



Guida per gli sviluppatori

Amazon Simple Workflow Service



Versione API 2012-01-25

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

Amazon Simple Workflow Service: Guida per gli sviluppatori

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione ad alcun prodotto o servizio che non sia di Amazon, in alcun modo che possa causare confusione tra i clienti, né in alcun modo che possa denigrare o screditare Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è Amazon SWF	1
Componenti del workflow	2
Componenti del workflow	2
Esecuzione del flusso di lavoro	4
Configurazione dell'ambiente di sviluppo	4
Sviluppa con AWS SDKs	5
Considera il AWS Flow Framework	5
Nozioni di base	7
Informazioni sul flusso di lavoro	8
Prerequisiti	8
Fasi del tutorial	9
Parte 1: Utilizzo di Amazon SWF con l'SDK for Ruby	9
Includi il AWS SDK per Ruby	10
Configurazione della sessione AWS	10
Registrazione di un dominio Amazon SWF	11
Fasi successive	13
Seconda parte: implementazione del flusso di lavoro	13
Progettazione del flusso di lavoro	13
Configurazione del codice del flusso di lavoro	14
Registrazione del flusso di lavoro	16
Polling delle decisioni	17
Avvio dell'esecuzione di flusso di lavoro	20
Fasi successive	22
Terza parte: implementazione delle attività	23
Definizione di un tipo di attività di base	23
Definendo GetContactActivity	25
Definizione SubscribeTopicActivity	27
Definizione WaitForConfirmationActivity	31
Definizione SendResultActivity	33
Fasi successive	35
Quarta parte: implementazione del poller dei task di attività	35
Esecuzione del flusso di lavoro	38
Cosa fare in seguito?	42
Lavorare nella console	43

Registrazione di un dominio	43
Registrazione dei tipi di flusso di lavoro	44
Registrazione dei tipi di attività	45
Avvio di un flusso di lavoro	45
Per avviare un'esecuzione utilizzando la console	46
Gestione delle esecuzioni dei flussi di lavoro	46
Concetti di base	50
Creazione di un flusso di lavoro	51
Modellazione del flusso di lavoro e delle relative attività	52
Flussi di lavoro in esecuzione	53
Storia del flusso di lavoro	53
Identificatori di oggetti	58
Domini	59
Attori	60
Cos'è un attore in Amazon SWF?	60
Starter di flusso di lavoro	61
Decisori	61
Lavoratori di attività	63
Scambio di dati tra attori	63
Attività	64
Elenchi di attività	65
Elenchi di task di decisione	66
Elenco di task di attività	66
Instradamento di task	66
Chiusura dell'esecuzione del workflow	67
Ciclo di vita dell'esecuzione del workflow	68
Ciclo di vita di un'esecuzione di flusso di lavoro	68
Sondaggi per le attività	75
Concetti avanzati	77
Controllo delle versioni	77
Segnali	78
Flussi di lavoro per bambini	80
Contrassegni	82
Tag	83
Gestione dei tag	83
Esecuzioni di flussi di lavoro di tag	84

Controlla l'accesso ai domini con i tag	86
Scelta esclusiva	86
Timer	89
Annulloamento delle attività	90
Sicurezza	93
Protezione dei dati	93
Crittografia	94
Identity and Access Management	95
Destinatari	96
Autenticazione con identità	96
Gestione dell'accesso tramite policy	98
Controllo degli accessi	99
Operazioni di policy	100
Risorse relative alle policy	100
Chiavi di condizione delle policy	101
ACLs	101
ABAC	101
Credenziali temporanee	102
Autorizzazioni dell'entità principale	102
Ruoli di servizio	102
Ruoli collegati ai servizi	103
Policy basate sull'identità	103
Policy basate sulle risorse	104
Come funziona Amazon Simple Workflow Service con IAM	104
Esempi di policy basate sull'identità	105
Principi di base	108
Politiche IAM di Amazon SWF	109
Riepilogo delle API	115
Policy basate su tag	124
Endpoint Amazon VPC	124
Risoluzione dei problemi	126
Registrazione e monitoraggio	128
Metriche di Amazon SWF per CloudWatch	129
Visualizzazione dei parametri di Amazon SWF	139
Registrazione su CloudTrail	143
EventBridge per Amazon SWF	150

Utilizzo Notifiche all'utente AWS con Amazon SWF	159
Convalida della conformità	159
Resilienza	160
Sicurezza dell'infrastruttura	161
Analisi della configurazione e delle vulnerabilità	161
Usando il AWS CLI	162
Lavorare con APIs	164
Effettuare richieste HTTP	164
Contenuti nell'intestazione HTTP	165
Contenuto del corpo HTTP	167
Richiesta e risposta JSON di esempio	167
Calcolare la firma HMAC-SHA	168
Elenco delle azioni Amazon SWF	171
Operazioni correlate alle attività	171
Operazioni correlate ai decisori	172
Operazioni correlate alle esecuzioni di flusso di lavoro	172
Operazioni correlate all'amministrazione	172
Operazioni di visibilità	173
Registrazione di un dominio	174
Vedi anche	175
Impostazione dei valori di timeout	175
Quote sui valori di timeout	175
Timeout dell'esecuzione del flusso di lavoro e del task di decisione	176
Timeout del task di attività	176
Vedi anche	177
Registrazione di un tipo di flusso di lavoro	177
Vedi anche	178
Registrazione di un tipo di attività	178
Vedi anche	178
Attività Lambda	179
Informazioni su AWS Lambda	179
Vantaggi e limiti dell'utilizzo delle attività Lambda	179
Utilizzo delle attività Lambda nei flussi di lavoro	180
Sviluppo di un lavoratore di attività	185
Esecuzione del polling dei task di attività	186
Esecuzione di un task di attività	186

Segnalazione di heartbeat di task di attività	187
Task di attività completato o non riuscito	187
Avvio di lavoratori di attività	189
Sviluppo dei decisori	189
Definire la logica di coordinamento	191
Esecuzione del polling dei task di decisione	191
Applicare la logica di coordinamento	193
Rispondere con decisioni	194
Chiudere un'esecuzione del flusso di lavoro	195
Lancio dei decisori	196
Avvio di flussi di lavoro	197
Impostazione della priorità delle attività	198
Impostazione della priorità di task per flussi di lavoro	199
Impostazione della priorità di task per attività	201
Operazioni che restituiscono informazioni relative alla priorità di task	202
Gestione degli errori	203
Errori di convalida	203
Errori nell'applicazione di operazioni o decisioni	204
Timeout	204
Errori generati da codice utente	205
Errori correlati alla chiusura di un'esecuzione di flusso di lavoro	205
Quote	206
Quote generali degli account per Amazon SWF	206
Quote sulle esecuzioni dei flussi di lavoro	207
Quote sulle esecuzioni delle attività	208
Quote di limitazione di Amazon SWF	209
Limitazione delle quote per tutte le regioni	209
Quote decisionali per tutte le regioni	212
Quote a livello di flusso di lavoro	212
Richiesta di aumento delle quote	213
Risorse aggiuntive	214
Tipi di timeout	214
I timeout nel flusso di lavoro e i task di decisione	215
Timeout nei task di attività	216
Endpoints	217
Documentazione supplementare di	217

Riferimento all'API Amazon Simple Workflow Service	218
AWS Flow Framework Documentazione	218
AWS Documentazione SDK	218
AWS CLI Documentazione	220
Risorse Web	220
Forum Amazon SWF	220
Domande frequenti su Amazon SWF	221
Video di Amazon SWF	221
Opzioni Ruby Flow	221
Continua a utilizzare Ruby Flow Framework	222
Esegui la migrazione a Java Flow Framework	222
Migrare a Step Functions	222
Usa direttamente l'API Amazon SWF	223
Cronologia dei documenti	224

ccxxviii

Cos'è Amazon Simple Workflow Service?

Con Amazon Simple Workflow Service (Amazon SWF) puoi creare, eseguire e scalare lavori in background con passaggi paralleli o sequenziali. Puoi coordinare il lavoro tra i componenti distribuiti e monitorare lo stato delle attività.

In Amazon SWF, un'attività rappresenta un'unità logica di lavoro eseguita da un componente dell'applicazione. Il coordinamento delle attività include la gestione delle dipendenze tra le attività, la pianificazione e la concorrenza nel flusso dell'applicazione. Con Amazon SWF, puoi controllare e coordinare le attività senza preoccuparti delle complessità sottostanti, come il monitoraggio dei progressi e il mantenimento dello stato delle attività.

Quando si utilizza Amazon SWF, si implementano i lavoratori per eseguire le attività. I lavoratori possono lavorare su un'infrastruttura cloud, come Amazon Elastic Compute Cloud (Amazon EC2), o presso la tua sede. Puoi creare attività di lunga durata o che potrebbero fallire, scadere o richiedere riavvii, oppure che potrebbero essere completate con velocità effettiva e latenza variabili. Amazon SWF archivia le attività e le assegna ai lavoratori quando sono pronti, monitora i progressi e mantiene lo stato, inclusi i dettagli del completamento delle attività.

Per coordinare le attività, scrivi un programma che ottiene lo stato delle attività più recente da Amazon SWF e lo utilizza per avviare le attività successive. Amazon SWF mantiene lo stato di esecuzione di un'applicazione in modo duraturo, quindi l'applicazione è resiliente ai guasti dei singoli componenti. Con Amazon SWF, puoi creare, distribuire, scalare e modificare i componenti delle applicazioni in modo indipendente.

Altri servizi AWS per il flusso di lavoro

Per la maggior parte dei casi d'uso, ti consigliamo di prenderlo in considerazione AWS Step Functions in base alle tue esigenze di workflow e orchestrazione.

Con Step Functions, puoi creare flussi di lavoro, chiamati anche macchine a stati, per creare applicazioni distribuite, automatizzare processi, orchestrare microservizi e creare pipeline di dati e apprendimento automatico. Nella console o nel AWS toolkit di Step Functions in VS Code, puoi utilizzare la versione grafica di Workflow Studio per visualizzare, modificare, testare ed eseguire il debug del flusso di lavoro dell'applicazione.

[Per ulteriori informazioni tecniche, consulta la Guida per gli sviluppatori.AWS Step Functions](#)

Sviluppo di componenti del flusso di lavoro con Amazon SWF

Lo sviluppo di applicazioni distribuite richiede il coordinamento di molti componenti e la gestione della latenza e dell'inaffidabilità inerenti alla comunicazione remota.

Con Amazon Simple Workflow Service (Amazon SWF), puoi sviluppare applicazioni asincrone e distribuite fornendo un modello di programmazione e un'infrastruttura per coordinare i componenti distribuiti e mantenerne lo stato di esecuzione in modo affidabile. Affidandoti ad Amazon SWF, sei libero di concentrarti sulla creazione degli aspetti della tua applicazione che la differenziano.

Componenti di un flusso di lavoro

[Componenti di un flusso di lavoro](#) Il concetto fondamentale di Amazon SWF è il flusso di lavoro. Un flusso di lavoro è un set di attività che, insieme alla logica necessaria per coordinarle, consentono di raggiungere un obiettivo. Ad esempio, un flusso di lavoro potrebbe ricevere l'ordine di un cliente e intraprendere le azioni necessarie per evadere l'ordine.

Ogni flusso di lavoro viene eseguito in una risorsa chiamata dominio, che controlla l'ambito del flusso di lavoro. Un AWS account può avere più domini, ognuno dei quali può contenere più flussi di lavoro, ma i flussi di lavoro in domini diversi non possono interagire.

Quando progetti un flusso di lavoro Amazon SWF, definisci ciascuna delle attività richieste. Quindi registri ogni attività con Amazon SWF come tipo di attività. Fornirai un nome, una versione e valori di timeout. Ad esempio, se un cliente prevede di ricevere un ordine entro 24 ore,

Durante l'esecuzione del flusso di lavoro, è possibile che alcune attività debbano essere eseguite più volte ed eventualmente con input variabili. Ad esempio, in un flusso di lavoro relativo a un ordine cliente, potresti avere un'attività che gestisce gli articoli acquistati. Se il cliente acquista più articoli, questa attività dovrebbe essere eseguita più volte. Amazon SWF ha il concetto di attività che rappresenta una chiamata di un'attività. Nel nostro esempio, l'elaborazione di ogni articolo sarebbe rappresentata da un singolo task di attività.

Un activity worker è un programma che riceve attività, le esegue e fornisce risultati. L'attività potrebbe essere effettivamente eseguita da una persona. Ad esempio, un analista statistico potrebbe ricevere set di dati, analizzarli e quindi restituire le proprie analisi.

Le attività e gli addetti alle attività che le eseguono possono essere eseguite in modo sincrono o asincrono. I lavoratori possono lavorare in un'unica sede o essere distribuiti su più computer, potenzialmente in diverse aree geografiche. Lavoratori di attività differenti possono essere scritti in linguaggi di programmazione differenti ed essere eseguiti su sistemi operativi diversi. Ad esempio, un Activity Worker potrebbe essere in esecuzione su un server in Asia, mentre un altro potrebbe essere in esecuzione su un dispositivo mobile in Nord America.

La logica di coordinamento in un flusso di lavoro è contenuta in un programma software denominato decisore. Un decisore pianifica le attività, fornisce input agli addetti alle attività, elabora gli eventi che arrivano mentre il flusso di lavoro è in corso e termina (o chiude) il flusso di lavoro dopo il raggiungimento dell'obiettivo.

Il ruolo del servizio Amazon SWF è quello di fungere da hub centrale affidabile attraverso il quale i dati vengono scambiati tra chi decide, gli addetti all'attività e altre entità pertinenti come la persona che amministra il flusso di lavoro. Amazon SWF mantiene anche lo stato di ogni esecuzione del flusso di lavoro, evitando all'applicazione di dover archiviare lo stato in modo durevole.

Il decisore dirige il flusso di lavoro ricevendo attività decisionali da Amazon SWF e rispondendo ad Amazon SWF con decisioni. Una decisione rappresenta un'azione o un insieme di azioni, che sono le fasi successive del flusso di lavoro. Una decisione tipica sarebbe di pianificare un task di attività. Le decisioni possono essere utilizzate anche per ritardare le attività con timer, richiedere l'annullamento di attività in corso e completare i flussi di lavoro.

Il meccanismo mediante il quale sia gli addetti all'attività che chi decide ricevono i propri compiti (rispettivamente compiti di attività e compiti decisionali) consiste nel sondare il servizio Amazon SWF.

Amazon SWF informa chi decide lo stato del flusso di lavoro includendo, in ogni attività decisionale, una copia della cronologia di esecuzione corrente del flusso di lavoro. Questa cronologia è costituita da eventi, dove ogni evento rappresenta una modifica significativa dello stato dell'esecuzione di flusso di lavoro. Esempi di eventi includono il completamento delle attività, i timeout delle attività o la scadenza di un timer. La cronologia è un record completo, coerente e attendibile dell'avanzamento del flusso di lavoro.

Amazon SWF Access Control Uses AWS Identity and Access Management (IAM), così puoi controllare l'accesso alle AWS risorse. Ad esempio, puoi autorizzare un utente a accedere al tuo account ma soltanto per eseguire determinati flussi di lavoro in un particolare dominio.

Esecuzione del flusso di lavoro

Di seguito viene fornita una panoramica dei passaggi necessari per sviluppare ed eseguire un flusso di lavoro in Amazon SWF:

1. Scrivi agli addetti alle attività per eseguire le fasi di elaborazione del tuo flusso di lavoro.
2. Scrivi un decisore per gestire la logica di coordinamento del tuo flusso di lavoro.
3. Registra le tue attività e il tuo flusso di lavoro con Amazon SWF.

Puoi eseguire questo passaggio a livello di codice o utilizzando il [Console di gestione AWS](#)

4. Avvia i lavoratori di attività e il decisore.

Questi attori possono funzionare su qualsiasi dispositivo informatico in grado di accedere a un endpoint Amazon SWF. Ad esempio, puoi utilizzare istanze di calcolo nel cloud, come Amazon Elastic Compute Cloud (Amazon EC2), server nel tuo data center o persino un dispositivo mobile per ospitare un decisore o un addetto alle attività. Una volta avviato, chi decide e chi lavora all'attività dovrebbe iniziare a chiedere ad Amazon SWF le attività da svolgere.

5. Avvia una o più esecuzioni del flusso di lavoro.

Puoi avviare i flussi di lavoro a livello di codice o tramite [Console di gestione AWS](#)

Ogni esecuzione viene avviata in modo indipendente e a ognuna puoi fornire uno specifico set di dati di input. Quando viene avviata un'esecuzione, Amazon SWF pianifica l'attività decisionale iniziale. In risposta, chi decide inizia a generare decisioni che avviano attività. L'esecuzione continua fino a che il decisore non prende la decisione di chiuderla.

6. Visualizza le esecuzioni dei flussi di lavoro utilizzando [Console di gestione AWS](#)

È possibile filtrare e visualizzare i dettagli completi delle esecuzioni in esecuzione e completate. Ad esempio, è possibile selezionare un'esecuzione aperta per vedere quali attività sono state completate e quali sono stati i relativi risultati.

Configurazione dell'ambiente di sviluppo

Hai la possibilità di sviluppare per Amazon SWF in uno qualsiasi dei linguaggi di programmazione supportati da AWS. Per gli sviluppatori Java, AWS Flow Framework è disponibile anche. Per ulteriori informazioni, consultate il [AWS Flow Framework](#) sito Web e consultate [AWS Flow Framework la Java Developer Guide](#).

Per ridurre la latenza e archiviare i dati in una posizione che soddisfi i tuoi requisiti, Amazon SWF fornisce endpoint in diverse regioni.

Ogni endpoint in Amazon SWF è completamente indipendente. Tutti i domini, i flussi di lavoro e le attività che hai registrato in una regione non condivideranno dati o attributi con quelli di un'altra regione.

Quando registri un dominio, un flusso di lavoro o un'attività Amazon SWF, questo esiste solo all'interno della regione in cui lo hai registrato. Ad esempio, puoi registrare un dominio denominato SWF-Flows-1 in due regioni diverse, ma non condivideranno dati o attributi tra loro, ognuna delle quali fungerà da dominio completamente indipendente.

Per un elenco degli endpoint Amazon SWF, consulta [Regioni](#) ed endpoint.

Sviluppa con AWS SDKs

Amazon SWF è supportato da Java, .NET, Node.js, PHP, Python e Ruby e offre un modo pratico AWS SDKs per utilizzare l'API HTTP di Amazon SWF nel linguaggio di programmazione di tua scelta.

Puoi sviluppare dispositivi decisionali, activity worker o iniziatori di flussi di lavoro utilizzando l'API esposta da queste librerie. Inoltre, puoi utilizzare le operazioni di visibilità tramite queste librerie in modo da sviluppare i tuoi strumenti di monitoraggio e reporting di Amazon SWF.

Per scaricare strumenti per lo sviluppo e la gestione di applicazioni su AWS, tra cui SDKs, vai al [Developer Center](#).

Per informazioni dettagliate sulle operazioni di Amazon SWF in ogni SDK, consulta la documentazione di riferimento specifica della lingua per l'SDK.

Considera il AWS Flow Framework

AWS Flow Framework È un SDK avanzato per la scrittura di programmi distribuiti e asincroni che vengono eseguiti come flussi di lavoro su Amazon SWF. Il framework è disponibile per il linguaggio di programmazione Java e fornisce classi per la scrittura di programmi distribuiti complessi.

Con AWS Flow Framework, è possibile utilizzare tipi preconfigurati per mappare la definizione del flusso di lavoro direttamente ai metodi del programma. AWS Flow Framework Supporta concetti standard orientati agli oggetti, come la gestione degli errori basata sulle eccezioni. I programmi scritti con AWS Flow Framework possono essere creati, eseguiti e sottoposti a debug

interamente all'interno dell'editor o dell'IDE preferito. Per ulteriori informazioni, consultate il [AWS Flow Framework](#)sito Web e consultate la [Java Developer AWS Flow Framework Guide](#).

Guida introduttiva ad Amazon SWF

Puoi iniziare con la seguente applicazione per il flusso di lavoro di Amazon Simple Workflow Service, che consiste in un set di quattro attività che operano in sequenza. Il tutorial tratta anche i seguenti argomenti:

- L'impostazione di opzioni di attività e di flusso di lavoro predefinite e nel tempo di esecuzione.
- Eseguire sondaggi su Amazon SWF per attività e decisioni.
- Trasferimento di dati tra le attività e il flusso di lavoro con Amazon SWF.
- Aspettando attività umane e segnalando i battiti cardiaci ad Amazon SWF da un'attività.
- Utilizzo di Amazon SNS per creare un argomento, iscrivere un utente e pubblicare messaggi sugli endpoint sottoscritti.

Puoi usare Amazon SWF e Amazon Simple Notification Service (Amazon SNS) insieme per emulare un flusso di lavoro «human task», in cui è necessario che un lavoratore umano esegua alcune azioni e poi comunichi con Amazon SWF per avviare l'attività successiva del flusso di lavoro.

Poiché Amazon SWF è un servizio Web basato sul cloud, la comunicazione con Amazon SWF può provenire da qualsiasi luogo sia disponibile una connessione a Internet. In questo caso, utilizzeremo Amazon SNS per comunicare con l'utente tramite e-mail, SMS o entrambi.

Questo tutorial utilizza [AWS SDK per Ruby](#) per accedere ad Amazon SWF e Amazon SNS, ma sono disponibili molte opzioni di sviluppo, inclusa AWS Flow Framework quella per Ruby, che semplifica il coordinamento e la comunicazione con Amazon SWF.

Note

[Questo tutorial utilizza AWS SDK per Ruby, ma ti consigliamo di utilizzare il per Java.AWS Flow Framework](#)

Argomenti

- [Informazioni sul flusso di lavoro](#)
- [Prerequisiti](#)
- [Fasi del tutorial](#)

- [Tutorial sul flusso di lavoro degli abbonamenti Parte 1: Utilizzo di Amazon SWF con AWS SDK per Ruby](#)
- [Seconda parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del flusso di lavoro](#)
- [Terza parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione delle attività](#)
- [Quarta parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del poller dei task di attività](#)
- [Tutorial sul flusso di lavoro di sottoscrizione: esecuzione del flusso di lavoro](#)

Informazioni sul flusso di lavoro

Il flusso di lavoro che andiamo a sviluppare consiste in quattro fasi principali:

1. Ricevere un indirizzo di sottoscrizione (e-mail o SMS) dall'utente.
2. Creare un argomento SNS e sottoscrivervi gli endpoint disponibili.
3. Attendere che l'utente confermi la sottoscrizione.
4. In caso di conferma dell'utente, pubblica un messaggio di congratulazioni sull'argomento.

Le fasi includono attività completamente automatizzate (fasi 2 e 4) e altre che richiedono al flusso di lavoro di attendere che un umano fornisca alcuni dati all'attività prima che il flusso di lavoro possa proseguire (fasi 1 e 3).

Ogni fase si basa sui dati generati dalla fase precedente (devi avere un endpoint prima di sottoscriverlo a un argomento, una sottoscrizione a un argomento prima di poter attendere la conferma e così via). Questo tutorial spiegherà anche come fornire i risultati delle attività una volta completata e come trasmettere input a un'attività che viene pianificata. Amazon SWF gestisce il coordinamento e la distribuzione di informazioni tra le attività e il flusso di lavoro e viceversa.

Utilizziamo anche l'input da tastiera e Amazon SNS per gestire la comunicazione tra Amazon SWF e l'essere umano che fornisce dati al flusso di lavoro. In pratica, puoi utilizzare molte tecniche diverse per comunicare con utenti umani, ma Amazon SNS offre un modo molto semplice per utilizzare e-mail o messaggi di testo per notificare all'utente gli eventi del flusso di lavoro.

