional, amostras, código e outros recursos da web para desenvolvedores do Amazon SWF.	19 de março de 2014
Atualizar	Adição de um tutorial de fluxo de trabalho. Consulte Começando a usar o Amazon SWF.	25 de outubro de 2013
Atualizar	Adição de informações e exemplos de AWS CLI.	26 de agosto de 2013
Atualizar	Atualizações e correções.	1 de agosto de 2013
Atualizar	Atualizou o documento para descrever como usar o IAM para controle de acesso.	22 de fevereiro de 2013

Alteração	Descrição	Alterado em
Versão inicial	Esta é a primeira versão do Guia do Desenvolvedor do Amazon Simple Workflow Service.	16 de outubro de 2012

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.