Prerequisiti

Per proseguire il tutorial, hai bisogno dei seguenti elementi:

- [Account Amazon Web Services](#)
- [Un interprete Ruby](#)
- [AWS SDK per Ruby](#)

Se hai già questi elementi configurati, sei pronto a continuare. Se non vuoi eseguire l'esempio, puoi comunque seguire il tutorial: gran parte del contenuto di questo tutorial si riferisce all'uso di Amazon SWF e Amazon SNS indipendentemente dall'opzione di sviluppo scelta.

Fasi del tutorial

Il tutorial si divide nelle seguenti fasi:

1. [Tutorial sul flusso di lavoro degli abbonamenti Parte 1: Utilizzo di Amazon SWF con AWS SDK per Ruby](#)
2. [Seconda parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del flusso di lavoro](#)
3. [Terza parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione delle attività](#)
4. [Quarta parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del poller dei task di attività](#)
5. [Tutorial sul flusso di lavoro di sottoscrizione: esecuzione del flusso di lavoro](#)

Tutorial sul flusso di lavoro degli abbonamenti Parte 1: Utilizzo di Amazon SWF con AWS SDK per Ruby

Argomenti

- [Includi il AWS SDK per Ruby](#)
- [Configurazione della sessione AWS](#)
- [Registrazione di un dominio Amazon SWF](#)
- [Fasi successive](#)

Includi il AWS SDK per Ruby

Per cominciare, creiamo un file denominato `utils.rb`. Il codice contenuto in questo file otterrà, o creerà se necessario, il dominio Amazon SWF utilizzato sia dal codice del flusso di lavoro che dal codice delle attività e fornirà una posizione in cui inserire il codice comune a tutte le nostre classi.

Innanzitutto, dobbiamo includere la `aws-sdk-v1` libreria nel nostro codice, in modo da poter utilizzare le funzionalità fornite dall'SDK for Ruby.

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

Questo ci dà accesso allo spazio dei AWS nomi, che offre la possibilità di impostare valori globali relativi alla sessione, come AWS le credenziali e la regione, e fornisce anche l'accesso al servizio AWS APIs.

Configurazione della sessione AWS

Configureremo la AWS sessione impostando AWS le nostre credenziali (necessarie per accedere ai AWS servizi) e la AWS regione da utilizzare.

Esistono diversi modi per [impostare AWS le credenziali nell' AWS SDK per Ruby](#): impostandole nelle variabili di ambiente `AWS_ACCESS_KEY_ID` `AWS_SECRET_ACCESS_KEY` e `AWS_ACCESS_KEY` o impostandole con [AWS.config](#). In questo tutorial, utilizzeremo il secondo metodo, caricandole da un file di configurazione YAML denominato `aws-config.txt`, come illustrato di seguito.

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

Crea subito questo file, sostituendo le stringhe che iniziano con `REPLACE_WITH_` con l'ID della chiave di accesso e la chiave di accesso segreta. AWS Per informazioni sulle tue chiavi di AWS accesso, vedi [Come posso ottenere le credenziali di sicurezza?](#) nel riferimento generale di Amazon Web Services.

Dobbiamo anche impostare la AWS regione da utilizzare. Poiché utilizzeremo lo [Short Message Service \(SMS\)](#) per inviare messaggi di testo al telefono dell'utente con Amazon SNS, dobbiamo assicurarci di utilizzare la regione supportata da Amazon SNS. Consulta [le regioni e i Paesi supportati](#) nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Note

Se non hai accesso a us-east-1 o non sei interessato a eseguire la demo con la messaggistica SMS attivata, imposta la regione che desideri. Puoi rimuovere la funzionalità SMS dall'esempio e utilizzare l'e-mail come unico endpoint per iscriverti all'argomento Amazon SNS.

Per ulteriori informazioni sull'invio di messaggi SMS, consulta [Invio e ricezione di notifiche SMS con Amazon SNS nella Amazon Simple Notification Service Developer Guide](#).

Ora aggiungeremo del codice `utils.rb` per caricare il file di configurazione, ottenere le credenziali dell'utente e fornire sia le credenziali che la regione. [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
```

Registrazione di un dominio Amazon SWF

Per utilizzare Amazon SWF, devi configurare un dominio: un'entità denominata che conterrà i tuoi flussi di lavoro e le tue attività. Puoi avere molti domini Amazon SWF registrati, ma tutti devono avere

nomi univoci all'interno del tuo AWS account e i flussi di lavoro non possono interagire tra domini: tutti i flussi di lavoro e le attività della tua applicazione devono trovarsi nello stesso dominio per interagire tra loro.

Poiché utilizzeremo lo stesso dominio in tutta la nostra applicazione, creeremo una funzione in `named` che recupererà il dominio SWFSample Amazon SWF `utils.rb` denominato `init_domain` Domain.

Dopo aver registrato un dominio, puoi riutilizzarlo per un gran numero di esecuzioni di flusso di lavoro. Tuttavia, è un errore cercare di registrare un dominio esistente, di conseguenza il nostro codice dapprima verificherà se esiste e se è così, lo utilizzerà. In caso contrario, lo dovremo creare.

Per lavorare con i domini Amazon SWF nell'SDK per Ruby, [AWS::Simpleusa](#) `Workflow.domains`, che restituisce [DomainCollection](#) un file che può essere utilizzato sia per enumerare che per registrare domini:

- [Per verificare se un dominio è già registrato, puoi consultare l'elenco fornito da `.domains.registered`. AWS::Simpleworkflow](#)
- [Per registrare un nuovo dominio, usa `.domains.register`. AWS::Simpleworkflow](#)

Di seguito è illustrato il codice per `init_domain` in `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
```

Fasi successive

Successivamente, creeremo il codice del flusso di lavoro e dello starter come descritto in [Seconda parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del flusso di lavoro](#).

Seconda parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del flusso di lavoro

Il codice che abbiamo creato fino a ora è alquanto generico. In questa parte del tutorial cominceremo quindi a definire realmente la funzione del nostro flusso di lavoro e le attività necessarie per implementarla.

Argomenti

- [Progettazione del flusso di lavoro](#)
- [Configurazione del codice del flusso di lavoro](#)
- [Registrazione del flusso di lavoro](#)
- [Polling delle decisioni](#)
- [Avvio dell'esecuzione di flusso di lavoro](#)
- [Fasi successive](#)

Progettazione del flusso di lavoro

L'idea iniziale di questo flusso di lavoro comprendeva le seguenti fasi:

1. Ricevere un indirizzo di sottoscrizione (e-mail o SMS) dall'utente.
2. Creare un argomento SNS e sottoscrivervi gli endpoint disponibili.
3. Attendere che l'utente confermi la sottoscrizione.
4. In caso di conferma dell'utente, pubblica un messaggio di congratulazioni sull'argomento.

Possiamo considerare ogni fase del nostro flusso di lavoro come un'attività che deve eseguire. Il flusso di lavoro ha la responsabilità di pianificare ogni attività al momento opportuno e di coordinare il trasferimento di dati tra le attività.

Per questo flusso di lavoro, creeremo un'attività distinta per ogni fase, a cui assegneremo i nomi descrittivi seguenti:

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

Queste attività saranno eseguite nell'ordine in cui sono elencate e i dati di ogni fase saranno utilizzati nella fase successiva.

Potremmo progettare la nostra applicazione in modo che tutto il codice esista in un unico file sorgente, ma ciò è contrario al modo in cui è stato progettato Amazon SWF. ovvero per flussi di lavoro il cui ambito copre l'integralità di Internet. Di conseguenza, suddividiamo l'applicazione in due eseguibili distinti:

- `swf_sns_workflow.rb` – Contiene il flusso di lavoro e il relativo starter.
- `swf_sns_activities.rb` – Contiene le attività e il relativo starter.

Le implementazioni di flusso di lavoro e attività possono essere eseguite in finestre o computer distinti o addirittura in differenti aree del mondo. Poiché Amazon SWF tiene traccia dei dettagli dei flussi di lavoro e delle attività, il flusso di lavoro può coordinare la pianificazione e il trasferimento dei dati delle attività indipendentemente da dove vengono eseguite.

Configurazione del codice del flusso di lavoro

Per prima cosa, creeremo un file denominato `swf_sns_workflow.rb`. In questo file, dichiara una classe chiamata `SampleWorkflow`. Di seguito è riportata la dichiarazione di classe e il relativo costruttore, il metodo `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
```

Come puoi vedere, conserviamo i seguenti dati dell'istanza di classe:

- `domain` – Il nome di dominio recuperato da `init_domain` in `utils.rb`.
- `workflowId` – L'elenco di task passato a `initialize`.
- `activity_list` – L'elenco di attività, con i nomi e le versioni delle attività che eseguiremo.

Il nome di dominio, il nome dell'attività e la versione dell'attività sono sufficienti per consentire ad Amazon SWF di identificare con certezza un tipo di attività, quindi questi sono tutti i dati che dobbiamo conservare sulle nostre attività per pianificarle.

L'elenco di task verrà utilizzato dal codice decisore del flusso di lavoro per eseguire il polling dei task di decisione e delle attività di pianificazione.

Al termine di questa funzione, chiamiamo un metodo che non abbiamo ancora definito, ovvero `register_workflow`. Definiremo questo metodo in seguito.

Registrazione del flusso di lavoro

Per utilizzare un tipo di flusso di lavoro, dobbiamo prima registrarlo. Come un tipo di attività, un tipo di flusso di lavoro è identificato dal relativo dominio, nome e versione. Inoltre, come per i domini e i tipi di attività, non è possibile registrare di nuovo un tipo di flusso di lavoro esistente. Se hai la necessità di modificare un tipo di flusso di lavoro, devi farlo mediante una nuova versione, che in pratica crea un nuovo tipo.

Di seguito è riportato il codice `register_workflow`, che utilizziamo per recuperare il tipo di flusso di lavoro esistente registrato durante un'esecuzione precedente oppure per registrare il flusso di lavoro se questa operazione non è ancora stata eseguita.

```
# 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
```

Per prima cosa, verifichiamo se il nome e la versione del flusso di lavoro sono già registrati scorrendo la raccolta [workflow_types](#) del dominio. Se troviamo una corrispondenza, utilizzeremo il tipo di flusso di lavoro già registrato.

Se non troviamo una corrispondenza, viene registrato un nuovo tipo di flusso di lavoro (chiamando [register](#) nella stessa `workflow_types` raccolta in cui stavamo cercando il flusso di lavoro) con il nome 'swf-sns-workflow', la versione '1' e le seguenti opzioni.

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

Le opzioni passate durante la registrazione sono utilizzate per impostare il comportamento di default del tipo di flusso di lavoro. Di conseguenza, non abbiamo bisogno di impostare questi valori ogni volta che avviamo una nuova esecuzione di flusso di lavoro.

Qui impostiamo soltanto alcuni valori di timeout: il periodo di tempo massimo tra l'avvio di un task e la chiusura dello stesso (un'ora) e la durata massima dell'esecuzione di flusso di lavoro (24 ore). Se uno di questi valori viene superato, si verifica il timeout del task o del flusso di lavoro.

Per ulteriori informazioni sui valori di timeout, consulta [Tipi di timeout di Amazon SWF](#).

Polling delle decisioni

Al centro di ogni esecuzione di flusso di lavoro si trova un decisore. Il ruolo del decisore è di gestire l'esecuzione del flusso di lavoro. Il decisore riceve i task di decisione e risponde agli stessi pianificando nuove attività, annullando e riavviando attività o definendo lo stato dell'esecuzione di flusso di lavoro come completa, annullata o non riuscita.

Il decisore utilizza il nome dell'elenco di task del esecuzione di flusso di lavoro per ricevere task di decisione a cui rispondere. Per eseguire il polling dei task di decisione, chiama [poll](#) sulla raccolta [decision_tasks](#) del dominio per scorrere i task di decisione disponibili. Successivamente, puoi cercare nuovi eventi nel task di decisione scorrendo la relativa raccolta [new_events](#).

Gli eventi restituiti sono [AWS::SimpleWorkflow::HistoryEvent](#) oggetti ed è possibile ottenere il tipo di evento utilizzando il membro [event_type](#) dell'evento restituito. Per un elenco e una descrizione dei tipi di eventi cronologici, consulta [HistoryEvent](#) Amazon Simple Workflow Service API Reference.

Di seguito viene riportato l'inizio della logica del poller dei task di decisione. Un nuovo metodo nella nostra classe di flusso di lavoro denominato `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
```

Ora creeremo diramazioni dell'esecuzione del decisore in base al valore `event_type` ricevuto. Il primo che probabilmente riceveremo è `WorkflowExecutionStarted`. Quando viene ricevuto questo evento, significa che Amazon SWF sta segnalando al decisore che deve iniziare l'esecuzione del flusso di lavoro. Cominceremo quindi col pianificare la prima attività chiamando [schedule_activity_task](#) sul task ricevuto durante il polling.

A questo metodo passeremo la prima attività dichiarata nel nostro elenco di attività. Questa attività occupa la posizione `last` nell'elenco, in quanto abbiamo invertito quest'ultimo per poterlo utilizzare come stack. Le «attività» che abbiamo definito sono solo mappe composte da un nome e un numero di versione, ma questo è tutto ciò di cui Amazon SWF ha bisogno per identificare l'attività per la pianificazione, supponendo che l'attività sia già stata registrata.

```
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 pianifichiamo un'attività, Amazon SWF invia un'attività all'elenco delle attività che trasmettiamo durante la pianificazione, segnalando l'inizio dell'attività. I task di attività sono descritti nella sezione [Terza parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione delle attività](#), ma vale comunque la pena segnalare che non eseguiamo il task in questa fase. Ad Amazon SWF diciamo solo che deve essere pianificato.

La prossima attività che dobbiamo affrontare è l'`ActivityTaskCompleted` evento, che si verifica quando Amazon SWF riceve una risposta al completamento dell'attività da un'attività.

```
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
```

Poiché eseguiamo le nostre attività in modo lineare e viene eseguita solo un'attività alla volta, coglieremo l'occasione per inserire l'attività completata dallo `activity_list` stack. Se viene restituito un elenco vuoto, significa che il nostro flusso di lavoro è completato. In questo caso, segnaliamo ad Amazon SWF che il nostro flusso di lavoro è completo chiamando [complete_workflow_execution](#) sull'attività.

Se invece l'elenco non è vuoto, pianificheremo l'attività successiva (sempre in ultima posizione). Questa volta, tuttavia, esamineremo se l'attività precedente ha restituito ad Amazon SWF dati di risultato al completamento, che vengono forniti al flusso di lavoro negli attributi dell'evento, nella chiave `optionalResult`. Se l'attività ha generato un risultato, lo passeremo come opzione `input` all'attività pianificata successiva, insieme all'elenco dei task di attività.

Recuperando i valori `result` delle attività completate e impostando i valori `input` delle attività pianificate, possiamo passare dati da un'attività a quella successiva oppure utilizzare i dati di un'attività per modificare il comportamento del decisore in base ai risultati di un'attività.

In questo tutorial, questi due tipi di evento sono i più importanti per definire il comportamento del flusso di lavoro. Tuttavia, un'attività può generare eventi diversi `ActivityTaskCompleted` da `ActivityTaskFailed` e `ActivityTaskTimedOut` per l'`WorkflowExecutionCompleted` evento, che

verrà generato quando Amazon SWF elabora la `complete_workflow_execution` chiamata che effettuiamo quando esauriamo le attività da eseguire.

```
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
```

Avvio dell'esecuzione di flusso di lavoro

Per consentire al flusso di lavoro di eseguire il polling dei task di decisione, dobbiamo dapprima avviare l'esecuzione di flusso di lavoro.

Per avviare l'esecuzione del flusso di lavoro, chiama `start_execution` sul tipo di flusso di lavoro registrato (). [AWS::SimpleWorkflow::WorkflowType](#) Per utilizzare il membro di istanza `workflow_type` che abbiamo recuperato nel costruttore della classe, definiremo un piccolo wrapper.

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

Una volta che il flusso di lavoro è in esecuzione, gli eventi di decisione cominceranno a apparire nell'elenco di task corrispondente, che viene passato come opzione dell'esecuzione di flusso di lavoro in [start-execution](#).

A differenza delle opzioni che vengono fornite quando il tipo di flusso di lavoro è registrato, le opzioni passate a `start_execution` non sono considerate come facenti parte del tipo di flusso di lavoro. Sei libero di modificarle per ogni esecuzione di flusso di lavoro senza dover cambiare la versione del flusso di lavoro.

Poiché vorremmo che il flusso di lavoro iniziasse l'esecuzione quando eseguiamo il file, aggiungiamo del codice che istanzi la classe e poi chiami il `start_execution` metodo che abbiamo appena definito.

```
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
```

Per evitare qualsiasi conflitto di denominazione negli elenchi di task, mediante `SecureRandom.uuid` genereremo un UUID aleatorio che possiamo utilizzare come nome di elenco di task. In questo modo, garantiremo l'utilizzo di un nome di elenco di task differente per ogni esecuzione di flusso di lavoro.

Note

Gli elenchi di task sono utilizzati per registrare eventi relativi a un'esecuzione di flusso di lavoro. Se quindi utilizzi lo stesso elenco di task per molteplici esecuzioni dello stesso tipo di flusso di lavoro, puoi ottenere eventi generati durante un'esecuzione precedente, soprattutto se le esegui quasi in successione, condizione piuttosto frequente quando provi nuovo codice o effettui dei test.

Per evitare il problema di dover gestire elementi di esecuzioni precedenti, possiamo utilizzare un nuovo elenco di task per ogni esecuzione, definendolo quando iniziamo l'esecuzione di flusso di lavoro.

Il codice fornisce anche istruzioni alla persona responsabile dell'esecuzione (tu nella maggior parte dei casi) e la versione di "attività" dell'elenco di task. Il decisore utilizzerà queste nome di elenco di task per pianificare attività per il flusso di lavoro, mentre l'implementazione di attività rimarrà in attesa di eventi di attività su questo nome di elenco di task per sapere quando iniziare le attività pianificate e per fornire aggiornamenti sull'esecuzione dell'attività.

Il codice attende inoltre che l'utente inizi l'esecuzione dello starter di attività prima dell'esecuzione di flusso di lavoro, di modo che lo starter sia in grado di rispondere quando i task di attività cominciano a apparire nell'elenco di task fornito.

Fasi successive

L'implementazione del flusso di lavoro è completata. Successivamente, definirai le attività e uno starter di attività nella sezione [Terza parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione delle attività](#).

Terza parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione delle attività

In questa parte del tutorial, implementeremo ogni attività nel nostro flusso di lavoro, cominciando con una classe di base che fornisce alcune funzionalità comuni per il codice delle attività.

Argomenti

- [Definizione di un tipo di attività di base](#)
- [Definendo GetContactActivity](#)
- [Definizione SubscribeTopicActivity](#)
- [Definizione WaitForConfirmationActivity](#)
- [Definizione SendResultActivity](#)
- [Fasi successive](#)

Definizione di un tipo di attività di base

Quando abbiamo progettato il flusso di lavoro, abbiamo identificato le seguenti attività:

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

Ora implementeremo ognuna di queste attività. Poiché le nostre attività condivideranno alcune funzionalità, facciamo un po' di lavoro di base e creiamo del codice comune che possano condividere. Lo chiameremo `BasicActivity` e lo definiremo in un nuovo file chiamato `basic_activity.rb`.

Come con gli altri file di origine, includeremo `utils.rb` per accedere alla funzione `init_domain` con cui configureremo l'esempio di dominio.

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

Successivamente, dichiareremo la classe di attività di base e alcuni dati comuni che ci interessano per ogni attività. Salveremo l'[AWS::SimpleWorkflow::ActivityType](#)istanza, il nome e i risultati dell'attività negli attributi della classe.

```
class BasicActivity

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

Questi attributi accedono ai dati dell'istanza definiti nel `initialize` metodo della classe, che richiede un nome di attività e una versione e una mappa opzionali di opzioni da utilizzare per la registrazione dell'attività con 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
```

```
end
```

Come con la registrazione del tipo di flusso di lavoro, se un tipo di attività è già registrato, possiamo recuperarlo esaminando la raccolta [workflow_types](#) del dominio. Se il tipo di attività è introvabile, verrà registrato.

Inoltre, come con i tipi di flusso di lavoro, puoi impostare delle opzioni di default che sono archiviate insieme al tipo di attività alla registrazione dello stesso.

L'ultima cosa di cui necessita la nostra attività di base è un modo di esecuzione coerente. A questo proposito, definiremo un metodo `do_activity` che accetta un task di attività. Come mostrato di seguito, possiamo utilizzare il task di attività passato per ricevere dati via il relativo attributo di istanza `input`.

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

Questo conclude la lezione. BasicActivity. Ora la utilizzeremo per definire le attività in modo semplice e coerente.

Definendo GetContactActivity

La prima attività che viene eseguita durante l'esecuzione di un flusso di lavoro è `get_contact_activity` quella di recuperare le informazioni sull'abbonamento all'argomento Amazon SNS dell'utente.

Crea un nuovo file chiamato `get_contact_activity.rb` e richiedi entrambi `yaml`, che useremo per preparare una stringa da passare ad Amazon SWF e `basic_activity.rb` che useremo come base per questa `GetContactActivity` classe.

```
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
```

Poiché abbiamo inserito il codice di registrazione dell'attività `BasicActivity`, il `initialize` metodo `GetContactActivity` è piuttosto semplice. È sufficiente chiamare il costruttore della classe di base con il nome dell'attività, ovvero `get_contact_activity`. Questo è tutto quello che dobbiamo fare per registrare la nostra attività.

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

Definiremo ora il `do_activity` metodo, che richiede il numero di and/or telefono e-mail dell'utente.

```
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
    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
```

Alla fine di `do_activity`, inseriamo l'indirizzo e-mail e il numero di telefono dell'utente in una mappa e utilizziamo `to_yaml` per convertire l'intera mappa in una stringa YAML. C'è una ragione importante per questo: tutti i risultati che passi ad Amazon SWF quando completi un'attività devono essere solo dati di stringa. La capacità di Ruby di convertire facilmente oggetti in stringhe YAML e quindi di riconvertirle in oggetti è particolarmente utile in questo caso.

L'implementazione di `get_contact_activity` è terminata. Utilizzeremo questi dati in seguito nell'implementazione di `subscribe_topic_activity`.

Definizione SubscribeTopicActivity

Ora approfondiremo Amazon SNS e creeremo un'attività che utilizza le informazioni generate `get_contact_activity` da per iscrivere l'utente a un argomento di Amazon SNS.

Crea un nuovo file denominato `subscribe_topic_activity.rb`, aggiungi gli stessi requisiti utilizzati per `get_contact_activity`, dichiara la classe e fornisci il relativo metodo `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
```

Ora che abbiamo il codice per configurare e registrare l'attività, aggiungeremo del codice per creare un argomento Amazon SNS. Per farlo, useremo il metodo [AWS::SNS::Clientcreate_topic](#) dell'oggetto.

Aggiungi il `create_topic` metodo alla tua classe, che accetta un oggetto client Amazon SNS passato.

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

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

  return topic_arn
end
```

Una volta ottenuto l'Amazon Resource Name (ARN) dell'argomento, possiamo utilizzarlo con il metodo `set_topic_attributes` [del](#) client Amazon SNS per impostare l'argomento, necessario per l'invio `DisplayNamed` messaggi SMS con Amazon SNS.

Infine, definiremo il metodo `do_activity`. Inizieremo con il raccogliere tutti i dati passati via l'opzione `input` durante la pianificazione dell'attività. Come menzionato in precedenza, questi dati devono essere passati come stringa creata utilizzando `to_yaml`. Dopo il recupero dei dati, utilizzeremo `YAML.load` per convertirli in oggetti Ruby.

Di seguito è illustrato l'inizio di `do_activity`, che ci consente di recuperare i dati di `input`.

```
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))
```

Nel caso non ricevessimo alcun input, non ci sarebbe molto da fare. Dovremmo semplicemente generare un errore nell'attività.

Supponendo che tutto vada bene, tuttavia, continueremo a compilare il nostro `do_activity` metodo, procureremo un client Amazon SNS con e lo passeremo al `create_topic` nostro metodo per creare AWS SDK per Ruby l'argomento 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
```

A questo punto, è importante considerare quanto segue:

- Utilizziamo `AWS.config.with` per impostare la regione per il nostro client Amazon SNS. Poiché intendiamo inviare messaggi SMS, utilizziamo la regione che supporta il servizio SMS dichiarata in `utils.rb`.

- Salviamo l'ARN dell'argomento nella mappa `activity_data`. Questi sono alcuni dei dati che saranno passati all'attività successiva nel flusso di lavoro.

Infine, questa attività iscrive l'utente all'argomento Amazon SNS, utilizzando gli endpoint trasmessi (e-mail e SMS). L'utente non deve immettere entrambi gli endpoint, ma almeno uno.

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

[AWS::SNS::Client.subscribe](#) tratta l'argomento ARN, il protocollo (che, abilmente, abbiamo mascherato da chiave `activity_data` della mappa per l'endpoint corrispondente).

Infine, reimpacchettiamo le informazioni per l'attività successiva in formato YAML, in modo da poterle rispedire ad 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
```

L'implementazione di `subscribe_topic_activity` è terminata. Nella sezione successiva, definiremo `wait_for_confirmation_activity`.

Definizione WaitForConfirmationActivity

Una volta che un utente si è iscritto a un argomento di Amazon SNS, dovrà comunque confermare la richiesta di abbonamento. In questo caso, attendiamo la conferma dell'utente via e-mail o SMS.

L'attività che attende la conferma della sottoscrizione da parte dell'utente è denominata `wait_for_confirmation_activity` e la definiremo in questa sezione. Per prima cosa, crea un nuovo file denominato `wait_for_confirmation_activity.rb` e configuralo come nelle attività precedenti.

```
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
```

Successivamente, inizieremo a definire il metodo `do_activity` e a recuperare i dati di input in una variabile locale denominata `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)
```

Ora che abbiamo l'argomento ARN, possiamo recuperare l'argomento creando una nuova istanza di [AWS::SNS::Topic](#) passandole l'ARN.

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

if topic.nil?
  @results = {
```

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

A questo punto, verificheremo l'argomento per determinare se l'utente ha confermato la sottoscrizione utilizzando uno degli endpoint. Per considerare l'attività come riuscita, è sufficiente che venga confermato un solo endpoint.

Un argomento di Amazon SNS contiene un elenco degli [abbonamenti](#) per quell'argomento e possiamo verificare se l'utente ha confermato o meno un determinato abbonamento controllando se l'ARN dell'abbonamento è impostato su qualcosa di diverso. PendingConfirmation

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

Se otteniamo un ARN per la sottoscrizione, lo registreremo nei dati di risultato dell'attività, lo convertiremo in YAML e restituiremo true da do_activity, a indicare il completamento senza errori dell'attività.

Poiché attendere la conferma di un abbonamento potrebbe richiedere del tempo, occasionalmente ci occuperemo dell'attività da record_heartbeat svolgere. Questo segnala ad Amazon SWF che l'attività è ancora in fase di elaborazione e può essere utilizzato anche per fornire aggiornamenti sullo stato di avanzamento dell'attività (se stai facendo qualcosa, come l'elaborazione di file, per cui puoi segnalare lo stato di avanzamento).

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

Qui si conclude il ciclo `while`. Se per un qualsiasi motivo, il ciclo `while` non riesce, segnaliamo un errore e termineremo il metodo `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
```

L'implementazione di `wait_for_confirmation_activity` risulta completata. Abbiamo quindi una sola attività da definire, ovvero `send_result_activity`.

Definizione SendResultActivity

Se il flusso di lavoro è progredito fino a questo punto, abbiamo correttamente iscritto l'utente a un argomento di Amazon SNS e l'utente ha confermato l'abbonamento.

La nostra ultima attività, `send_result_activity`, invia all'utente la conferma dell'avvenuta sottoscrizione all'argomento, utilizzando l'argomento in questione e l'endpoint con il quale l'utente ha confermato la sottoscrizione.

Crea un nuovo file denominato `send_result_activity.rb` e configuralo come per le altre attività.

```
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
```

Anche il nostro `do_activity` metodo inizia in modo simile, ottenendo i dati di input dal flusso di lavoro, convertendoli da YAML e quindi utilizzando l'argomento ARN per creare un'istanza.

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
```

Una volta che abbiamo l'argomento, vi [pubblicheremo](#) un messaggio (che riprodurremo anche sullo schermo).

```
@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
```

La pubblicazione su un argomento Amazon SNS invia il messaggio fornito a tutti gli endpoint sottoscritti e confermati esistenti per quell'argomento. Di conseguenza, se l'utente ha confermato

la sottoscrizione con un'e-mail e un numero SMS, riceverà due messaggi di conferma, uno a ogni endpoint.

Fasi successive

L'implementazione di `send_result_activity` risulta completata. Nella fase successiva, combinerai tutte queste attività in un'applicazione di attività che gestisce i task di attività e che può avviare attività in risposta, come descritto nella sezione [Quarta parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del poller dei task di attività](#).

Quarta parte del tutorial sul flusso di lavoro di sottoscrizione: implementazione del poller dei task di attività

In Amazon SWF, le attività per l'esecuzione di un flusso di lavoro in esecuzione vengono visualizzate nell'elenco delle attività, fornito quando si pianifica un'attività nel flusso di lavoro.

Implementeremo un poller di attività di base per gestire queste attività per il nostro flusso di lavoro e lo utilizzeremo per avviare le nostre attività quando Amazon SWF inserisce un'attività nell'elenco delle attività per avviare l'attività.

Per prima cosa, crea un nuovo file denominato `swf_sns_activities.rb`, Utilizzeremo questo file per le seguenti operazioni:

- Creare un'istanza della classi di attività che abbiamo generato.
- Registra ogni attività con Amazon SWF.
- Eseguire il polling delle attività e chiamare `do_activity` per ogni attività quando il relativo nome è visualizzato nell'elenco di task di attività.

In `swf_sns_activities.rb`, aggiungi le istruzioni seguenti per richiedere ognuna delle classi di attività definite.

```
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'
```

A questo punto, creeremo la classe e forniremo codice di inizializzazione.

```
class ActivitiesPoller

  def initialize(domain, workflowId)
    @domain = domain
    @workflowId = workflowId
    @activities = {}

    # These are the activities we'll run
    activity_list = [
      GetContactActivity,
      SubscribeTopicActivity,
      WaitForConfirmationActivity,
      SendResultActivity ]

    activity_list.each do | activity_class |
      activity_obj = activity_class.new
      puts "** initialized and registered activity: #{activity_obj.name}"
      # add it to the hash
      @activities[activity_obj.name.to_sym] = activity_obj
    end
  end
end
```

Oltre a salvare il dominio e l'elenco di task passati, questo codice crea un'istanza di ogni classe di attività creata. Poiché ogni classe registra l'attività associata (consulta `basic_activity.rb` se hai bisogno di rivedere il codice), questo è sufficiente per far conoscere ad Amazon SWF tutte le attività che eseguiremo.

Ogni attività per le quale viene creata un'istanza è archiviata in una mappa utilizzando il nome dell'attività (ad esempio `get_contact_activity`) come chiave. In questo modo, potremo cercare facilmente le attività nel codice del poller delle attività, che definiremo in seguito.

Crea un nuovo metodo denominato `poll_for_activities` e chiama [poll](#) su [activity_tasks](#) mantenuti dal dominio per ottenere task di attività.

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

Possiamo ottenere il nome di attività dal membro [activity_type](#) del task. In seguito, utilizzeremo il nome di attività associato a questo task per cercare la classe su cui eseguire `do_activity`, passandolo al task (che include tutti i dati di input da trasferire all'attività).

```
# 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
```

Il codice attende il completamento di `do_activity` e quindi chiama [complete!](#) o [fail!](#) sul task in funzione del codice restituito.

Note

Questo codice esce dal poller una volta avviata l'attività finale, perché ha completato la sua missione e ha avviato tutte le attività. Nel tuo codice Amazon SWF, se le tue attività potrebbero essere eseguite nuovamente, potresti voler mantenere attivo il poller delle attività a tempo indeterminato.

Questa è la fine del codice per la nostra `ActivitiesPoller` classe, ma aggiungeremo altro codice alla fine del file per consentire all'utente di eseguirlo dalla riga di 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 l'utente esegue il file dalla riga di comando (passandogli un elenco di task di attività come primo argomento), questo codice creerà un'istanza della classe del poller e avvierà il polling delle attività. Al termine dell'esecuzione del poller (dopo l'avvio dell'attività finale), dobbiamo soltanto stampare un messaggio e uscire.

Questo è tutto per il poller delle attività. Non ti resta che eseguire il codice e verificarne il funzionamento, come descritto in [Tutorial sul flusso di lavoro di sottoscrizione: esecuzione del flusso di lavoro](#).

Tutorial sul flusso di lavoro di sottoscrizione: esecuzione del flusso di lavoro

Ora che hai completato l'implementazione del flusso di lavoro, delle attività e dei rispettivi poller, sei pronto a eseguire il flusso di lavoro.

Se non l'hai già fatto, dovrà fornire le chiavi di AWS accesso nel `aws-config.txt` file, come [Configurazione della sessione AWS](#) nella parte 1 del tutorial.

Ora, vai alla riga di comando e modificalo nella directory in cui si trovano i file di origine del tutorial. Devi disporre dei seguenti file:

```
.  
|-- 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
```

Ora, avvia il flusso di lavoro con il comando seguente.

```
ruby swf_sns_workflow.rb
```

In questo modo inizia il flusso di lavoro e viene stampato un messaggio con una riga che puoi copiare e incollare in una finestra distinta di riga di comando (o in un altro computer, se vi hai copiato i file di origine del tutorial).

Amazon SWF Example

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

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

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

Press return when you're ready...

Il codice del flusso di lavoro attenderà pazientemente che tu avvii il poller di attività in una finestra separata.

Apri una nuova finestra di riga di comando, modificala nuovamente sulla directory dove si trovano i file di origine, quindi utilizza il comando fornito dal file `swf_sns_workflow.rb` per avviare il poller di attività. Ad esempio, se hai ricevuto l'output precedente, puoi digitare (o incollare) quanto segue.

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

Una volta iniziata, l'esecuzione del poller di attività comincerà a emettere informazioni sulla registrazione delle attività.

```
** 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
```

Ora puoi tornare alla finestra di riga di comando originale e premere return (restituisci) per avviare l'esecuzione del flusso di lavoro. Il comando registra il flusso di lavoro e pianifica la prima attività.

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

Torna all'altra finestra, quella in cui il poller di attività è in esecuzione. Viene mostrato il risultato della prima attività in esecuzione con un prompt per inserire l'indirizzo e-mail o il numero di telefono per gli SMS. Inserisci uno o tutti e due i dati e conferma l'inserimento del testo.

```
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

Il numero di telefono fornito è fittizio ed è utilizzato solo a scopo illustrativo. Utilizza qui il tuo numero di telefono e l'indirizzo e-mail!

Subito dopo aver inserito queste informazioni, riceverai un'e-mail o un messaggio di testo da Amazon SNS in cui ti verrà chiesto di confermare l'abbonamento all'argomento. Se hai inserito un numero per gli SMS, sul telefono vedrai qualcosa dall'aspetto simile.

Would you like to receive messages from SWFSAMPLE? Reply YES SWFSAMPLE to receive messages. Reply HELP or STOP. Msg&data rates may apply.

3:39 PM

Se rispondi a questo messaggio con YES, riceverai la risposta che abbiamo fornito in `send_result_activity`.

SWFSAMPLE> Thanks, you've successfully confirmed registration, and your workflow is complete!

3:39 PM

Mentre succedeva tutto questo, hai visto cosa succedeva nella finestra delle righe di comando? Il flusso di lavoro e i poller di attività sono stati messi a dura prova.

Ecco l'output dal poller del flusso di lavoro.

```
** 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...
```

Ecco l'output dal poller di attività, che stava accadendo nello stesso momento in un'altra finestra delle righe di 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!
```

Complimenti, il flusso di lavoro è completo, come questo tutorial!

Puoi scegliere di eseguire nuovamente il flusso di lavoro per vedere come funzionano i timeout o per inserire dati diversi. Ricorda che una volta che sei sottoscritto a un argomento, sei sottoscritto fino a quando non annulli la sottoscrizione. La riesecuzione del flusso di lavoro prima di annullare l'iscrizione agli argomenti avrà probabilmente esito positivo automaticamente, poiché `wait_for_confirmation_activity` vedrà che l'abbonamento è già confermato.

Per annullare l'iscrizione all'argomento Amazon SNS

- Rispondi in forma negativa (invia STOP) al messaggio di testo.
- Scegli il link per annullare la sottoscrizione che hai ricevuto nell'e-mail.

Ora sei pronto a rieseguire la sottoscrizione all'argomento.

Cosa fare in seguito?

Questo tutorial ha coperto molti argomenti, ma c'è ancora molto altro che puoi imparare su Amazon SWF o Amazon SNS. AWS SDK per Ruby Per ulteriori informazioni e per tanti altri esempi, consulta la documentazione ufficiale per ciascuno dei seguenti:

- [Documentazione AWS SDK per Ruby](#)
- [Documentazione del servizio Amazon Simple Notification](#)
- [Documentazione del servizio Amazon Simple Workflow](#)

Funzionamento nella console Amazon SWF

La console Amazon SWF offre opzioni per configurare, avviare e gestire le esecuzioni dei flussi di lavoro.

Con la console Amazon SWF, puoi:

- Registrare domini di flusso di lavoro.
- Registrare i tipi di flusso di lavoro e i tipi di attività.
- Avvia, visualizza, segnala, annulla, termina e riavvia le esecuzioni dei flussi di lavoro.

Registrazione di un dominio

I flussi di lavoro vengono eseguiti in una AWS risorsa chiamata dominio, che controlla l'ambito del flusso di lavoro. Un AWS account può avere più domini, ognuno dei quali può contenere più flussi di lavoro, ma i flussi di lavoro in domini diversi non possono interagire.

La registrazione del dominio è l'unica funzionalità inizialmente disponibile nella console. Dopo aver registrato almeno un dominio, puoi eseguire le seguenti azioni per il dominio:

- Registra i flussi di lavoro e i tipi di attività.
- Avviare esecuzioni di flusso di lavoro.
- Annullare, terminare e inviare segnali a esecuzioni di flusso di lavoro in corso.
- Riavviare esecuzioni di flusso di lavoro chiuse.

È inoltre possibile eseguire azioni di gestione del dominio, come l'eliminazione e l'eliminazione dei domini obsoleti.

Dopo aver reso obsoleto un dominio, non puoi utilizzarlo per creare nuove esecuzioni di flussi di lavoro o registrare nuovi flussi di lavoro. La deprecazione di un dominio rende obsolette anche tutte le attività e i flussi di lavoro registrati nel dominio. Le esecuzioni avviate prima che il dominio diventasse obsoleto continuano a essere eseguite.

Dopo aver reso obsoleto un dominio precedentemente obsoleto, puoi riprendere a utilizzare il dominio per registrare i tipi di workflow e avviare nuove esecuzioni di workflow.

Per ulteriori informazioni su queste azioni di gestione del dominio, vedere e.

[DeprecateDomain](#)[UndeprecateDomain](#)

Registrazione dei tipi di flusso di lavoro

Puoi registrare i tipi di flusso di lavoro nella console Amazon SWF dopo aver registrato almeno un dominio.

Un tipo di flusso di lavoro è un insieme di tipi di attività che persegono un obiettivo e contengono la logica che coordina le attività. I tipi di workflow coordinano e gestiscono l'esecuzione di attività che possono essere eseguite in modo asincrono su più dispositivi di elaborazione e prevedono metodi di elaborazione sequenziali e paralleli.

Per registrare un tipo di flusso di lavoro Amazon SWF utilizzando la console

1. Apri il dominio in cui desideri registrare un flusso di lavoro.
2. Scegli **Registra**, quindi scegli **Registra flusso di lavoro**.
3. Nella pagina **Registra flusso di lavoro**, inserisci il nome del flusso di lavoro e la versione del flusso di lavoro. Facoltativamente, è anche possibile specificare un [elenco di attività predefinito](#) che verrà utilizzato per pianificare le attività decisionali per l'esecuzione di questo flusso di lavoro.
4. (Facoltativo) Scegliete Opzioni avanzate per specificare i seguenti dettagli per il flusso di lavoro:
 - [Priorità predefinita delle attività](#): la priorità predefinita delle attività da assegnare al flusso di lavoro.
 - [Timeout predefinito di inizio e chiusura dell'esecuzione](#): la durata massima predefinita per le esecuzioni di questo flusso di lavoro.
 - [Timeout predefinito di inizio e chiusura dell'attività](#): la durata massima predefinita delle attività decisionali per questo flusso di lavoro.
 - [Politica secondaria predefinita: la politica](#) predefinita da utilizzare per le esecuzioni dei flussi di lavoro secondari.
 - Ruolo [Lambda predefinito: il ruolo IAM](#) predefinito associato a questo flusso di lavoro.
5. Scegli **Registra flusso di lavoro**.

Registrazione dei tipi di attività

Le attività sono attività che devono essere coordinate ed eseguite dal tipo di flusso di lavoro (ad esempio: verifica dell'ordine del cliente, addebito sulla carta di credito, ecc.). L'ordine in cui vengono eseguite le attività è determinato dalla logica di coordinamento del tipo di flusso di lavoro.

È possibile registrare i tipi di attività dopo aver registrato almeno un dominio.

Per registrare un tipo di attività Amazon SWF utilizzando la console

1. Apri il dominio in cui desideri registrare un'attività.
2. Scegli **Registra**, quindi scegli **Registra attività**.
3. Nella pagina **Registra attività**, inserisci il nome dell'attività e la versione dell'attività.
Facoltativamente, puoi anche specificare un elenco di attività predefinito che verrà utilizzato per pianificare le attività di questa attività.
4. (Facoltativo) Scegliete Opzioni avanzate per specificare i seguenti dettagli per la vostra attività:
 - Priorità predefinita dell'attività: la priorità predefinita dell'attività da assegnare all'attività.
 - Timeout predefinito per la pianificazione delle attività: la durata massima predefinita che un'attività di questa attività può attendere prima di essere assegnata a un lavoratore.
 - Timeout predefinito di inizio e chiusura dell'attività: la durata massima predefinita che un lavoratore può impiegare per elaborare le attività di questa attività.
 - Timeout di pianificazione delle attività predefinito per la chiusura: la durata massima predefinita per un'attività di questa attività.
 - Timeout predefinito dell'operazione: il tempo massimo predefinito prima del quale un lavoratore che elabora un'operazione di questo tipo deve segnalare l'avanzamento chiamando [RecordActivityTaskHeartbeat](#)
5. Scegli **Registra attività**.

Avvio di un flusso di lavoro

Puoi avviare l'esecuzione di un flusso di lavoro dalla console Amazon SWF. Non puoi avviare l'esecuzione di un flusso di lavoro finché non ne hai registrato almeno uno.

Per avviare un'esecuzione utilizzando la console

1. Apri la console Amazon SWF e, nel riquadro di navigazione a sinistra, scegli Domini.
2. Sotto il nome di dominio, scegli Workflows.
3. Nella pagina Flussi di lavoro, scegli il flusso di lavoro che desideri eseguire.
4. Selezionare Start execution (Avvia esecuzione).
5. Nella pagina Avvia esecuzione, inserisci il [nome del flusso di lavoro](#) e l'ID di esecuzione per identificare l'esecuzione con un nome. Facoltativamente, puoi anche specificare un [elenco di attività](#) che verrà utilizzato per le attività decisionali generate per l'esecuzione di questo flusso di lavoro.
6. (Facoltativo) Scegliete Opzioni avanzate per specificare i seguenti dettagli per l'esecuzione del flusso di lavoro:
 - [Priorità delle attività](#): la priorità dell'attività da utilizzare per l'esecuzione di questo flusso di lavoro.
 - [Timeout di inizio e chiusura dell'esecuzione](#): la durata totale dell'esecuzione di questo flusso di lavoro.
 - [Timeout di inizio e chiusura dell'attività](#): la durata massima delle attività decisionali per l'esecuzione di questo flusso di lavoro.
 - [Criterio secondario](#): il criterio da utilizzare per l'esecuzione del workflow secondario dell'esecuzione di questo flusso di lavoro se questa viene interrotta, richiamando l'[TerminateWorkflowExecution](#)azione in modo esplicito o a causa di un timeout scaduto.
 - Ruolo [Lambda: il ruolo](#) IAM da associare all'esecuzione di questo flusso di lavoro.
7. Selezionare Start execution (Avvia esecuzione).

Gestione delle esecuzioni dei flussi di lavoro

È possibile filtrare le esecuzioni del flusso di lavoro per nome, stato, ID e tag. È possibile inviare segnali con input a esecuzioni di workflow attive. Se è necessario annullare o terminare un flusso di lavoro, è possibile utilizzare l'opzione Try-cancel. L'annullamento è preferibile rispetto all'interruzione dell'esecuzione di un flusso di lavoro perché l'annullamento offre al flusso di lavoro l'opportunità di eseguire qualsiasi attività di pulizia e quindi chiuderlo correttamente.

Nella console, è possibile gestire le esecuzioni dei flussi di lavoro attualmente in esecuzione chiuse. and/or

Per gestire le esecuzioni dei flussi di lavoro

1. Apri un dominio per gestirne le esecuzioni del flusso di lavoro.
2. Scegli Find Execution.
3. Nella pagina Esecuzioni del flusso di lavoro, scegli Filtra le esecuzioni per proprietà, quindi in Proprietà scegli uno dei seguenti filtri:

Scegliere	Per applicare questo filtro
Flusso di lavoro	<p>Scegli questo filtro per elencare le esecuzioni di un flusso di lavoro specifico. Ad esempio, per visualizzare le esecuzioni di <code>fiction-books-order-workflow</code>, procedi come segue:</p> <ol style="list-style-type: none"> 1. Scegliete Workflow. 2. In Operatori, scegli Uguali. 3. In Flussi di lavoro, scegli <code>fiction-books-order-workflow</code> 4. (Facoltativo) Scegli Cancella filtri per rimuovere il filtro e iniziare una nuova ricerca di esecuzioni.
Stato	<p>Scegli questo filtro per elencare le esecuzioni con uno stato specifico. Ad esempio, per visualizzare le esecuzioni con lo stato Non riuscito, procedi come segue:</p> <ol style="list-style-type: none"> 1. Scegliete Status. 2. In Operatori, scegli Uguali. 3. In Stati, scegli Non riuscito. 4. (Facoltativo) Scegli Cancella filtri per rimuovere il filtro e iniziare una nuova ricerca di esecuzioni.
ID di esecuzione	<p>Scegli questo filtro per visualizzare l'esecuzione di un flusso di lavoro in base al relativo ID. Ad esempio, per visualizzare l'esecuzione con ID <code>fiction-books-order-category1</code>, procedi come segue:</p> <ol style="list-style-type: none"> 1. Scegliete Execution ID. 2. In Operatori, scegli Uguali. 3. In Esecuzione IDs, scegli <code>fiction-books-order-category 1</code>.

Scegliere	Per applicare questo filtro
	<p>4. (Facoltativo) Scegli Cancella filtri per rimuovere il filtro e iniziare una nuova ricerca di esecuzioni.</p>
Tag	<p>Scegli questo filtro per elencare le esecuzioni con un tag specifico. Ad esempio, per visualizzare le esecuzioni con lo statopurchaseOrder , procedi come segue:</p> <ol style="list-style-type: none"> 1. Scegliete Tag. 2. In Operatori, scegli Uguali. 3. In Tag, scegli PurchaseOrder. 4. (Facoltativo) Scegli Cancella filtri per rimuovere il filtro e iniziare una nuova ricerca di esecuzioni.

4. (Facoltativo) Dopo aver applicato il filtro richiesto per elencare le esecuzioni del flusso di lavoro, è possibile eseguire le seguenti operazioni su un'esecuzione attiva:

- Segnale: utilizzate questa opzione per inviare dati aggiuntivi sull'esecuzione di un workflow in esecuzione. Per farlo:
 1. Scegliete l'esecuzione a cui desiderate inviare dati aggiuntivi.
 2. Scegliete Signal, quindi specificate i dati nella finestra di dialogo di esecuzione del segnale.
 3. Scegliete Signal.
- Try-Cancel: utilizzate questa opzione per provare ad annullare l'esecuzione di un flusso di lavoro. È preferibile annullare un'esecuzione piuttosto che terminarla. L'annullamento offre all'esecuzione la possibilità di eseguire task di pulizia e di chiudersi correttamente.
 1. Scegliete l'esecuzione che desiderate annullare.
 2. Scegli Try-cancel.
- Termina: utilizzate questa opzione per interrompere l'esecuzione di un flusso di lavoro. Ricorda che è preferibile annullare un'esecuzione piuttosto che terminarla.
 1. Scegliete l'esecuzione che desiderate interrompere.
 2. Per la politica sui minori, assicurati che sia selezionata Termina.
 3. (Facoltativo) Specificate il motivo e i dettagli per interrompere l'esecuzione.
 4. Scegliere Terminate (Termina).

5. (Facoltativo) Riesegui: utilizzate questa opzione per eseguire nuovamente un'esecuzione chiusa del workflow.
 1. Nell'elenco delle esecuzioni, sceglie una chiusa da eseguire nuovamente. Quando si seleziona un'esecuzione chiusa, il pulsante Riesegui viene abilitato. Scegli Re-run.
 2. Nella pagina Esecuzione ripeti, specificate i dettagli per l'esecuzione del flusso di lavoro come indicato in. [Avvio di un flusso di lavoro](#)

Concetti di base sul flusso di lavoro in Amazon SWF

Note

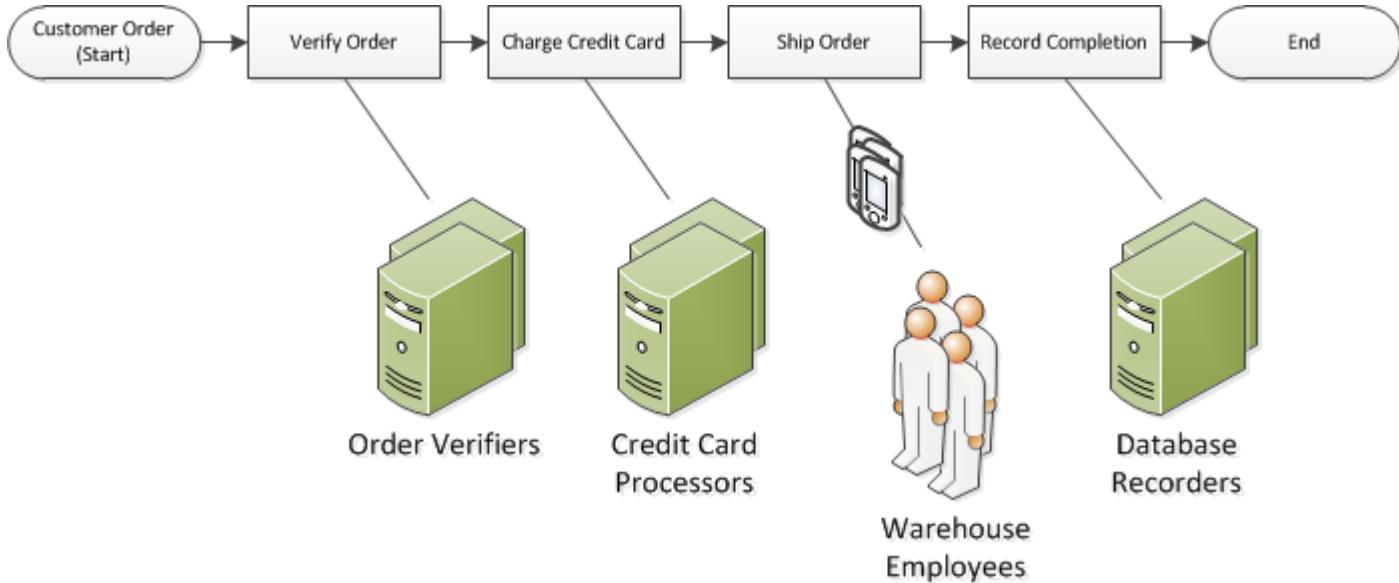
I concetti di questo capitolo forniscono una panoramica di Amazon Simple Workflow Service e ne descrivono le caratteristiche principali. Se stai cercando degli esempi, consulta [Lavorare con Amazon SWF APIs](#).

Utilizzando Amazon Simple Workflow Service (Amazon SWF), puoi implementare applicazioni distribuite e asincrone come flussi di lavoro. I flussi di lavoro consentono di coordinare e gestire l'esecuzione delle attività che possono essere eseguite in modo asincrono in molteplici dispositivi informatici e in grado di realizzare l'elaborazione sequenziale e parallela.

Quando definisci un flusso di lavoro, analizzi l'applicazione per identificarne i task di componente. In Amazon SWF, queste attività sono rappresentate da attività. L'ordine in cui le attività vengono eseguite è determinato dalla logica di coordinamento del flusso di lavoro.

Esempio di flusso di lavoro per un'applicazione di e-commerce

La figura seguente mostra un flusso di lavoro di elaborazione degli ordini di e-commerce che coinvolge sia persone che processi automatizzati:



Il flusso di lavoro dell'applicazione di e-commerce inizia quando un cliente effettua un ordine e include quattro attività:

1. Verifica dell'ordine.
2. Se l'ordine è valido, addebito dell'importo dell'ordine al cliente.
3. Se il pagamento viene effettuato, spedizione dell'ordine.
4. Se l'ordine viene spedito, registrazione dei dettagli relativi all'ordine.

I task in questo flusso di lavoro sono sequenziali: un ordine deve essere verificato prima di poter addebitare una carta di credito, la carta di credito deve essere addebitata prima di poter spedire l'ordine e un ordine deve essere spedito di prima di poter essere registrato. Tuttavia, poiché Amazon SWF supporta processi distribuiti, queste attività possono essere eseguite in luoghi diversi. I task devono essere creati a livello di codice, ma possono essere scritti in linguaggi di programmazione differenti o con strumenti differenti.

Oltre all'elaborazione sequenziale delle attività, Amazon SWF supporta anche flussi di lavoro con elaborazione parallela delle attività. I task paralleli sono eseguiti contemporaneamente e possono essere svolti autonomamente da differenti applicazioni o lavoratori umani. Il flusso di lavoro decide come procedere dopo il completamento di uno o più task paralleli.

Concetti aggiuntivi

- [Creazione di un flusso di lavoro in Amazon SWF](#)
- [Esecuzione di flussi di lavoro in Amazon SWF](#)
- [Cronologia del flusso di lavoro in Amazon SWF](#)
- [Identificatori di oggetti in Amazon SWF](#)
- [Domini in Amazon SWF](#)
- [Attori in Amazon SWF](#)
- [Attività in Amazon SWF](#)
- [Elenchi di attività in Amazon SWF](#)
- [Chiusura dell'esecuzione del flusso di lavoro in Amazon SWF](#)
- [Ciclo di vita di un flusso di lavoro Amazon SWF](#)
- [Sondaggi per le attività in Amazon SWF](#)

Creazione di un flusso di lavoro in Amazon SWF

La creazione di un flusso di lavoro sequenziale di base comporta le fasi elencate di seguito.

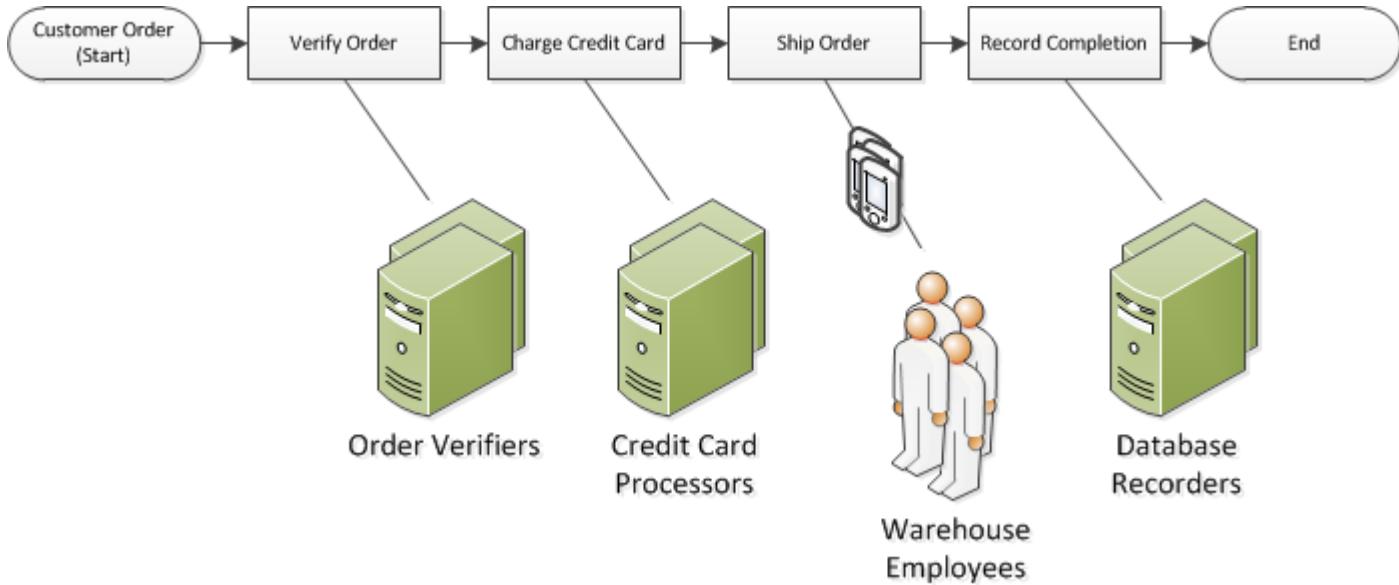
- Modellazione di un flusso di lavoro, registrazione del tipo di flusso di lavoro e registrazione dei relativi tipi di attività.
- Sviluppo e avvio di lavoratori di attività che eseguono task di attività.
- Sviluppo e avvio di decisori che utilizzano la cronologia del flusso di lavoro per determinare le operazioni successive.
- Sviluppo e avvio di starter di flusso di lavoro, ovvero applicazioni che avviano esecuzioni di flusso di lavoro.

Modellazione del flusso di lavoro e delle relative attività

Per utilizzare Amazon SWF, modella i passaggi logici della tua applicazione come attività. Un'attività rappresenta una singola fase logica o un task del flusso di lavoro. Ad esempio, l'autorizzazione di una carta di credito è un'attività che consiste nel fornire un numero di carta di credito e altre informazioni e nel ricevere un codice di approvazione o un messaggio indicante che la carta è stata rifiutata.

Oltre a definire le attività, devi anche definire la logica di coordinamento che gestisce i punti di decisione. Ad esempio, la logica di coordinamento potrebbe pianificare un'attività di follow-up differente a seconda che la carta di credito è stata approvata o rifiutata.

La figura seguente mostra un esempio di flusso di lavoro sequenziale per un ordine cliente con quattro attività (conferma dell'ordine, addebito della carta di credito, spedizione dell'ordine e registrazione del completamento dell'ordine).



Esecuzione di flussi di lavoro in Amazon SWF

Dopo aver progettato la logica di coordinamento e le attività, registri questi componenti come flussi di lavoro e tipi di attività con Amazon SWF. Durante la registrazione, specifichi un nome, una versione e valori di configurazione predefiniti per ogni tipo.

Solo i flussi di lavoro e i tipi di attività registrati possono essere utilizzati con Amazon SWF.

Nell'esempio dell'e-commerce, dovresti registrare il tipo di CustomerOrder flusso di lavoro e i tipi di RecordCompletion attività VerifyOrder ChargeCreditCard ShipOrder,, e.

Dopo la registrazione del tipo di flusso di lavoro, puoi eseguirlo quanto vuoi. Un'esecuzione di flusso di lavoro è un'istanza in esecuzione di un flusso di lavoro.

Un'esecuzione di flusso di lavoro può essere avviata da qualsiasi processo o applicazione e persino da un'altra esecuzione di flusso di lavoro. Nell'esempio di e-commerce, una nuova esecuzione di flusso di lavoro viene avviata a ogni ordine cliente. Il tipo di applicazione che avvia il flusso di lavoro dipende dal modo in cui il cliente effettua l'ordine. Il flusso di lavoro potrebbe essere avviato da un sito Web, un'app per dispositivi mobili oppure da un rappresentante che utilizza un'applicazione aziendale.

Con Amazon SWF, puoi associare un identificatore, chiamato `runId`, alle esecuzioni del flusso di lavoro, in modo da poter integrare gli identificativi aziendali esistenti nel tuo flusso di lavoro. `workflowId` Nell'esempio di e-commerce, ogni esecuzione di flusso di lavoro potrebbe essere identificata utilizzando il numero di fattura del cliente.

Oltre all'identificatore fornito, Amazon SWF associa un identificatore univoco generato dal sistema, `runId`, a ogni esecuzione del flusso di lavoro. `runId`. Amazon SWF consente l'esecuzione di una sola esecuzione di flusso di lavoro con questo identificatore alla volta; sebbene sia possibile avere più esecuzioni di flussi di lavoro dello stesso tipo di flusso di lavoro, ogni esecuzione di flusso di lavoro ha una sua esecuzione distinta. `runId`

Cronologia del flusso di lavoro in Amazon SWF

Amazon SWF registra lo stato di avanzamento di ogni esecuzione del flusso di lavoro nella cronologia del flusso di lavoro: una registrazione dettagliata, completa e coerente di ogni evento che si è verificato dall'inizio dell'esecuzione del flusso di lavoro.

Un evento rappresenta una modifica discreta dello stato di esecuzione del flusso di lavoro, ad esempio la pianificazione di una nuova attività o il completamento di un'attività in corso. La cronologia

del flusso di lavoro contiene tutti gli eventi che causano modifiche allo stato delle esecuzioni del flusso di lavoro, come attività pianificate e completate, timeout dei task e segnali.

In genere, le operazioni che non modificano lo stato dell'esecuzione non compaiono nella cronologia del flusso di lavoro. Ad esempio, la cronologia non mostra i tentativi di polling o l'utilizzo di operazioni di visibilità.

La cronologia del flusso di lavoro presenta diversi vantaggi principali:

- Le applicazioni possono essere prive di stato, poiché tutte le informazioni sull'esecuzione di un flusso di lavoro sono archiviate nella cronologia del flusso di lavoro.
- Per ogni esecuzione del flusso di lavoro, la cronologia funziona da registro delle attività pianificate, dello stato attuale e dei risultati. L'esecuzione del flusso di lavoro utilizza queste informazioni per stabilire i passaggi successivi.
- La cronologia offre un audit trail dettagliato da utilizzare per monitorare le esecuzioni in corso e verificare quelle completate.

Di seguito vi è una visione concettuale della cronologia del flusso di lavoro dell'e-commerce:

```
Invoice0001
  Start Workflow Execution
  Schedule Verify Order
  Start Verify Order Activity
  Complete Verify Order Activity
  Schedule Charge Credit Card
  Start Charge Credit Card Activity
  Complete Charge Credit Card Activity
  Schedule Ship Order
  Start Ship Order Activity
```

Nell'esempio precedente, l'ordine è in attesa di essere spedito. Nell'esempio seguente, l'ordine è completo. Dato che la cronologia del flusso di lavoro è cumulativa, vengono aggiunti gli eventi più recenti:

```
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

A livello di programmazione, gli eventi nella cronologia di esecuzione del workflow sono rappresentati come JavaScript oggetti Object Notation (JSON). La stessa cronologia è una matrice JSON di questi oggetti. Ciascun evento presenta quanto segue:

- Un tipo, ad esempio o [WorkflowExecutionStarted](#)[ActivityTaskCompleted](#)
- Un timestamp in formato Unix
- Un ID che identifica l'evento in modo univoco

Inoltre, ogni tipo di evento ha un set distinto di attributi descrittivi adeguati al tipo. Ad esempio, l'ActivityTaskCompleted evento ha attributi che contengono gli IDs eventi che corrispondono all'ora in cui l'attività è stata pianificata e quando è stata avviata, oltre a un attributo che contiene i dati dei risultati.

È possibile ottenere una copia dello stato corrente della cronologia di esecuzione del flusso di lavoro utilizzando l'[GetWorkflowExecutionHistory](#)azione. Inoltre, nell'ambito dell'interazione tra Amazon SWF e il decisore per il tuo flusso di lavoro, quest'ultimo riceve periodicamente copie della cronologia.

Di seguito una sezione dell'esempio di cronologia dell'esecuzione del flusso di lavoro in 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"
}
]
```

Per un elenco dettagliato dei diversi tipi di eventi che possono comparire nella cronologia di esecuzione del flusso di lavoro, consulta il tipo di [HistoryEvent](#)dati nell'Amazon Simple Workflow Service API Reference.

Amazon SWF archivia la cronologia completa di tutte le esecuzioni del flusso di lavoro per un numero configurabile di giorni dopo la chiusura dell'esecuzione. Questo periodo, noto come periodo di conservazione della cronologia del flusso di lavoro, viene specificato quando registri un dominio per il tuo flusso di lavoro. I domini sono descritti con particolare attenzione più avanti in questa sezione.

Identifieri di oggetti in Amazon SWF

L'elenco seguente descrive come gli oggetti Amazon SWF, come le esecuzioni dei flussi di lavoro, vengono identificati in modo univoco.

- **Tipo di flusso di lavoro:** un tipo di flusso di lavoro registrato viene identificato in base al dominio, al nome e alla versione. I tipi di flusso di lavoro sono specificati nella chiamata a `RegisterWorkflowType`.
- **Tipo di attività:** un tipo di attività registrato è identificato dal dominio, dal nome e dalla versione. I tipi di attività sono specificati nella chiamata a `RegisterActivityType`.

- Attività decisionali e attività: ogni attività decisionale e attività è identificata da un token di attività univoco. Il token dell'attività viene generato da Amazon SWF e viene restituito con altre informazioni sull'attività nella risposta di `PollForDecisionTask` o `PollForActivityTask`. Sebbene il token sia in genere utilizzato dal processo che ha ricevuto il task, può essere passato a un altro processo, il quale può segnalare il completamento o l'esito negativo del task.
- Esecuzione del flusso di lavoro: una singola esecuzione di un flusso di lavoro è identificata dal dominio, dall'ID del flusso di lavoro e dall'ID di esecuzione. I primi due sono parametri che vengono passati a [StartWorkflowExecution](#). L'ID di esecuzione viene restituito da `StartWorkflowExecution`.

Domini in Amazon SWF

I flussi di lavoro vengono eseguiti in una AWS risorsa chiamata dominio che consente di definire l'ambito delle risorse Amazon SWF all'interno del tuo account. AWS Tutti i componenti di un flusso di lavoro, come il tipo di flusso di lavoro e i tipi di attività, devono appartenere a un dominio.

Un AWS account può avere più domini, ognuno dei quali può contenere più flussi di lavoro, ma i flussi di lavoro in domini diversi non possono interagire.

Quando configuri un nuovo flusso di lavoro, prima di impostare uno qualsiasi degli altri componenti devi registrare un dominio, se ancora non lo hai fatto.

Quando registri un dominio, devi specificare un periodo di conservazione della cronologia del flusso di lavoro. Il periodo di conservazione è il periodo di tempo in cui Amazon SWF continuerà a conservare le informazioni sull'esecuzione del flusso di lavoro dopo il completamento dell'esecuzione del flusso di lavoro.

La registrazione del dominio è l'unica funzionalità inizialmente disponibile nella console. Dopo aver registrato almeno un dominio, puoi eseguire le seguenti azioni per il dominio:

- Registra i flussi di lavoro e i tipi di attività.
- Avviare esecuzioni di flusso di lavoro.
- Annullare, terminare e inviare segnali a esecuzioni di flusso di lavoro in corso.
- Riavviare esecuzioni di flusso di lavoro chiuse.

È inoltre possibile eseguire azioni di gestione del dominio, come l'eliminazione e l'eliminazione dei domini obsoleti.

Dopo aver reso obsoleto un dominio, non puoi utilizzarlo per creare nuove esecuzioni di flussi di lavoro o registrare nuovi flussi di lavoro. La deprecazione di un dominio comporta anche l'eliminazione di tutte le attività e i flussi di lavoro registrati nel dominio. Le esecuzioni avviate prima che il dominio diventasse obsoleto continuano a essere eseguite.

Dopo aver reso obsoleto un dominio precedentemente obsoleto, puoi riprendere a utilizzare il dominio per registrare i tipi di workflow e avviare nuove esecuzioni di workflow.

Per ulteriori informazioni su queste azioni di gestione del dominio, consulta e.

[DeprecateDomain](#)[UndeprecateDomain](#)

Attori in Amazon SWF

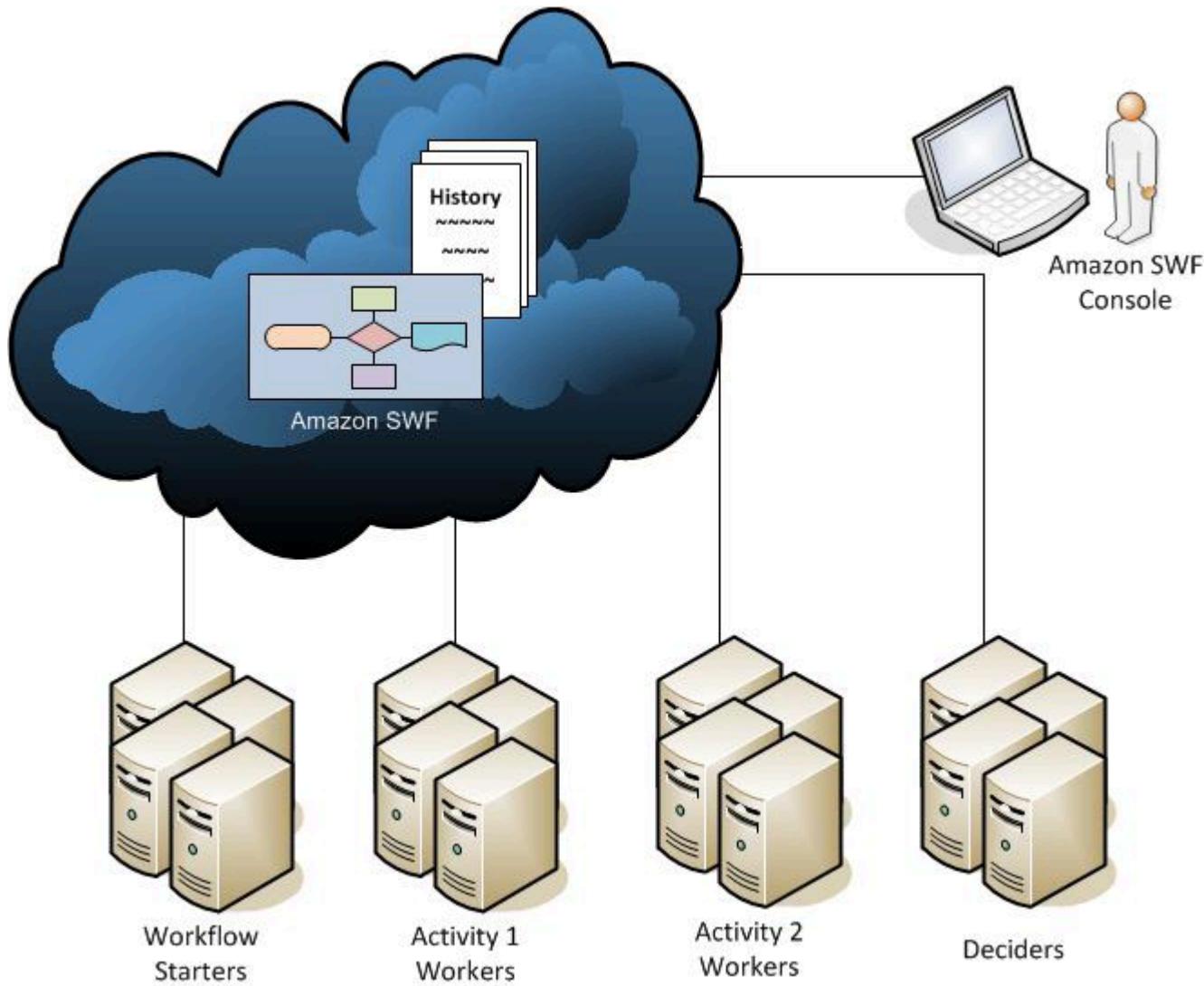
Argomenti

- [Cos'è un attore in Amazon SWF?](#)
- [Starter di flusso di lavoro](#)
- [Decisori](#)
- [Lavoratori di attività](#)
- [Scambio di dati tra attori](#)

Cos'è un attore in Amazon SWF?

Nel corso delle sue operazioni, Amazon SWF interagisce con diversi tipi di attori programmatici. Gli attori possono essere [starter di flusso di lavoro](#), [decisori](#) o [lavoratori di attività](#). Questi attori comunicano con Amazon SWF tramite la sua API. Puoi sviluppare questi attori in qualunque linguaggio di programmazione.

Il diagramma seguente mostra l'architettura Amazon SWF, inclusi Amazon SWF e i suoi attori.



Starter di flusso di lavoro

Uno starter di flusso di lavoro è un'applicazione in grado di avviare le esecuzioni del flusso di lavoro. Nell'esempio di e-commerce, uno starter di flusso di lavoro potrebbe essere il sito Web su cui il cliente effettua l'ordine. Un altro starter di flusso di lavoro potrebbe essere l'applicazione mobile o il sistema utilizzati da un rappresentante per effettuare un ordine a nome del cliente.

Decisori

Un decisore è l'implementazione della logica di coordinamento di un flusso di lavoro. I decisori controllano il flusso dei task di attività in un'esecuzione del flusso di lavoro. Se avviene una modifica durante l'esecuzione del flusso di lavoro, come il completamento di un task, un task di decisione che include l'intera cronologia del flusso di lavoro verrà passato a un decisore. Quando il decisore riceve l'attività decisionale da Amazon SWF, analizza la cronologia di esecuzione del flusso di lavoro

per determinare i passaggi successivi appropriati nell'esecuzione del flusso di lavoro. Chi decide comunica questi passaggi indietro ad Amazon SWF utilizzando le decisioni. Una decisione è un tipo di dati Amazon SWF che può rappresentare diverse azioni successive. Per un elenco delle possibili decisioni, consulta [Decision](#) nel riferimento alle API di Amazon Simple Workflow Service.

Ecco un esempio di decisione in formato JSON, il formato in cui viene trasmessa ad Amazon SWF. Questa decisione pianifica un nuovo task di attività.

```
{  
  "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"  
  }  
}
```

Un decisore riceve un task di decisione quando l'esecuzione del flusso di lavoro si avvia e ogni volta in cui si verifica una modifica di stato nell'esecuzione del flusso di lavoro. I decisori continuano a portare avanti l'esecuzione del flusso di lavoro ricevendo attività decisionali e rispondendo ad Amazon SWF con ulteriori decisioni fino a quando il decisore non stabilisce che l'esecuzione del flusso di lavoro è completa. Risponde quindi con una decisione per chiudere l'esecuzione del flusso di lavoro. Dopo la chiusura dell'esecuzione del flusso di lavoro, Amazon SWF non pianificherà attività aggiuntive per tale esecuzione.

Nell'esempio di e-commerce, il decisore determina se ogni fase sia stata eseguita correttamente e poi pianifica la fase successiva o gestisce le condizioni di errore.

Un decisore rappresenta una singola elaborazione o un singolo thread del computer. Più decisori possono elaborare i task per lo stesso tipo di flusso di lavoro.

Lavoratori di attività

Un lavoratore di attività è un processo o un thread che esegue i task di attività che fanno parte del flusso di lavoro. Il task di attività rappresenta uno dei task che hai identificato nella tua applicazione.

Per utilizzare un'attività nel tuo flusso di lavoro, devi registrarla utilizzando la console Amazon SWF o l'azione [RegisterActivityType](#).

Ogni addetto all'attività interroga Amazon SWF per individuare nuove attività appropriate per l'addetto all'attività; alcune attività possono essere eseguite solo da determinati addetti all'attività. Dopo aver ricevuto un'attività, l'addetto all'attività la elabora fino al completamento, quindi segnala ad Amazon SWF che l'attività è stata completata e fornisce il risultato. Il lavoratore di attività va poi alla ricerca di un nuovo task. I lavoratori di attività associati a un'esecuzione del flusso di lavoro continuano a elaborare i task fino al completamento dell'esecuzione del flusso di lavoro. Nell'esempio di e-commerce, i lavoratori di attività sono processi e applicazioni indipendenti utilizzati dalle persone, come i dipendenti delle aziende che elaborano le transazioni con carta di credito e quelli dei magazzini, che eseguono compiti individuali nel processo.

Un lavoratore di attività rappresenta una singola elaborazione (o thread) del computer. Più decisori di attività possono elaborare i task dello stesso tipo di attività.

Scambio di dati tra attori

I dati di input possono essere forniti a un'esecuzione del flusso di lavoro durante il suo avvio. Analogamente, i dati di input possono essere forniti ai lavoratori di attività quando pianificano i task di attività. Una volta completata un'attività, l'addetto all'attività può restituire i risultati ad Amazon SWF. Analogamente, un decisore può segnalare i risultati di un'esecuzione del flusso di lavoro una volta completata l'esecuzione. Ogni attore può inviare e ricevere dati da Amazon SWF tramite stringhe, il cui formato è definito dall'utente. A seconda delle dimensioni e della sensibilità dei dati, puoi passare i dati direttamente o passare un puntatore ai dati archiviati su un altro sistema o servizio (come Amazon S3 o DynamoDB). Sia i dati trasmessi direttamente che i puntatori ad altri archivi dati vengono registrati nella cronologia di esecuzione del flusso di lavoro; tuttavia, Amazon SWF non copia o memorizza nella cache alcun dato da archivi esterni come parte della cronologia.

Poiché Amazon SWF mantiene lo stato di esecuzione completo di ogni esecuzione del flusso di lavoro, inclusi gli input e i risultati delle attività, tutti gli attori possono essere apolidi. Di conseguenza, l'elaborazione del flusso di lavoro è altamente scalabile. Se il carico sui sistemi aumenta, puoi aggiungere altri attori per aumentare la capacità.

Attività in Amazon SWF

Amazon SWF interagisce con gli addetti alle attività e i decisori assegnando loro incarichi di lavoro noti come attività. Esistono tre tipi di attività in Amazon SWF:

- Attività: un'attività indica a un addetto all'attività di svolgere la sua funzione, ad esempio controllare l'inventario o addebitare una carta di credito. Il task di attività contiene tutte le informazioni di cui il lavoratore di attività necessita per eseguirlo.
- Attività Lambda: un'attività Lambda è simile a un'attività Attività, ma esegue una funzione Lambda anziché un'attività Amazon SWF tradizionale. Per ulteriori informazioni su come definire un task Lambda, consulta [AWS Lambda attività in Amazon SWF](#).
- Attività decisionale: un'attività decisionale indica a un decisore che lo stato di esecuzione del flusso di lavoro è cambiato in modo che il decisore possa determinare l'attività successiva da eseguire. Il task di decisione contiene la cronologia del flusso di lavoro corrente.

Amazon SWF pianifica un'attività decisionale all'avvio del flusso di lavoro e ogni volta che lo stato del flusso di lavoro cambia, ad esempio quando un'attività viene completata. Ogni task di decisione contiene una vista impaginata dell'intera cronologia dell'esecuzione di flusso di lavoro. Il decisore analizza la cronologia di esecuzione del flusso di lavoro e risponde ad Amazon SWF con una serie di decisioni che specificano cosa deve avvenire successivamente nell'esecuzione del flusso di lavoro. In sostanza, ogni attività decisionale offre a chi decide l'opportunità di valutare il flusso di lavoro e fornire indicazioni ad Amazon SWF.

Per garantire che non vengano elaborate decisioni in conflitto, Amazon SWF assegna ogni attività decisionale a un solo decisore e consente a una sola attività decisionale alla volta di essere attiva nell'esecuzione di un flusso di lavoro.

La tabella che segue mostra la relazione tra i differenti costrutti correlati a flussi di lavoro e decisori.

Struttura logica	Registrato come	Eseguito da	Riceve ed esegue	Genera
Flusso di lavoro	Tipo di flusso di lavoro	Decisore	Task di decisione	Decisioni

Quando un addetto all'attività ha completato l'attività, segnala ad Amazon SWF che l'attività è stata completata e include tutti i risultati pertinenti generati. Amazon SWF aggiorna la cronologia di esecuzione del flusso di lavoro con un evento che indica l'attività completata e quindi pianifica un'attività decisionale per trasmettere la cronologia aggiornata al decisore.

Amazon SWF assegna ogni attività a un solo addetto all'attività. Dopo l'attribuzione del task, nessun altro lavoratore di attività può richiederlo o eseguirlo.

La tabella che segue mostra la relazione tra i differenti costrutti correlati alle attività.

Struttura logica	Registrato come	Eseguito da	Riceve ed esegue	Genera
Attività	Tipo di attività	Lavoratore di attività	Task di attività	Dati dei risultati

Elenchi di attività in Amazon SWF

Gli elenchi di task consentono di organizzare i vari task associati a un flusso di lavoro. Puoi considerare questi elenchi come code dinamiche. Quando un'attività è pianificata in Amazon SWF, puoi specificare una coda (elenco delle attività) in cui inserirla. Allo stesso modo, quando esegui un sondaggio su Amazon SWF per un'attività, dici da quale coda (elenco delle attività) ottenere l'attività.

Gli elenchi di task offrono un meccanismo flessibile per instradare task ai lavoratori in funzione del caso d'uso. Questi elenchi sono dinamici in quanto non è necessario registrarli o crearli esplicitamente mediante un'operazione: la pianificazione di un task è sufficiente per creare l'elenco di task se questo non esiste.

Esistono elenchi distinti per i task di attività e i task di decisione. Un task è sempre pianificato in un solo elenco di task e non è condiviso con altri elenchi. Inoltre, come le attività e i flussi di lavoro, gli elenchi di attività sono limitati a una particolare AWS regione e a un dominio Amazon SWF.

Argomenti

- [Elenchi di task di decisione](#)
- [Elenco di task di attività](#)
- [Instradamento di task](#)

Elenchi di task di decisione

Ogni esecuzione di flusso di lavoro è associata a uno specifico elenco di task di decisione. Quando un tipo di flusso di lavoro è registrato ([RegisterWorkflowType](#)azione), puoi specificare un elenco di attività predefinito per le esecuzioni di quel tipo di flusso di lavoro. Quando lo starter di flusso di lavoro avvia l'esecuzione di flusso di lavoro (operazione [StartWorkflowExecution](#)), ha la possibilità di specificare un elenco di task differente per quell'esecuzione di flusso di lavoro.

Quando un decisore esegue il polling di un nuovo task di decisione (operazione [PollForDecisionTask](#)), specifica un elenco di task di decisione in cui cercarlo. Un singolo decisore può servire più esecuzioni di flusso di lavoro chiamando [PollForDecisionTask](#) più volte, utilizzando un elenco di task differente in ogni chiamata, dove ogni elenco di task è specifico a una determinata esecuzione di flusso di lavoro. In alternativa, il decisore può eseguire il polling in un unico elenco di task di decisione che fornisce task di decisione per molteplici esecuzioni di flusso di lavoro. È anche possibile avere più decisori che servono un'unica esecuzione di flusso di lavoro e che eseguono il polling nell'elenco di task per quell'esecuzione.

Elenco di task di attività

Un singolo elenco di task di attività può contenere task di differenti tipi di attività. Le attività sono pianificate nell'elenco delle attività in ordine. Amazon SWF restituisce le attività dall'elenco in ordine con la massima diligenza possibile. In alcuni casi, questo ordine non è rispettato.

Quando un tipo di attività è registrato ([RegisterActivityType](#)azione), puoi specificare un elenco di attività predefinito per quel tipo di attività. Per impostazione predefinita, le attività di questo tipo vengono pianificate nell'elenco di attività specificato; tuttavia, quando chi decide pianifica un'attività ([ScheduleActivityTask](#)decisione), può facoltativamente specificare un elenco di attività diverso in base al quale pianificare l'attività. Se il decisore non specifica un elenco di task, viene utilizzato l'elenco di task di default. Di conseguenza, puoi inserire task di attività in specifici elenchi di task in base agli attributi del task. Ad esempio, potresti inserire tutte le istanze di un task di attività per un determinato tipo di carta di credito in un particolare elenco di task.

Instrandamento di task

Quando un addetto alle attività richiede un sondaggio per una nuova attività ([PollForActivityTask](#)azione), può specificare un elenco di attività da cui attingere. In tal caso, il lavoratore di attività accetterà task soltanto da quell'elenco. In questo modo, hai la certezza che determinati task sono assegnati solo a specifici lavoratori di attività. Ad esempio, potresti creare un

elenco di task contenente task che richiedono l'utilizzo di un computer ad alte prestazioni. Solo i lavoratori di attività in esecuzione sull'hardware appropriato eseguirebbero il polling in quell'elenco di task. Un altro esempio sarebbe di creare un elenco di task per una determinata regione geografica e quindi assicurarti che solo i lavoratori distribuiti in quella regione ricevano quei task. Oppure potresti creare un elenco di task per ordini prioritari e verificare quell'elenco sempre per primo.

Questo modo di assegnare particolari task a specifici lavoratori di attività è denominato instradamento di task. L'instradamento di task è facoltativo; se non specifichi un elenco di task durante la pianificazione di un task di attività, il task viene automaticamente inserito nell'elenco di task di default.

Chiusura dell'esecuzione del flusso di lavoro in Amazon SWF

Una volta avviata, l'esecuzione di un workflow è aperta. Un'esecuzione del flusso di lavoro può essere chiusa come completata, annullata, non riuscita o scaduta. Può anche proseguire come nuova esecuzione o può essere terminata. L'esecuzione di un flusso di lavoro può essere chiusa dal decisore, dalla persona che amministra il flusso di lavoro o da Amazon SWF.

Se il decisore stabilisce che le attività del flusso di lavoro sono finite, chiude l'esecuzione come completata con l'operazione [RespondDecisionTaskCompleted](#) e trasferisce la decisione [CompleteWorkflowExecution](#).

In alternativa, un decisore può chiudere l'esecuzione come annullata o non riuscita. Per annullare l'esecuzione, il decisore utilizza l'operazione `RespondDecisionTaskCompleted` e trasferisce la decisione [CancelWorkflowExecution](#).

Un decisore stabilisce l'esito negativo dell'esecuzione se questa entra in uno stato esterno alla realtà di un normale completamento. Per chiudere con esito negativo l'esecuzione, il decisore utilizza l'operazione `RespondDecisionTaskCompleted` e trasferisce la decisione [FailWorkflowExecution](#).

Amazon SWF monitora le esecuzioni dei flussi di lavoro per garantire che non superino le impostazioni di timeout specificate dall'utente. Se l'esecuzione di un flusso di lavoro scade, Amazon SWF lo chiude automaticamente. Per ulteriori informazioni sui valori di timeout, consulta la sezione [Tipi di timeout di Amazon SWF](#).

Un decisore può anche chiudere l'esecuzione e proseguirla logicamente come nuova esecuzione utilizzando l'operazione `RespondDecisionTaskCompleted` e trasferendo la decisione [ContinueAsNewWorkflowExecution](#). Questa strategia è utile per le esecuzioni di lunga durata per le quali la cronologia può crescere a dismisura nel tempo.

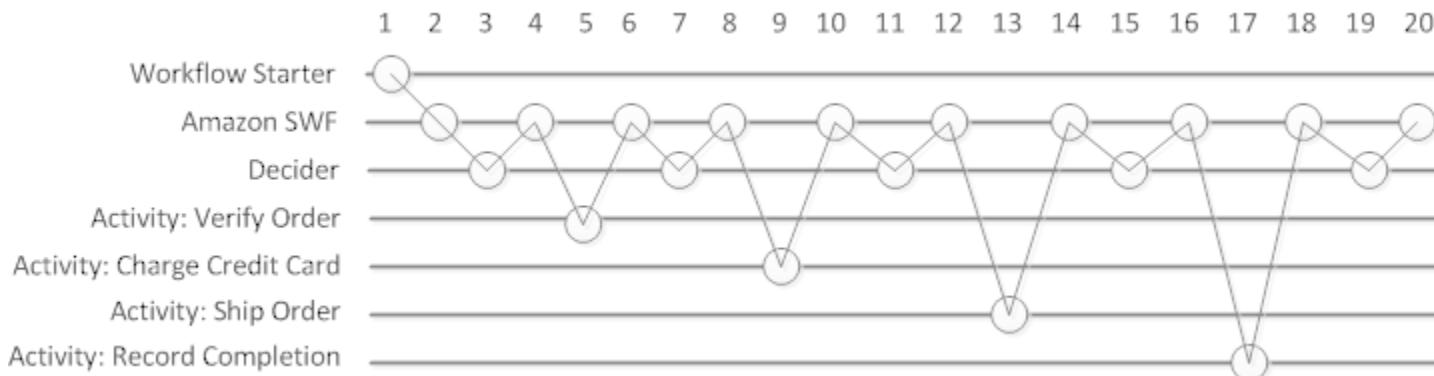
Infine, puoi terminare le esecuzioni dei flussi di lavoro direttamente dalla console Amazon SWF o programmaticamente utilizzando l'API. [TerminateWorkflowExecution](#) La terminazione forza la chiusura dell'esecuzione. L'annullamento è preferibile alla terminazione, perché i decisori possono gestire la chiusura dell'esecuzione del flusso di lavoro.

Amazon SWF interrompe l'esecuzione di un flusso di lavoro se l'esecuzione supera determinati limiti definiti dal servizio. Amazon SWF interrompe un flusso di lavoro secondario se il flusso di lavoro principale è terminato e la politica secondaria applicabile indica che anche il flusso di lavoro secondario deve essere interrotto.

Ciclo di vita di un flusso di lavoro Amazon SWF

Dall'inizio dell'esecuzione di un flusso di lavoro fino al suo completamento, Amazon SWF interagisce con gli attori assegnando loro attività appropriate, attività o attività decisionali.

Il diagramma seguente mostra il ciclo di vita di un'esecuzione di flusso di lavoro relativa all'elaborazione di un ordine dal punto di vista dei componenti utilizzati.



Ciclo di vita di un'esecuzione di flusso di lavoro

Nella tabella seguente viene descritto ogni task nell'immagine precedente.

Descrizione	Operazione, decisione o evento
1. Lo starter del flusso di lavoro richiama l'azione Amazon SWF appropriata per avviare l'esecuzi	Operazione StartWorkflowExecution .

Descrizione	Operazione, decisione o evento
one del flusso di lavoro per un ordine, fornendo le informazioni sull'ordine.	
2. Amazon SWF riceve la richiesta di avvio dell'esecuzione del flusso di lavoro e quindi pianifica la prima attività decisionale.	Eventi WorkflowExecutionStarted e DecisionTaskScheduled .
3. Il decisore riceve l'attività da Amazon SWF, ne esamina la cronologia, applica la logica di coordinamento per determinare che non si sono verificate attività precedenti, decide di pianificare l'attività di Verifica dell'ordine con le informazioni di cui l'addetto all'attività ha bisogno per elaborare l'attività e restituisce la decisione ad Amazon SWF.	Operazione PollForDecisionTask . Operazione RespondDecisionTaskCompleted e decisione ScheduleActivityTask .

Descrizione	Operazione, decisione o evento
4. Amazon SWF riceve la decisione, pianifica l'attività Verify Order e attende il completamento o il timeout dell'attività.	Evento ActivityTaskScheduled .
5. Un lavoratore in grado di eseguire l'attività Verify Order riceve l'attività, la esegue e restituisce i risultati ad Amazon SWF.	Operazioni PollForActivityTask e RespondActivityTaskCompleted .
6. Amazon SWF riceve i risultati dell'attività Verify Order, li aggiunge alla cronologia del flusso di lavoro e pianifica un'attività decisionale.	Eventi ActivityTaskCompleted e DecisionTaskScheduled .

Descrizione	Operazione, decisione o evento
7. Chi decide riceve l'attività da Amazon SWF, ne esamina la cronologia, applica la logica di coordinamento, decide di pianificare ChargeCreditCard un'attività con le informazioni di cui l'addetto all'attività ha bisogno per elaborarla e restituisce la decisione ad Amazon SWF.	Operazione PollForDecisionTask . Operazione RespondDecisionTaskCompleted con decisione ScheduleActivityTask .
8. Amazon SWF riceve la decisione, pianifica l' ChargeCreditCard attività e attende che venga completata o scada.	Eventi DecisionTaskCompleted e ActivityTaskScheduled .
9. Un lavoratore in grado di eseguire l' ChargeCreditCard attività riceve l'attività, la esegue e restituisce i risultati ad Amazon SWF.	Operazioni PollForActivityTask e RespondActivityTaskCompleted .

Descrizione	Operazione, decisione o evento
10. Amazon SWF riceve i risultati dell' ChargeCreditCard attività, li aggiunge alla cronologia del flusso di lavoro e pianifica un'attività decisionale.	Eventi ActivityTaskCompleted e DecisionTaskScheduled .
11. Chi decide riceve l'attività da Amazon SWF, ne esamina la cronologia, applica la logica di coordinamento, decide di pianificare ShipOrder un'attività con le informazioni necessarie all'operatore per eseguire l'attività e restituisce la decisione ad Amazon SWF.	Operazione PollForDecisionTask . RespondDecisionTaskCompleted con decisione ScheduleActivityTask .
12. Amazon SWF riceve la decisione, pianifica un' ShipOrder attività e attende che venga completata o scada.	Eventi DecisionTaskCompleted e ActivityTaskScheduled .

Descrizione	Operazione, decisione o evento
13. Un lavoratore in grado di eseguire l' <code>ShipOrder</code> attività riceve l'attività, la esegue e restituisce i risultati ad Amazon SWF.	Operazioni PollForActivityTask e RespondActivityTaskCompleted .
14. Amazon SWF riceve i risultati dell' <code>ShipOrder</code> attività, li aggiunge alla cronologia del flusso di lavoro e pianifica un'attività decisionale.	Eventi ActivityTaskCompleted e DecisionTaskScheduled .
15. Chi decide riceve l'attività da Amazon SWF, ne esamina la cronologia, applica la logica di coordinamento, decide di pianificare <code>RecordCompletion</code> un'attività con le informazioni necessarie all'operatore per eseguire l'attività e restituisce la decisione ad Amazon SWF.	Operazione PollForDecisionTask . Operazione RespondDecisionTaskCompleted con decisione ScheduleActivityTask .

Descrizione	Operazione, decisione o evento
16. Amazon SWF riceve la decisione, pianifica un'RecordCompletion attività e attende che venga completata o scada.	Eventi DecisionTaskCompleted e ActivityTaskScheduled .
17. Un lavoratore in grado di eseguire l'RecordCompletion attività riceve l'attività, la esegue e restituisce i risultati ad Amazon SWF.	Operazioni PollForActivityTask e RespondActivityTaskCompleted .
18. Amazon SWF riceve i risultati dell'RecordCompletion attività, li aggiunge alla cronologia del flusso di lavoro e pianifica un'attività decisionale.	Eventi ActivityTaskCompleted e DecisionTaskScheduled .

Descrizione	Operazione, decisione o evento
19. Il decisore riceve l'operazione da Amazon SWF, esamina la cronologia, applica la logica di coordinamento, decide di chiudere l'esecuzione del flusso di lavoro e restituisce la decisione insieme agli eventuali risultati ad Amazon SWF.	Operazione PollForDecisionTask . Operazione RespondDecisionTaskCompleted con decisione CompleteWorkflowExecution .
20. Amazon SWF chiude l'esecuzione del flusso di lavoro e archivia la cronologia per riferimenti futuri.	Evento WorkflowExecutionCompleted .

Sondaggi per le attività in Amazon SWF

I decisori e gli addetti alle attività comunicano con Amazon SWF utilizzando sondaggi lunghi. Il decisore o l'addetto alle attività avvia periodicamente una comunicazione con Amazon SWF, notificando ad Amazon SWF la sua disponibilità ad accettare un'attività e quindi specifica un elenco di attività da cui ottenere le attività.

Se un'attività è disponibile nell'elenco delle attività specificato, Amazon SWF la restituisce immediatamente nella risposta. Se non è disponibile alcuna attività, Amazon SWF mantiene la connessione TCP aperta per un massimo di 60 secondi in modo che, se un'attività diventa disponibile durante quel periodo, possa essere restituita nella stessa connessione. Se nessun task diventa disponibile entro 60 secondi, restituisce una risposta vuota e chiude la connessione (Una risposta vuota è una struttura Task in cui il valore di taskToken è una stringa vuota). In tal caso, il decisore o il lavoratore di attività deve ripetere il polling.

Il polling lungo è una soluzione appropriata per l'elaborazione di task voluminosi. Consente a decisori e lavoratori di attività di gestire la relativa capacità ed è facile da utilizzare quando questi sono protetti da un firewall.

Per ulteriori informazioni, consultare [Esecuzione del polling dei task di decisione](#) e [Esecuzione del polling dei task di attività](#).

Concetti avanzati di flusso di lavoro in Amazon SWF

L'esempio dell'e-commerce nella sezione [???](#) rappresenta uno scenario di flusso di lavoro semplificato. In realtà, è preferibile che il flusso di lavoro esegua task simultanei (l'invio di un'e-mail di conferma di un ordine mentre si autorizza una carta di credito), registri gli eventi principali (il confezionamento di tutti gli articoli), aggiorni l'ordine con modifiche (l'aggiunta o la rimozione di un articolo) e prenda altre decisioni più avanzate come parte dell'esecuzione del flusso di lavoro. Questa sezione descrive concetti avanzati di flusso di lavoro che puoi utilizzare per costruire i tuoi flussi di lavoro.

Concetti avanzati

- [Controllo delle versioni](#)
- [Segnali](#)
- [Flussi di lavoro per bambini in Amazon SWF](#)
- [Marcatori in Amazon SWF](#)
- [Tag in Amazon SWF](#)
- [Implementazione di scelte esclusive con Amazon SWF](#)
- [Timer in Amazon SWF](#)
- [Annullamento delle attività in Amazon SWF](#)

Controllo delle versioni

Le esigenze aziendali richiedono spesso che tu abbia implementazioni o variazioni diverse dello stesso flusso di lavoro o attività in esecuzione simultanea. Ad esempio, puoi voler testare una nuova implementazione di un flusso di lavoro mentre un'altra è in produzione. Puoi anche voler eseguire due implementazioni diverse con due diversi set di caratteristiche, ad esempio un'implementazione base e una premium. La funzione Versioni multiple ti permette di eseguire simultaneamente più implementazioni di flussi di lavoro e attività, per qualsiasi scopo che soddisfi i tuoi requisiti.

I tipi di attività e di flusso di lavoro hanno una loro versione associata, che viene specificata al momento della registrazione. Una versione è una stringa in formato libero; puoi scegliere il tuo schema di versioni multiple. Per creare una nuova versione di un tipo registrato, devi registrarla con lo stesso nome e con una versione diversa. [Elenchi di attività in Amazon SWF](#), descritti in precedenza, possono aiutarti meglio a implementare le versioni multiple. Consideriamo una

situazione in cui siano già in corso lunghe esecuzioni di un certo tipo e che le circostanze richiedano che tu riveda il flusso di lavoro, ad esempio con l'aggiunta di una nuova funzione. Puoi implementare la nuova funzione creando nuove versioni dei tipi di attività e dei lavoratori e creando un nuovo decisore. Poi puoi lanciare esecuzioni della nuova versione del flusso di lavoro utilizzando un set diverso di elenchi di task. In questo modo, le diverse versioni delle esecuzioni possono operare contemporaneamente senza che una disturbi l'altra.

Segnali

I segnali permettono di introdurre informazioni in un'esecuzione in corso. In alcuni scenari, puoi voler aggiungere informazioni a un'esecuzione in corso per comunicarle qualche modifica o per informarla di un evento esterno. Qualsiasi processo può inviare un segnale a un'esecuzione aperta. Ad esempio, un'esecuzione può inviare un segnale a un'altra.

Note

In caso di tentativo di invio di un segnale a un'esecuzione di flusso di lavoro non aperta, `SignalWorkflowExecution` non riesce con `UnknownResourceFault`.

Per utilizzare i segnali, definite il nome del segnale e i dati da trasmettere al segnale, se presenti. Quindi, programmate il decisore in modo che riconosca l'evento del segnale ([WorkflowExecutionSignaled](#)) nella cronologia e lo elabori in modo appropriato. Quando un processo desidera segnalare l'esecuzione di un flusso di lavoro, effettua una chiamata ad Amazon SWF (utilizzando l'[SignalWorkflowExecution](#)azione o, nel caso di un decisore, utilizzando la [SignalExternalWorkflowExecution](#)decisione) che specifica l'identificatore per l'esecuzione del flusso di lavoro di destinazione, il nome del segnale e i dati del segnale. Amazon SWF riceve quindi il segnale, lo registra nella cronologia dell'esecuzione del flusso di lavoro di destinazione e pianifica un'attività decisionale per esso. Quando il decisore riceve il task di decisione, anch'esso riceve il segnale nella cronologia dell'esecuzione. Il decisore può così intervenire correttamente in base al segnale e ai suoi dati.

In certi casi, è possibile che si preferisca attendere un segnale. Ad esempio, un utente può annullare un ordine inviando un segnale, ma solo entro un'ora dall'effettuazione dell'ordine. Amazon SWF non dispone di una primitiva che consenta a un decisore di attendere un segnale dal servizio. La funzionalità di pausa deve essere implementata nel decisore stesso. Per effettuare una pausa, il decisore deve avviare un timer mediante la decisione `StartTimer`, la quale specifica l'intervallo

di tempo durante il quale il decisore attende il segnale mentre continua a eseguire il polling di task di decisione. Quando il decisore riceve un task di decisione, deve verificare la cronologia per determinare se un segnale è stato ricevuto o se il timer è stato attivato. Se un segnale è stato ricevuto, il decisore deve annullare il timer. Se invece il timer è stato attivato, significa che il segnale non è arrivato entro l'intervallo di tempo specificato. Per riassumere, se intendi attendere uno specifico segnale, procedi come segue.

1. Crea un timer per il periodo di tempo durante il quale il decisore deve attendere.
2. Quando il decisore riceve un task di decisione, verifica la cronologia per determinare se il segnale è arrivato o se il timer è stato attivato.
3. Se un segnale è arrivato, annulla il timer utilizzando una decisione `CancelTimer` ed elabora il segnale. A seconda del timing, la cronologia può contenere entrambi gli eventi `TimerFired` e `WorkflowExecutionSignaled`. In tal caso, puoi utilizzare l'ordine relativo degli eventi nella cronologia per determinare quale si è verificato per primo.
4. Se il timer è stato attivato prima dell'arrivo di un segnale, si è verificato il timeout del decisore durante l'attesa del segnale. Puoi quindi generare un errore nell'esecuzione o utilizzare qualsiasi altra logica appropriata per il caso d'uso.

Nei casi in cui un flusso di lavoro deve essere annullato, ad esempio l'ordine stesso è stato annullato dal cliente, è necessario utilizzare l'`RequestCancelWorkflowExecution` anziché inviare un segnale al flusso di lavoro.

Alcune applicazioni per i segnali includono quanto segue:

- La messa in pausa delle esecuzioni fino alla ricezione del segnale (ad es., in attesa della spedizione di un inventario).
- La fornitura a un'esecuzione di informazioni che possono influenzare la logica di decisione dei decisori. Ciò è utile per i flussi di lavoro interessati da eventi esterni (ad es. il tentativo di completare la vendita di scorte dopo la chiusura del mercato).
- L'aggiornamento di un'esecuzione se si anticipano le possibili modifiche (ad es. la modifica delle quantità dell'ordine dopo il suo inserimento e prima della spedizione).

Nell'esempio seguente, all'esecuzione del flusso di lavoro viene inviato un segnale per annullare un ordine.

<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 l'esecuzione del flusso di lavoro riceve il segnale, Amazon SWF restituisce una risposta HTTP corretta simile alla seguente. Amazon SWF genererà un'attività decisionale per informare il decisore sull'elaborazione del segnale.

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

Flussi di lavoro per bambini in Amazon SWF

I flussi di lavoro complicati possono essere suddivisi in componenti più piccoli, più gestibili e potenzialmente riutilizzabili usando i flussi di lavoro figli. Un flusso di lavoro figlio è un'esecuzione avviata da un'altra esecuzione (padre). Per avviare un flusso di lavoro figlio, il decisore del flusso di lavoro padre utilizza la decisione `StartChildWorkflowExecution`. I dati di input specificati con questa decisione vengono messi a disposizione del flusso di lavoro figlio tramite la sua cronologia.

Gli attributi della `StartChildWorkflowExecution` decisione specificano anche la policy secondaria, ovvero il modo in cui Amazon SWF deve gestire la situazione in cui l'esecuzione del flusso di lavoro principale termina prima dell'esecuzione del flusso di lavoro secondario. Esistono tre possibili valori:

- **TERMINA**: Amazon SWF interromperà le esecuzioni secondarie.
- **REQUEST_CANCEL**: Amazon SWF tenterà di annullare l'esecuzione secondaria inserendo `WorkflowExecutionCancelRequested` un evento nella cronologia di esecuzione del workflow del figlio.
- **ABBANDONO**: Amazon SWF non intraprenderà alcuna azione; le esecuzioni di bambini continueranno a essere eseguite.

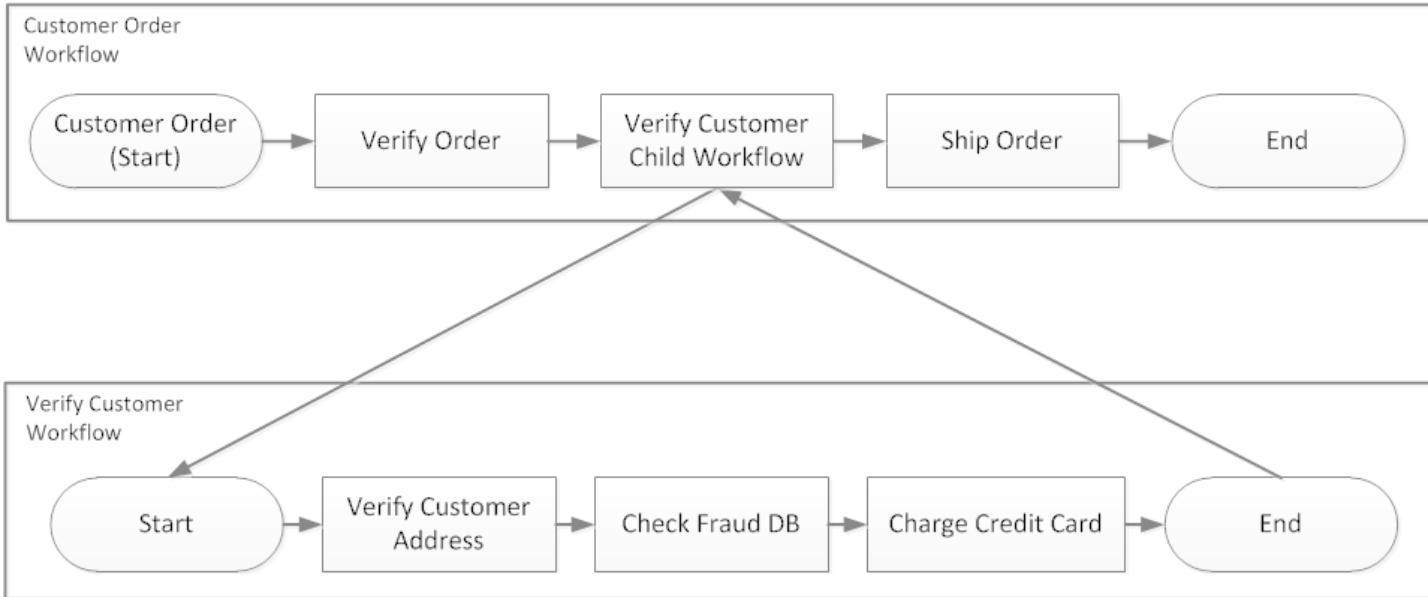
Dopo che viene avviata, l'esecuzione figlio si svolge come un'esecuzione normale. Al termine, Amazon SWF registra il completamento, insieme ai relativi risultati, nella cronologia del flusso di

lavoro dell'esecuzione del flusso di lavoro principale. Tra gli esempi di flussi di lavoro figlio ci sono i seguenti:

- Flusso di lavoro figlio per l'elaborazione di carte di credito, utilizzato dai flussi di lavoro su siti Web diversi
- Flusso di lavoro figlio per l'e-mail, che verifica l'indirizzo e-mail del cliente, controlla l'elenco di chi ha rifiutato, invia l'e-mail e ne verifica il corretto recapito.
- Flusso di lavoro figlio per l'archiviazione e il recupero del database, che combina la connessione, la configurazione, la transazione e la verifica.
- Flusso di lavoro per compilare il codice sorgente, che combina la costruzione, la confezione e la verifica.

Nell'esempio dell'e-commerce, puoi creare un flusso di lavoro figlio per l'attività Addebita carta di credito. Per farlo, puoi registrare un nuovo flusso di lavoro Verifica cliente, registrare le attività Verifica indirizzo del cliente e Controlla database delle frodi e definire la logica di coordinamento dei task. Poi, un decisore nel flusso di lavoro Ordine Cliente può avviare un flusso di lavoro figlio Verifica cliente pianificando la decisione `StartChildWorkflowExecution` che specifica il tipo di flusso di lavoro.

La figura seguente mostra un flusso di lavoro Ordine cliente che comprende un nuovo flusso di lavoro Verifica cliente, che controlla l'indirizzo del cliente e il database della frodi e addebita la carta di credito.



Più flussi di lavoro possono creare esecuzioni figlio utilizzando lo stesso tipo di flusso di lavoro. Ad esempio, il flusso di lavoro figlio Verifica cliente può essere usato anche in altre parti di

un'organizzazione. Gli eventi di un flusso di lavoro figlio sono contenuti nella sua cronologia e non sono inclusi in quella del flusso di lavoro padre.

Poiché i flussi di lavoro figlio sono semplicemente esecuzioni avviate da un decisore, possono essere avviati anche come normali esecuzioni di flussi di lavoro standalone.

Marcatori in Amazon SWF

A volte, puoi voler registrare informazioni nella cronologia di un'esecuzione specifica per il tuo caso d'uso. I contrassegni ti permettono di registrare nella cronologia dell'esecuzione del flusso di lavoro informazioni da utilizzare per scopi personalizzati o specifici di uno scenario.

Per utilizzare i marker, un decisore utilizza la `RecordMarker` decisione, assegna un nome al marker, allega i dati desiderati alla decisione e notifica Amazon SWF utilizzando l'azione.

`RespondDecisionTaskCompleted` Amazon SWF riceve la richiesta, registra il marker nella cronologia del flusso di lavoro e adotta qualsiasi altra decisione nella richiesta. Da quel momento in poi, i decisori possono visualizzare il contrassegno nella cronologia e utilizzarlo in qualunque modo tu lo programmi.

La registrazione di un contrassegno non avvia un task di decisione. Per evitare il blocco dell'esecuzione di flusso di lavoro, deve verificarsi un evento che assicura la continuità dell'esecuzione. Ad esempio, il decisore pianifica un altro task di attività, l'esecuzione di flusso di lavoro riceve un segnale o un task di attività pianificato in precedenza viene completato.

Tra gli esempi di contrassegni ci sono i seguenti:

- Un contatore che conta il numero di loop in un flusso di lavoro ricorsivo.
- I progressi dell'esecuzione del flusso di lavoro in base ai risultati delle attività.
- Un riepilogo delle informazioni a partire dai primi eventi della cronologia.

Nell'esempio dell'e-commerce, puoi voler aggiungere un'attività che verifichi l'inventario ogni giorno e incrementi ogni volta il conteggio di un contrassegno. Puoi quindi aggiungere una logica di decisione che invia e-mail al cliente o avvisa il manager quando il conteggio supera il numero cinque senza dover rivedere tutta la cronologia.

Nell'esempio seguente, il decisore completa un task di decisione e risponde con un'operazione `RespondDecisionTaskCompleted` che contiene una decisione `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 Amazon SWF registra correttamente il marker, restituisce una risposta HTTP corretta simile alla seguente.

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

Tag in Amazon SWF

Amazon SWF supporta l'etichettatura dell'esecuzione di un flusso di lavoro. Questo è particolarmente utile quando si dispone di numerose risorse.

Amazon SWF supporta l'etichettatura di un'esecuzione del flusso di lavoro con un massimo di cinque tag. Ciascun tag è una stringa in formato libero e può essere lungo fino a 256 caratteri. Se vuoi utilizzare i tag, devi assegnarli in fase di avvio di un'esecuzione del flusso di lavoro. Non puoi aggiungere tag a un'esecuzione dopo il suo avvio, né puoi modificare o rimuovere i tag assegnati a un'esecuzione del flusso di lavoro.

IAM supporta il controllo dell'accesso ai domini Amazon SWF in base ai tag. Per controllare gli accessi in base ai tag, devi fornire informazioni sui tag nell'elemento condizione di una policy IAM.

Gestione dei tag

Gestisci i tag di Amazon Simple Workflow Service utilizzando AWS SDKs o interagendo direttamente con l'API Amazon SWF. Utilizzando l'API è possibile aggiungere tag quando si registra un dominio, elencare i tag per un dominio esistente e aggiungere o eliminare i tag per un dominio esistente.

Note

Vi è un limite di 50 tag per ogni risorsa. Per informazioni, consultare [Quote generali degli account per Amazon SWF](#).

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

Per ulteriori informazioni [Lavorare con Amazon SWF APIs](#), consulta [Amazon Simple Workflow Service API Reference](#).

Esecuzioni di flussi di lavoro di tag

Con Amazon SWF, puoi associare tag alle esecuzioni dei flussi di lavoro e quindi eseguire query per le esecuzioni dei flussi di lavoro basate su tali tag. Puoi filtrare l'elenco quando utilizzi le operazioni di visibilità. Selezionando attentamente i tag che assegna a un'esecuzione, puoi usarli per fornire elenchi significativi.

Ad esempio, supponiamo che tu gestisca diversi centri di consegna. Con i tag, puoi elencare i processi che avvengono in un centro logistico specifico. Oppure, se un cliente sta convertendo diversi tipi di file multimediali, i tag potrebbero indicare processi diversi durante la conversione di file video, audio e immagini.

Quando avvii un'esecuzione del flusso di lavoro puoi associarvi fino a cinque tag utilizzando l'operazione `StartWorkflowExecution`, la decisione `StartChildWorkflowExecution` o la decisione `ContinueAsNewWorkflowExecution`. Quando utilizzi le azioni di visibilità per elencare o contare le esecuzioni del flusso di lavoro, puoi filtrare i risultati in base ai tag.

Per utilizzare il tagging

1. Elabora una strategia di tagging. Pensa ai tuoi requisiti aziendali e crea un elenco di tag che ti possano essere utili. Stabilisci i tipi di tag associare alle esecuzioni. Anche se un'esecuzione può ricevere un massimo di cinque tag, la libreria di tag può contenerne un numero illimitato. Poiché un tag può essere qualsiasi valore di stringa e contenere fino a 256 caratteri, può descrivere praticamente ogni concetto aziendale.

2. Quando crei un'esecuzione, associale fino a cinque tag.
3. Elenca o conta le esecuzioni che hanno un determinato tag specificando il parametro `tagFilter` con le operazioni `ListOpenWorkflowExecutions`, `ListClosedWorkflowExecutions`, `CountOpenWorkflowExecutions` e `CountClosedWorkflowExecutions`. L'operazione filtra le esecuzioni in base ai tag specificati.

Quando associ un tag a un'esecuzione del flusso di lavoro, l'associazione è permanente e non può essere rimossa.

Puoi specificare un solo tag nel parametro `tagFilter` con `ListWorkflowExecutions`. Inoltre, la corrispondenza dei tag distingue tra lettere maiuscole e minuscole; solo le corrispondenze esatte restituiscono risultati.

Supponiamo che tu abbia già configurato due esecuzioni taggate nel modo seguente.

Nome esecuzione	Tag assegnati
Esecuzione-Uno	Consumatori, 2011-Febbraio
Esecuzione-Due	Ingrosso, 2011-Marzo

Puoi filtrare l'elenco delle esecuzioni restituito da `ListOpenWorkflowExecutions` per il tag `Consumatori`. I valori `oldestDate` e `latestDate` sono specificati come valori di [tempo Unix](#).

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

Controlla l'accesso ai domini con i tag

Puoi controllare l'accesso ai domini Amazon Simple Workflow Service facendo riferimento ai tag associati ai domini Amazon SWF in IAM.

Ad esempio, puoi limitare i domini Amazon SWF che includono un tag con la chiave `environment` e il valore `production` con la seguente condizione:

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

Per ulteriori informazioni, consultare:

- [Controllo degli accessi tramite tag IAM](#)
- [Policy basate su tag](#)

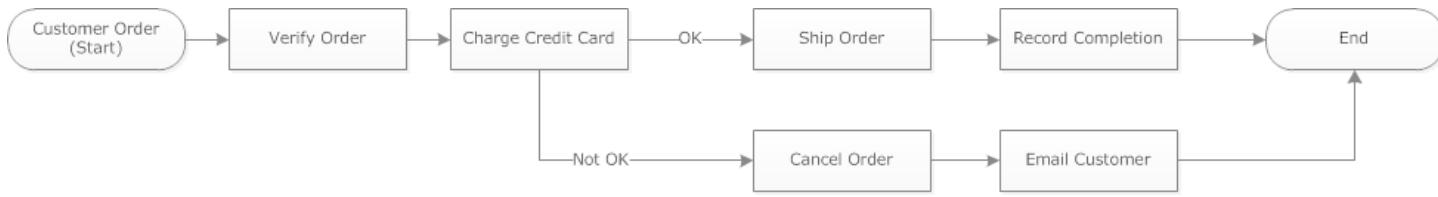
Implementazione di scelte esclusive con Amazon SWF

In alcuni scenari, è possibile che tu intenda pianificare un set di attività differente in funzione del risultato di un'attività precedente. Con lo schema di scelta esclusivo, puoi creare flussi di lavoro flessibili che soddisfano i requisiti complessi della tua applicazione.

Amazon SWF non prevede un'azione di scelta esclusiva specifica. Per implementare la scelta esclusiva, devi scrivere la tua logica decisionale per prendere decisioni basate sui risultati di un'attività precedente. Di seguito sono elencati alcuni casi di utilizzo della scelta esclusiva:

- Esecuzione di attività di pulizia se i risultati dell'attività precedente si sono rivelati insoddisfacenti.
- Pianificazione di attività differenti a seconda che il cliente ha acquistato un piano di base o avanzato.
- Esecuzione di differenti attività di autenticazione a seconda dello storico ordini del cliente.

Nell'esempio di e-commerce, potresti utilizzare la scelta esclusiva per spedire o annullare un ordine in base al risultato dell'addebito della carta di credito. Nella figura seguente, il decisore pianifica i task di attività relativi alla spedizione dell'ordine e alla registrazione del completamento dello stesso se l'addebito della carta di credito viene completato. In caso contrario, pianifica i task di attività relativi alla cancellazione dell'ordine e all'invio di e-mail al cliente.



Il decisore pianifica l'attività `ShipOrder` se l'addebito della carta di credito viene eseguito senza problemi. In caso contrario, il decisore pianifica l'attività `CancelOrder`.

In questo caso, programma il decisore affinché interpreti lo storico e determini se la carta di credito è stata addebitata senza problemi. A tale scopo, potresti utilizzare una logica simile alla seguente:

```

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 l'addebito della carta di credito riesce, il decisore deve rispondere con `RespondDecisionTaskCompleted` per pianificare l'attività `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 l'addebito della carta di credito non riesce, il decisore deve rispondere con `RespondDecisionTaskCompleted` per pianificare l'attività `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 Amazon SWF è in grado di convalidare i dati durante l'RespondDecisionTaskCompleted dazione, Amazon SWF restituisce una risposta HTTP corretta simile alla seguente.

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

Timer in Amazon SWF

Con un timer, puoi avvisare chi decide quando è trascorso un certo periodo di tempo.

Quando risponde a un task di decisione, il decisore ha l'opzione di rispondere con una decisione `StartTimer`. Questa decisione specifica un periodo di tempo dopo il quale viene attivato il timer. Trascorso il tempo specificato, Amazon SWF aggiungerà `TimerFired` un evento alla cronologia di esecuzione del flusso di lavoro e pianificherà un'attività decisionale. Il decisore può quindi utilizzare questa informazione per influenzare ulteriori decisioni. Un'applicazione comune per un timer è quella

di posporre l'esecuzione di un task di attività. Ad esempio, un cliente può preferire che la consegna di un articolo avvenga in ritardo.

Annulloamento delle attività in Amazon SWF

L'annulloamento dell'attività informa chi decide di porre fine alle attività che non devono più essere eseguite. Amazon SWF utilizza un meccanismo di annulloamento cooperativo e non interrompe forzatamente le attività in esecuzione. Devi programmare i lavoratori di attività per gestire le richieste di annulloamento.

Il decisore può decidere di annullare un task di attività mentre sta elaborando un task di decisione. Per annullare un task di attività, il decisore utilizza l'azione `RespondDecisionTaskCompleted` con la decisione `RequestCancelActivityTask`.

Se il task di attività non è stato ancora acquisito da un lavoratore di attività, il servizio annullerà il task. C'è una race condition potenziale per cui un lavoratore di attività può acquisire il task in qualunque momento. Se il task è già stato assegnato a un lavoratore di attività, allora al lavoratore di attività verrà richiesto di annullare il task.

In questo esempio, l'esecuzione di flusso di lavoro riceve un segnale per annullare l'ordine.

```
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 l'esecuzione del flusso di lavoro riceve il segnale, Amazon SWF restituisce una risposta HTTP corretta simile alla seguente. Amazon SWF genererà un'attività decisionale per informare il decisore sull'elaborazione del segnale.

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

Quando il decisore elabora il task di decisione e visualizza il segnale nella cronologia, il decisore tenta di annullare l'attività in sospeso che ha l'ID attività `ShipOrderActivity0001`. L'ID attività viene fornito nella cronologia del flusso di lavoro dall'evento del task di attività pianificato.

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

Se Amazon SWF riceve correttamente la richiesta di annullamento, restituisce una risposta HTTP corretta simile alla seguente:

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

Il tentativo di annullamento viene registrato nella cronologia come l'evento `ActivityTaskCancelRequested`.

Se l'operazione viene annullata con successo, come indicato da un `ActivityTaskCanceled` evento, programmate il decisore in modo che adotti le misure appropriate da seguire dopo l'annullamento dell'attività, come la chiusura dell'esecuzione del flusso di lavoro.

Se l'attività non può essere annullata, ad esempio se l'attività viene completata, ha esito negativo o scade invece di essere annullata, il decisore deve accettare i risultati dell'attività o eseguire le operazioni di pulizia o mitigazione necessarie dal caso d'uso.

Se il task di attività è già stato acquisito da un lavoratore di attività, allora la richiesta di annullamento viene trasmessa attraverso un meccanismo task-heartbeat. Gli addetti all'attività possono utilizzare periodicamente `RecordActivityTaskHeartbeat` per segnalare ad Amazon SWF che l'attività è ancora in corso.

I lavoratori di attività non devono fornire un heartbeat, sebbene sia consigliato per i task di lunga durata. L'annullamento del task richiede una registrazione periodica dell'heartbeat; se il lavoratore non fornisce l'heartbeat, il task non può essere annullato.

Se il decisore richiede l'annullamento dell'attività, Amazon SWF imposta il valore `cancelRequest` dell'oggetto su `true`. L'oggetto `cancelRequest` fa parte dell'oggetto `ActivityTaskStatus` che viene restituito dal servizio in risposta a `RecordActivityTaskHeartbeat`.

Amazon SWF non impedisce il completamento con successo di un'attività di cui è stata richiesta l'annullamento; spetta all'attività determinare come gestire la richiesta di annullamento. In base ai requisiti, programma il lavoratore di attività affinché annulli il task di attività o ignori la richiesta di annullamento.

Se desideri che il lavoratore di attività indichi che il lavoro per il task di attività è stato annullato, programmallo in modo che risponda con `RespondActivityTaskCanceled`. Se desideri che il lavoratore di attività completi il task, programmallo con `RespondActivityTaskCompleted` standard.

Quando Amazon SWF riceve la `RespondActivityTaskCanceled` richiesta `RespondActivityTaskCompleted` or, aggiorna la cronologia di esecuzione del flusso di lavoro e pianifica un'attività decisionale per informare il decisore.

Programma il decisore per elaborare il task di decisione e restituisci le decisioni aggiuntive. Se il task di attività viene annullato correttamente, programma il decisore affinché esegua i task necessari per continuare o chiudere l'esecuzione del flusso di lavoro. Se il task di attività non viene annullato correttamente, programma il decisore affinché accetti o ignori i risultati o programmi la pulizia necessaria.

Sicurezza in Amazon Simple Workflow Service

Questa sezione fornisce informazioni sulla sicurezza e l'autenticazione di Amazon Simple Workflow Service.

Argomenti

- [Protezione dei dati in Amazon Simple Workflow Service](#)
- [Identity and Access Management in Amazon Simple Workflow Service](#)
- [Registrazione e monitoraggio](#)
- [Convalida della conformità per Amazon Simple Workflow Service](#)
- [Resilienza in Amazon Simple Workflow Service](#)
- [Sicurezza dell'infrastruttura in Amazon Simple Workflow Service](#)
- [Analisi della configurazione e delle vulnerabilità in Amazon Simple Workflow Service](#)

Amazon SWF utilizza IAM per controllare l'accesso ad altri AWS servizi e risorse. Per una panoramica sul funzionamento di IAM, vedi [Panoramica della gestione degli accessi](#) nella Guida per l'utente di IAM. Per una panoramica delle credenziali di sicurezza, consultare [Credenziali di sicurezza AWS](#) in Riferimenti generali di Amazon Web Services.

Protezione dei dati in Amazon Simple Workflow Service

Il modello di [responsabilità AWS condivisa modello](#) si applica alla protezione dei dati in Amazon Simple Workflow Service. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Amazon SWF o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia in Amazon Simple Workflow Service

Crittografia a riposo

Amazon SWF crittografa sempre i dati inattivi. I dati in Amazon Simple Workflow Service sono crittografati quando sono inattivi utilizzando una crittografia trasparente lato server. Questo consente di ridurre gli oneri operativi e la complessità associati alla protezione dei dati sensibili. La crittografia dei dati inattivi consente di creare applicazioni sicure che rispettano rigorosi requisiti normativi e di conformità per la crittografia.

Crittografia in transito

Tutti i dati che passano tra Amazon SWF e altri servizi vengono crittografati utilizzando Transport Layer Security (TLS).

Identity and Access Management in Amazon Simple Workflow Service

L'accesso ad Amazon SWF richiede credenziali che AWS possono essere utilizzate per autenticare le tue richieste. Queste credenziali devono disporre delle autorizzazioni per accedere alle AWS risorse, ad esempio recuperare i dati degli eventi da altre risorse. AWS Le seguenti sezioni forniscono dettagli su come utilizzare [AWS Identity and Access Management \(IAM\)](#) e Amazon SWF per proteggere le risorse controllandone l'accesso.

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse Amazon SWF. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso tramite policy](#)
- [Controllo degli accessi](#)
- [Azioni politiche per Amazon SWF](#)
- [Risorse relative alle policy per Amazon SWF](#)
- [Chiavi relative alle condizioni delle politiche per Amazon SWF](#)
- [ACLs in Amazon SWF](#)
- [ABAC con Amazon SWF](#)
- [Utilizzo di credenziali temporanee con Amazon SWF](#)
- [Autorizzazioni principali multiservizio per Amazon SWF](#)
- [Ruoli di servizio per Amazon SWF](#)
- [Ruoli collegati ai servizi per Amazon SWF](#)
- [Politiche basate sull'identità per Amazon SWF](#)
- [Politiche basate sulle risorse all'interno di Amazon SWF](#)
- [Come funziona Amazon Simple Workflow Service con IAM](#)
- [Esempi di policy basate sull'identità per Amazon Simple Workflow Service](#)

- [Principi di base](#)
- [Politiche IAM di Amazon SWF](#)
- [Riepilogo delle API](#)
- [Policy basate su tag](#)
- [Endpoint Amazon VPC per Amazon SWF](#)
- [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Simple Workflow Service](#)

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (vedi [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Simple Workflow Service](#))
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (vedi [Come funziona Amazon Simple Workflow Service con IAM](#))
- Amministratore IAM: scrivi policy per gestire l'accesso (vedi [Esempi di policy basate sull'identità per Amazon Simple Workflow Service](#))

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali Google/ Facebook Per maggiori informazioni sull'accesso, consultare la sezione [Come accedere a Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Si consiglia vivamente di non

utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Cos'è IAM Identity Center?](#) nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un [utente IAM](#) è un'identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e collegandole a identità o risorse. AWS Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate sull'identità possono essere policy in linea (incorporate direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consultare [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: impostano il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in AWS Organizations. Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- Le policy di sessione sono policy avanzate che si passano come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

Controllo degli accessi

Puoi avere credenziali valide per autenticare le tue richieste, ma a meno che tu non disponga delle autorizzazioni non puoi creare o accedere a risorse Amazon SWF. Ad esempio, devi disporre delle autorizzazioni per invocare obiettivi AWS Lambda Amazon Simple Notification Service (Amazon SNS) e Amazon Simple Queue Service (Amazon SQS) associati alle tue regole Amazon SWF.

Le seguenti sezioni descrivono come gestire le autorizzazioni per Amazon SWF. Consigliamo di leggere prima la panoramica.

- [Principi di base](#)
- [Politiche IAM di Amazon SWF](#)
- [Scrittura di politiche per Amazon SWF](#)

Azioni politiche per Amazon SWF

Supporta le operazioni di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di azioni Amazon SWF, consulta [Resources Defined by Amazon Simple Workflow Service nel Service](#) Authorization Reference.

Le azioni politiche in Amazon SWF utilizzano il seguente prefisso prima dell'azione:

```
swf
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

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

Per visualizzare esempi di politiche basate sull'identità di Amazon SWF, consulta [Esempi di policy basate sull'identità per Amazon Simple Workflow Service](#)

Risorse relative alle policy per Amazon SWF

Supporta le risorse relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, utilizzare un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di risorse Amazon SWF e relativi ARNs, consulta [Actions Defined by Amazon Simple Workflow Service nel Service Authorization Reference](#). Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta [Resources Defined by Amazon Simple Workflow Service](#).

Per visualizzare esempi di politiche basate sull'identità di Amazon SWF, consulta. [Esempi di policy basate sull'identità per Amazon Simple Workflow Service](#)

Chiavi relative alle condizioni delle politiche per Amazon SWF

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [conto delle condizioni AWS globali nella Guida per l'utente IAM](#).

Per visualizzare un elenco di chiavi di condizione di Amazon SWF, consulta [Condition Keys for Amazon Simple Workflow Service nel Service Authorization Reference](#). Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Resources Defined by Amazon Simple Workflow Service](#).

Per visualizzare esempi di politiche basate sull'identità di Amazon SWF, consulta. [Esempi di policy basate sull'identità per Amazon Simple Workflow Service](#)

ACLs in Amazon SWF

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Amazon SWF

Supporta ABAC (tag nelle policy): parzialmente

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi, chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con Amazon SWF

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono un accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza temporanee in IAM](#) e [Servizi AWS compatibili con IAM](#) nelle Guida per l'utente IAM.

Autorizzazioni principali multiservizio per Amazon SWF

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso diretto (FAS) utilizzano le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

Ruoli di servizio per Amazon SWF

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per maggiori informazioni, consulta la sezione [Creare un ruolo per delegare le autorizzazioni a una persona Servizio AWS](#) nella Guida per l'utente di IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di Amazon SWF. Modifica i ruoli di servizio solo quando Amazon SWF fornisce indicazioni in tal senso.

Ruoli collegati ai servizi per Amazon SWF

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati in Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Politiche basate sull'identità per Amazon SWF

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte.

Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Esempi di policy basate sull'identità per Amazon SWF

Per visualizzare esempi di politiche basate sull'identità di Amazon SWF, consulta [Esempi di policy basate sull'identità per Amazon Simple Workflow Service](#)

Politiche basate sulle risorse all'interno di Amazon SWF

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli di IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#). I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Come funziona Amazon Simple Workflow Service con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon SWF, scopri quali funzionalità IAM sono disponibili per l'uso con Amazon SWF.

Funzionalità IAM che puoi utilizzare con Amazon Simple Workflow Service

Funzionalità IAM	Supporto per Amazon SWF
Policy basate sull'identità	Sì
Policy basate su risorse	No
Operazioni di policy	Sì
Risorse relative alle policy	Sì

Funzionalità IAM	Supporto per Amazon SWF
Chiavi di condizione della policy (specifica del servizio)	Sì
ACLs	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una visione di alto livello di come Amazon SWF e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, [AWS consulta i servizi che funzionano con IAM](#) nella IAM User Guide.

Esempi di policy basate sull'identità per Amazon Simple Workflow Service

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon SWF. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM \(console\)](#) nella Guida per l'utente IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da Amazon SWF, incluso il formato di ARNs per ogni tipo di risorsa, consulta [Actions, Resources and Condition Keys for Amazon Simple Workflow Service nel Service](#) Authorization Reference.

Argomenti

- [Best practice per le policy](#)
- [Utilizzo della console Amazon SWF](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon SWF nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le politiche AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS. Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consulta [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consulta [Protezione dell'accesso API con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della console Amazon SWF

Per accedere alla console di Amazon Simple Workflow Service, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon SWF presenti nel tuo Account AWS. Se si crea una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, è opportuno concedere l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano continuare a utilizzare la console Amazon SWF, collega anche Amazon **ConsoleAccess** SWF **ReadOnly** AWS o la policy gestita alle entità. Per maggiori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente di IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono correlate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI 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": "*"  
}  
}  
]
```

Principi di base

Il controllo degli accessi di Amazon SWF si basa principalmente su due tipi di autorizzazioni:

- Autorizzazioni per le risorse: a quali risorse Amazon SWF può accedere un utente.

Puoi esprimere autorizzazioni a livello di risorsa solo per i domini.

- Autorizzazioni API: quali azioni Amazon SWF può richiamare un utente.

L'approccio più semplice consiste nel concedere l'accesso completo all'account, richiamando qualsiasi azione Amazon SWF in qualsiasi dominio, o negare completamente l'accesso. Tuttavia, IAM supporta un approccio più granulare al controllo degli accessi che è spesso più utile. È ad esempio possibile:

- Consentire a un utente di eseguire qualsiasi azione Amazon SWF senza restrizioni, ma solo in un dominio specifico. Potresti utilizzare tale policy per consentire alle applicazioni di flusso di lavoro in fase di sviluppo di utilizzare qualsiasi operazione, ma soltanto in un dominio "sandbox".
- Consentire a un utente di accedere a qualsiasi dominio, ma limitando il modo in cui utilizza l'API. Potresti utilizzare tale policy per consentire a un'applicazione "revisore" di chiamare l'API in qualsiasi dominio, ma concedendo unicamente l'accesso in lettura.

- Consentire a un utente di chiamare soltanto un set limitato di operazioni in determinati domini. Potresti utilizzare tale policy per consentire a un starter di flusso di lavoro di chiamare unicamente l'operazione `StartWorkflowExecution` in un determinato dominio.

Il controllo degli accessi di Amazon SWF si basa sui seguenti principi:

- Le decisioni sul controllo degli accessi si basano solo sulle politiche IAM; tutte le verifiche e le manipolazioni delle politiche vengono eseguite tramite IAM.
- Il modello di controllo degli accessi utilizza una deny-by-default policy; qualsiasi accesso non esplicitamente consentito viene negato.
- Puoi controllare l'accesso alle risorse Amazon SWF associando politiche IAM appropriate agli attori del flusso di lavoro.
- Le autorizzazioni a livello di risorsa possono essere espresse solo per i domini.
- Puoi limitare ulteriormente l'utilizzo di alcune operazioni applicando condizioni a uno o più parametri.
- Se concedi l'autorizzazione all'uso [RespondDecisionTaskCompleted](#), puoi esprimere le autorizzazioni per l'elenco delle decisioni incluse in quell'azione.

Ogni decisione comporta uno o più parametri, come una chiamata API standard. Affinché le policy siano le più chiare possibili, puoi esprimere delle autorizzazioni a livello delle decisioni come se fossero chiamate API effettive, inclusa l'applicazione di condizioni a certi parametri. Questi tipi di autorizzazioni sono denominati autorizzazioni pseudo API.

Per un riepilogo dei parametri dell'API classica e della pseudo API che possono essere limitati utilizzando le condizioni, consulta [Riepilogo delle API](#).

Politiche IAM di Amazon SWF

Una policy IAM contiene uno o più Statement elementi, ognuno dei quali contiene un set di elementi che definiscono la policy. Per un elenco completo degli elementi e una discussione generale su come costruire le politiche, vedi [The Access Policy Language](#). Il controllo degli accessi di Amazon SWF si basa sui seguenti elementi:

Effetto

(Obbligatorio) L'effetto dell'istruzione: deny o allow.

Note

È necessario consentire esplicitamente l'accesso; IAM nega l'accesso per impostazione predefinita.

Risorsa

(Obbligatoria) La risorsa, un'entità in un AWS servizio con cui un utente può interagire, a cui si applica l'istruzione.

Puoi esprimere autorizzazioni a livello di risorsa solo per i domini. Ad esempio, una policy può consentire l'accesso soltanto a determinati domini nel tuo account. Per esprimere le autorizzazioni per un dominio, imposta Resource l'Amazon Resource Name (ARN) del dominio, che ha il formato «*Region*arn:aws:swf:: /domain/». *AccountID DomainName Region* è la AWS regione, è l'ID dell'account senza *AccountID* trattini ed è il nome di dominio. *DomainName*

Azione

(Obbligatorio) L'azione a cui si applica l'istruzione, a cui si fa riferimento utilizzando il seguente formato: *serviceId:action*. Per Amazon SWF, imposta su *serviceID*. swf Ad esempio, swf:StartWorkflowExecution si riferisce all'[StartWorkflowExecution](#)azione e viene utilizzato per controllare quali utenti sono autorizzati ad avviare flussi di lavoro.

Se concedi l'autorizzazione all'uso [RespondDecisionTaskCompleted](#), puoi anche controllare l'accesso all'elenco di decisioni incluso utilizzando Action to express le autorizzazioni per la pseudo API. Poiché IAM nega l'accesso per impostazione predefinita, la decisione di chi decide deve essere esplicitamente consentita o non verrà accettata. Puoi utilizzare un valore * per consentire tutte le decisioni.

Condizione

(Facoltativo) Esprime un vincolo su uno o più parametri di un'operazione che limita i valori consentiti.

Le azioni di Amazon SWF hanno spesso un ambito ampio, che puoi ridurre utilizzando le condizioni IAM. Ad esempio, per limitare gli elenchi di attività a cui l'[PollForActivityTask](#)azione è autorizzata a accedere, includi a Condition e utilizza la swf:taskList.name chiave per specificare gli elenchi consentiti.

Puoi esprimere vincoli per le entità elencate di seguito.

- Il tipo di flusso di lavoro. Il nome e la versione hanno chiavi distinte.
- Il tipo di attività. Il nome e la versione hanno chiavi distinte.
- Elenchi di task.
- Tags. Puoi specificare più tag per alcune operazioni. In tal caso, ogni tag ha una chiave distinta.

Note

Per Amazon SWF, i valori sono tutte stringhe, quindi è possibile vincolare un parametro utilizzando un operatore di stringa `comeStringEquals`, che limita il parametro a una stringa specificata. Tuttavia, con gli operatori di confronto di tipo stringa standard `come StringEquals` tutte le richieste devono includere il parametro. Se non includi il parametro in modo esplicito e non esiste alcun valore di default, come l'elenco di task di default specificato durante la registrazione del tipo, l'accesso verrà negato.

È spesso utile considerare le condizioni come facoltative, di modo che sia possibile chiamare un'operazione senza necessariamente includere il parametro associato.

Ad esempio, potresti voler consentire a un decisore di specificare un insieme di [RespondDecisionTaskCompleted](#) decisioni, ma anche consentirgli di specificarne solo una per ogni chiamata particolare. In tal caso, applichi un vincolo ai parametri appropriati utilizzando un operatore `StringEqualsIfExists`, che consente l'accesso se il parametro soddisfa la condizione, ma che non lo nega se il parametro è assente.

Per un elenco completo dei parametri a cui è possibile applicare un vincolo e delle chiavi associate, consulta [Riepilogo delle API](#).

La sezione seguente fornisce esempi di come creare policy di Amazon SWF. Per informazioni dettagliate, consulta la sezione relativa alle [condizioni di stringa](#).

Scrittura di politiche per Amazon SWF

Un flusso di lavoro è composto da più attori: attività, decisori e così via. Puoi controllare l'accesso per ogni attore allegando una policy IAM appropriata.

Con la seguente azione, all'attore verrà concesso l'accesso completo all'account in tutte le regioni:

- Operazione: `swf:*`
- Risorsa: `arn:aws:swf:*:123456789012:/domain/*`

Puoi utilizzare i caratteri jolly per rappresentare più risorse, operazioni o regioni con un singolo valore.

- Il primo carattere jolly (*) nel **Resource** valore indica che le autorizzazioni delle risorse si applicano a tutte le regioni.

Per limitare le autorizzazioni a una singola regione, sostituisci il carattere jolly con la stringa di regione appropriata, ad esempio us-east-1.

- Il secondo carattere jolly (*) nel valore **Resource** consente all'attore di accedere a qualsiasi dominio dell'account nelle regioni specificate.
- Il carattere jolly (*) nel **Action** valore consente all'attore di richiamare qualsiasi azione Amazon SWF.

Per informazioni dettagliate su come utilizzare i caratteri jolly, consulta la sezione relativa alle [descrizioni degli elementi](#).

Autorizzazioni di dominio

Per limitare i flussi di lavoro di un reparto a un determinato dominio, puoi concedere un'autorizzazione che consenta a un attore di eseguire qualsiasi azione, ma solo per un reparto specifico.

Per concedere a un attore l'accesso a più di un dominio, esprimi l'autorizzazione per ogni dominio sotto forma di un elenco di dichiarazioni:

- Operazione: swf:*
- Risorsa: arn:aws:swf:*:123456789012:/domain/department1
- Risorsa: arn:aws:swf:*:123456789012:/domain/department2

Puoi consentire a un attore di utilizzare qualsiasi azione Amazon SWF nei domini department1 and department2. A volte, puoi anche utilizzare i caratteri jolly per rappresentare più domini.

Vincoli e autorizzazioni dell'API

Puoi controllare le azioni che un attore può utilizzare specificando l'azione nell'elemento **Action**.

Con l'azione seguente, un attore può chiamare solo `StartWorkflowExecution` per avviare flussi di lavoro. Non può utilizzare altre azioni.

- Operazione: swf:StartWorkflowExecution

Condizioni

Facoltativamente, puoi limitare i valori dei parametri consentiti dell'azione utilizzando un elemento.

Condition

Per limitare i flussi di lavoro che un attore può avviare, vincola uno o più valori dei `StartWorkflowExecution` parametri nel modo seguente:

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

Un attore con i vincoli precedenti può eseguire solo `version2` di `workflow1` ed entrambi i parametri devono essere inclusi nella richiesta.

Puoi applicare un vincolo a un parametro senza che questo sia incluso in una richiesta utilizzando un operatore `StringEqualsIfExists`, come illustrato di seguito:

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

Un attore con la politica precedente può facoltativamente specificare un elenco di attività quando avvia l'esecuzione di un flusso di lavoro.

Puoi applicare un vincolo a un elenco di tag per alcune operazioni. Ogni tag ha una chiave separata, che puoi usare `swf:tagList.member.0` per vincolare il primo tag dell'elenco, `swf:tagList.member.1` per vincolare il secondo tag dell'elenco e così via, fino a un massimo di 5.

Devi stare attento a come vincoli gli elenchi di tag. Ad esempio, la seguente condizione non è consigliata.

La seguente condizione non è consigliata perché consente di specificare facoltativamente una delle due `some_ok_tag` o `another_ok_tag`. Tuttavia, la Condizione limita solo il primo elemento dell'elenco dei tag. L'elenco potrebbe contenere elementi aggiuntivi con valori arbitrari, tutti consentiti perché la condizione non applica alcuna condizione a `swf:tagList.member.1` `swf:tagList.member.2`, e così via.

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

Un modo per risolvere il problema precedente consiste nel non consentire l'uso di elenchi di tag.

La policy seguente autorizza soltanto some_ok_tag o another_ok_tag in quanto impone la presenza di un solo elemento nell'elenco.

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

Vincoli e autorizzazioni della pseudo API

Per limitare le decisioni disponibili `RespondDecisionTaskCompleted`, devi prima consentire all'attore di chiamare `RespondDecisionTaskCompleted`. Quindi esprimete le autorizzazioni per i membri della pseudo API appropriati utilizzando la stessa sintassi dell'API normale, come segue:

- Dichiarazione 1

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

Operazione: `swf:RespondDecisionTaskCompleted`

- Dichiarazione 2

Risorsa: `*`

Operazione: `swf:ScheduleActivityTask`

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

La prima Statement consente all'attore di chiamare `RespondDecisionTaskCompleted`. La seconda dichiarazione consente all'attore di utilizzare la `ScheduleActivityTask` decisione di indirizzare Amazon SWF a pianificare un'attività. Per consentire tutte le decisioni, sostituisci «`swf:ScheduleActivityTask`» con «`swf: *`».

Per applicare un vincolo ai parametri, puoi utilizzare gli operatori di condizione, esattamente come con l'API classica. L'`StringEquals` operatore dell'esempio precedente `Condition` consente di `RespondDecisionTaskCompleted` pianificare un'attività per l'`SomeActivityType` attività e deve pianificare tale attività. Se intendi consentire a `RespondDecisionTaskCompleted` di utilizzare facoltativamente un valore di parametro, puoi invece utilizzare l'operatore `StringEqualsIfExists`.

AWS politica gestita: `SimpleWorkflowFullAccess`

È possibile allegare la policy `SimpleWorkflowFullAccess` alle identità IAM.

Questa policy fornisce l'accesso completo al servizio di configurazione Amazon SWF.

Limitazioni del modello di servizio sulle politiche IAM

È necessario considerare i vincoli del modello di servizio durante la creazione delle policy IAM. È possibile creare una policy IAM sintatticamente valida che rappresenti una richiesta Amazon SWF non valida; una richiesta consentita in termini di controllo degli accessi può comunque fallire perché non è valida.

Ad esempio, il modello di servizio Amazon SWF non consente l'utilizzo `tagFilter` dei parametri `typeFilter` and nella stessa [ListOpenWorkflowExecutions](#) richiesta. La condizione seguente consentirebbe le chiamate che il servizio rifiuterà, `ValidationException` lanciandole, come richieste non valide:

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

Riepilogo delle API

Questa sezione descrive brevemente come utilizzare le policy IAM per controllare in che modo un attore può utilizzare ciascuna API e pseudo API per accedere alle risorse Amazon SWF.

- Per tutte le operazioni ad eccezione di `RegisterDomain` e `ListDomains`, puoi consentire o negare l'accesso a tutti o a una parte dei domini di un account esprimendo autorizzazioni per la risorsa di dominio.
- Puoi consentire o negare l'autorizzazione per qualsiasi membro dell'API classica e, se concedi l'autorizzazione a chiamare [RespondDecisionTaskCompleted](#), per qualsiasi membro della pseudo API.
- Puoi utilizzare una condizione per applicare un vincolo ai valori consentiti di certi parametri.

Le sezioni seguenti elencano i parametri a cui è possibile applicare un vincolo per ogni membro dell'API classica e della pseudo API nonché le chiavi associate. Indicano inoltre le limitazioni relative al modo in cui è possibile controllare l'accesso ai domini.

API classica

Questa sezione elenca i membri dell'API classica e descrive brevemente i parametri a cui è possibile applicare dei vincoli e le chiavi associate. Indica inoltre le limitazioni relative al modo in cui è possibile controllare l'accesso ai domini.

[CountClosedWorkflowExecutions](#)

- `tagFilter.tag`— Vincolo di stringa. La chiave è `swf:tagFilter.tag`.
- `typeFilter.name`— Vincolo di stringa. La chiave è `swf:typeFilter.name`.
- `typeFilter.version`— Vincolo di stringa. La chiave è `swf:typeFilter.version`.

 Note

`CountClosedWorkflowExecutions` richiede che `typeFilter` e `tagFilter` si escludano a vicenda.

[CountOpenWorkflowExecutions](#)

- `tagFilter.tag`— Vincolo di stringa. La chiave è `swf:tagFilter.tag`.
- `typeFilter.name`— Vincolo di stringa. La chiave è `swf:typeFilter.name`.
- `typeFilter.version`— Vincolo di stringa. La chiave è `swf:typeFilter.version`.

Note

CountOpenWorkflowExecutions richiede che typeFilter e tagFilter si escludano a vicenda.

[CountPendingActivityTasks](#)

- taskList.name— Vincolo di stringa. La chiave è swf:taskList.name.

[CountPendingDecisionTasks](#)

- taskList.name— Vincolo di stringa. La chiave è swf:taskList.name.

[DeleteActivityType](#)

- activityType.name— Vincolo di stringa. La chiave è swf:activityType.name.
- activityType.version— Vincolo di stringa. La chiave è swf:activityType.version.

[DeprecateActivityType](#)

- activityType.name— Vincolo di stringa. La chiave è swf:activityType.name.
- activityType.version— Vincolo di stringa. La chiave è swf:activityType.version.

[DeprecateDomain](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[DeleteWorkflowType](#)

- workflowType.name— Vincolo di stringa. La chiave è swf:workflowType.name.
- workflowType.version— Vincolo di stringa. La chiave è swf:workflowType.version.

[DeprecateWorkflowType](#)

- workflowType.name— Vincolo di stringa. La chiave è swf:workflowType.name.

- `workflowType.version`— Vincolo di stringa. La chiave è `swf:workflowType.version`.

[DescribeActivityType](#)

- `activityType.name`— Vincolo di stringa. La chiave è `swf:activityType.name`.
- `activityType.version`— Vincolo di stringa. La chiave è `swf:activityType.version`.

[DescribeDomain](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[DescribeWorkflowExecution](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[DescribeWorkflowType](#)

- `workflowType.name`— Vincolo di stringa. La chiave è `swf:workflowType.name`.
- `workflowType.version`— Vincolo di stringa. La chiave è `swf:workflowType.version`.

[GetWorkflowExecutionHistory](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[ListActivityTypes](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[ListClosedWorkflowExecutions](#)

- `tagFilter.tag`— Vincolo di stringa. La chiave è `swf:tagFilter.tag`.
- `typeFilter.name`— Vincolo di stringa. La chiave è `swf:typeFilter.name`.
- `typeFilter.version`— Vincolo di stringa. La chiave è `swf:typeFilter.version`.

 Note

`ListClosedWorkflowExecutions` richiede che `typeFilter` e `tagFilter` si escludano a vicenda.

[ListDomains](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[ListOpenWorkflowExecutions](#)

- `tagFilter.tag`— Vincolo di stringa. La chiave è `swf:tagFilter.tag`.
- `typeFilter.name`— Vincolo di stringa. La chiave è `swf:typeFilter.name`.
- `typeFilter.version`— Vincolo di stringa. La chiave è `swf:typeFilter.version`.

 Note

`ListOpenWorkflowExecutions` richiede che `typeFilter` e `tagFilter` si escludano a vicenda.

[ListWorkflowTypes](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[PollForActivityTask](#)

- `taskList.name`— Vincolo di stringa. La chiave è `swf:taskList.name`.

[PollForDecisionTask](#)

- `taskList.name`— Vincolo di stringa. La chiave è `swf:taskList.name`.

[RecordActivityTaskHeartbeat](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[RegisterActivityType](#)

- `defaultTaskList.name`— Vincolo di stringa. La chiave è `swf:defaultTaskList.name`.
- `name`— Vincolo di stringa. La chiave è `swf:name`.
- `version`— Vincolo di stringa. La chiave è `swf:version`.

[RegisterDomain](#)

- `name`— Il nome del dominio da registrare è disponibile come risorsa per questa azione.

[RegisterWorkflowType](#)

- `defaultTaskList.name`— Vincolo di stringa. La chiave è `swf:defaultTaskList.name`.
- `name`— Vincolo di stringa. La chiave è `swf:name`.
- `version`— Vincolo di stringa. La chiave è `swf:version`.

[RequestCancelWorkflowExecution](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[RespondActivityTaskCanceled](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[RespondActivityTaskCompleted](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[RespondActivityTaskFailed](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[RespondDecisionTaskCompleted](#)

- `decisions.member.N`— Limitato indirettamente tramite pseudo autorizzazioni API. Per informazioni dettagliate, vedi [Pseudo API](#).

[SignalWorkflowExecution](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

[StartWorkflowExecution](#)

- `tagList.member.0`— Vincolo di stringa. La chiave è `swf:tagList.member.0`.
- `tagList.member.1`— Vincolo di stringa. La chiave è `swf:tagList.member.1`.
- `tagList.member.2`— Vincolo di stringa. La chiave è `swf:tagList.member.2`.
- `tagList.member.3`— Vincolo di stringa. La chiave è `swf:tagList.member.3`.
- `tagList.member.4`— Vincolo di stringa. La chiave è `swf:tagList.member.4`.
- `taskList.name`— Vincolo di stringa. La chiave è `swf:taskList.name`.
- `workflowType.name`— Vincolo di stringa. La chiave è `swf:workflowType.name`.
- `workflowType.version`— Vincolo di stringa. La chiave è `swf:workflowType.version`.

 Note

Non puoi applicare un vincolo a più di cinque tag.

[TerminateWorkflowExecution](#)

- Non puoi applicare un vincolo ai parametri di questa operazione.

Pseudo API

Questa sezione elenca i membri della pseudo API, che rappresentano le decisioni incluse in [RespondDecisionTaskCompleted](#). Se hai concesso l'autorizzazione a utilizzare `RespondDecisionTaskCompleted`, la tua policy può esprimere autorizzazioni per i membri di questa API come avviene con l'API classica. Puoi limitare ulteriormente alcuni membri della pseudo API impostando condizioni su uno o più parametri. Questa sezione elenca i membri della pseudo API e descrive brevemente i parametri a cui è possibile applicare un vincolo nonché le chiavi associate.

 Note

Le chiavi `aws:SourceIP`, `aws:UserAgent` e `aws:SecureTransport` non sono disponibili per la pseudo API. Se la policy di sicurezza prevista richiede tali chiavi per controllare l'accesso alla pseudo API, puoi utilizzarle con l'operazione `RespondDecisionTaskCompleted`.

CancelTimer

- Non puoi applicare un vincolo ai parametri di questa operazione.

CancelWorkflowExecution

- Non puoi applicare un vincolo ai parametri di questa operazione.

CompleteWorkflowExecution

- Non puoi applicare un vincolo ai parametri di questa operazione.

ContinueAsNewWorkflowExecution

- `tagList.member.0`— Vincolo di stringa. La chiave è `swf:tagList.member.0`.
- `tagList.member.1`— Vincolo di stringa. La chiave è `swf:tagList.member.1`.
- `tagList.member.2`— Vincolo di stringa. La chiave è `swf:tagList.member.2`.
- `tagList.member.3`— Vincolo di stringa. La chiave è `swf:tagList.member.3`.
- `tagList.member.4`— Vincolo di stringa. La chiave è `swf:tagList.member.4`.
- `taskList.name`— Vincolo di stringa. La chiave è `swf:taskList.name`.
- `workflowTypeVersion`— Vincolo di stringa. La chiave è `swf:workflowTypeVersion`.

 Note

Non puoi applicare un vincolo a più di cinque tag.

FailWorkflowExecution

- Non puoi applicare un vincolo ai parametri di questa operazione.

RecordMarker

- Non puoi applicare un vincolo ai parametri di questa operazione.

RequestCancelActivityTask

- Non puoi applicare un vincolo ai parametri di questa operazione.

RequestCancelExternalWorkflowExecution

- Non puoi applicare un vincolo ai parametri di questa operazione.

ScheduleActivityTask

- `activityType.name`— Vincolo di stringa. La chiave è `swf:activityType.name`.
- `activityType.version`— Vincolo di stringa. La chiave è `swf:activityType.version`.
- `taskList.name`— Vincolo di stringa. La chiave è `swf:taskList.name`.

SignalExternalWorkflowExecution

- Non puoi applicare un vincolo ai parametri di questa operazione.

StartChildWorkflowExecution

- `tagList.member.0`— Vincolo di stringa. La chiave è `swf:tagList.member.0`.
- `tagList.member.1`— Vincolo di stringa. La chiave è `swf:tagList.member.1`.
- `tagList.member.2`— Vincolo di stringa. La chiave è `swf:tagList.member.2`.
- `tagList.member.3`— Vincolo di stringa. La chiave è `swf:tagList.member.3`.
- `tagList.member.4`— Vincolo di stringa. La chiave è `swf:tagList.member.4`.
- `taskList.name`— Vincolo di stringa. La chiave è `swf:taskList.name`.
- `workflowType.name`— Vincolo di stringa. La chiave è `swf:workflowType.name`.

- `workflowType.version`— Vincolo di stringa. La chiave è `swf:workflowType.version`.

 Note

Non puoi applicare un vincolo a più di cinque tag.

StartTimer

- Non puoi applicare un vincolo ai parametri di questa operazione.

Policy basate su tag

Amazon SWF supporta politiche basate sui tag. Ad esempio, puoi limitare i domini Amazon SWF che includono un tag con la chiave `environment` e il valore `production` con la seguente condizione:

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

Per ulteriori informazioni sui tag, consulta:

- [Tag in Amazon SWF](#)
- [Controllo degli accessi tramite tag IAM](#)

Endpoint Amazon VPC per Amazon SWF

 Note

AWS PrivateLink il supporto è attualmente disponibile solo nelle regioni AWS Top AWS Secret: Est, Regione segreta e Cina.

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione tra i flussi di lavoro di Amazon VPC e Amazon Simple Workflow Service. Puoi utilizzare questa connessione con i tuoi flussi di lavoro Amazon SWF senza dover accedere alla rete Internet pubblica.

Amazon VPC ti consente di avviare AWS risorse in una rete virtuale personalizzata. Puoi utilizzare un VPC per controllare le impostazioni di rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Per ulteriori informazioni VPCs, consulta la [Amazon VPC User Guide](#).

Per connettere Amazon VPC ad Amazon SWF, devi prima definire un endpoint VPC di interfaccia, che ti permetta di connettere il tuo VPC ad altri. Servizi AWS L'endpoint offre una connettività scalabile e affidabile senza necessità di disporre di un gateway Internet, un'istanza NAT (Network Address Translation) o una connessione VPN. Per ulteriori informazioni, consulta [Endpoint VPC di interfaccia \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon VPC.

Creazione dell'endpoint

Puoi creare un endpoint Amazon SWF nel tuo VPC utilizzando Console di gestione AWS, the AWS Command Line Interface (AWS CLI), un AWS SDK, l'API Amazon SWF oppure. CloudFormation

Per informazioni sulla creazione e sulla configurazione di un endpoint utilizzando la console Amazon VPC o il AWS CLI, consulta la sezione [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di Amazon VPC.

Note

Quando crei un endpoint, specifica Amazon SWF come servizio a cui desideri connettere il tuo VPC. Nella console Amazon VPC, i nomi dei servizi variano in base alla AWS regione. Ad esempio, nella regione AWS Top Secret - East, il nome del servizio per Amazon SWF è com.amazonaws.us-iso-east-1.swf.

Per informazioni sulla creazione e configurazione di un endpoint utilizzando CloudFormation, consulta la VPCEndpoint risorsa [AWS::EC2:](#) nella CloudFormation User Guide.

Politiche degli endpoint Amazon VPC

Per controllare l'accesso alla connettività ad Amazon SWF, puoi allegare una policy per gli endpoint AWS Identity and Access Management (IAM) durante la creazione di un endpoint Amazon VPC. Puoi creare regole IAM complesse collegando più policy per gli endpoint. Per ulteriori informazioni, consultare:

- [Policy degli endpoint di Amazon Virtual Private Cloud per Amazon SWF](#)
- [Controllo dell'accesso ai servizi con endpoint VPC](#)

Policy degli endpoint di Amazon Virtual Private Cloud per Amazon SWF

Puoi creare una policy di endpoint Amazon VPC per Amazon SWF in cui specifichi quanto segue:

- Il principio che può eseguire azioni.
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

L'esempio seguente aggiunge un ruolo IAM specifico a una policy:

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

- Per ulteriori informazioni sulla creazione di policy per gli endpoint, consulta [Controllo dell'accesso ai servizi con endpoint VPC](#).
- Per informazioni su come utilizzare IAM per controllare l'accesso alle tue risorse AWS e a quelle di Amazon SWF, consulta [Identity and Access Management in Amazon Simple Workflow Service](#)

Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Simple Workflow Service

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Amazon SWF e IAM.

Argomenti

- [Non sono autorizzato a eseguire un'azione in Amazon SWF](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon SWF](#)

Non sono autorizzato a eseguire un'azione in Amazon SWF

Se ricevi un errore che indica che non disponi dell'autorizzazione per eseguire un'operazione, le tue policy devono essere aggiornate in modo che ti sei consentito eseguire tale operazione.

Il seguente esempio di errore si verifica quando l'utente `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa `my-example-widget` fittizia, ma non dispone di autorizzazioni `swf:GetWidget` fittizie.

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

In questo caso, la policy deve essere aggiornata in modo che Mateo possa accedere alla risorsa `my-example-widget` mediante l'operazione `swf:GetWidget`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire `iam:PassRole`, le tue politiche devono essere aggiornate per consentirti di trasferire un ruolo ad Amazon SWF.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio anziché creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in Amazon SWF. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

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

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon SWF

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per consentire alle persone di accedere alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon SWF supporta queste funzionalità, consulta [Come funziona Amazon Simple Workflow Service con IAM](#)
- Per sapere come fornire l'accesso alle tue risorse su tutto Account AWS ciò che possiedi, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Registrazione e monitoraggio

Questa sezione fornisce informazioni sulla registrazione e il monitoraggio di Amazon SWF.

Argomenti

- [Metriche di Amazon SWF per CloudWatch](#)
- [Visualizzazione delle metriche di Amazon SWF per l'utilizzo di CloudWatch Console di gestione AWS](#)
- [Registrazione delle chiamate API con AWS CloudTrail](#)
- [EventBridge per le modifiche allo stato di esecuzione di Amazon SWF](#)
- [Utilizzo Notifiche all'utente AWS con Amazon Simple Workflow Service](#)

Metriche di Amazon SWF per CloudWatch

Amazon SWF ora fornisce parametri CloudWatch che puoi utilizzare per tenere traccia dei flussi di lavoro e delle attività e impostare allarmi sui valori di soglia che scegli. Puoi visualizzare le metriche utilizzando. Console di gestione AWS Per ulteriori informazioni, consulta [Visualizzazione delle metriche di Amazon SWF per l'utilizzo di CloudWatch Console di gestione AWS](#).

Argomenti

- [Unità di reporting per le metriche di Amazon SWF](#)
- [Parametri di eventi API e di decisione](#)
- [Metriche di Amazon SWF](#)
- [Nomi e dimensioni delle risorse non ASCII di Amazon SWF CloudWatch](#)

Unità di reporting per le metriche di Amazon SWF

Parametri che indicano un intervallo di tempo

Alcune delle metriche di Amazon SWF CloudWatch sono gli intervalli di tempo, sempre misurati in millisecondi. L'unità è riportata come. CloudWatch Time Questi parametri corrispondono generalmente alle fasi dell'esecuzione del flusso di lavoro, per le quali è possibile impostare timeout per attività e flussi di lavoro, e hanno nomi simili.

Ad esempio, il parametro `DecisionTaskStartToCloseTime` misura il tempo necessario per il completamento del task di decisione dal momento in cui viene eseguito, che è lo stesso periodo di tempo per il quale è possibile impostare un valore `DecisionTaskStartToCloseTimeout`.

Per un diagramma di ognuna di queste fasi e per informazioni su quando si verificano nei cicli di vita di flussi di lavoro e attività, consulta [Tipi di timeout di Amazon SWF](#).

Parametri che indicano un numero

Conteggio di alcune metriche di Amazon SWF per i risultati dei CloudWatch report. Ad esempio, `WorkflowsCanceled` registra un risultato che può essere uno o zero per indicare se il flusso di lavoro è stato annullato o meno. Il valore zero non indica che il parametro non è stato fornito, ma che la condizione descritta dal parametro non si è verificata.

Alcune delle metriche di Amazon SWF per CloudWatch quel rapporto a Count in CloudWatch sono un conteggio al secondo. Ad esempio `ProvisionedRefillRate`, che viene riportato come un Count in CloudWatch, rappresenta una frequenza Count delle richieste al secondo.

Per i parametri di numero, minimo e massimo saranno sempre zero o uno, ma la media sarà un valore che va da zero a uno.

Parametri di eventi API e di decisione

Puoi monitorare sia gli eventi dell'API che quelli decisionali CloudWatch per fornire informazioni dettagliate sull'utilizzo e sulla capacità. Consulta [i decisori](#) nella [Concetti di base sul flusso di lavoro in Amazon SWF](#) sezione e l'argomento [Decisione](#) nel [riferimento alle API di Amazon Simple Workflow Service](#).

Puoi anche monitorare questi limiti per generare allarmi quando ti avvicini ai limiti di limitazione di Amazon SWF. Per una descrizione di questi limiti e delle relative impostazioni di default, consulta [Quote di limitazione di Amazon SWF](#). Questi limiti impediscono ai flussi di lavoro non corretti di utilizzare risorse di sistema eccessive. Per richiedere un aumento di tali limiti, consulta [???](#).

Come best practice, dovresti configurare gli CloudWatch allarmi a circa il 60% della capacità delle tue API o degli eventi decisionali. Ciò ti consentirà di modificare il flusso di lavoro o richiedere un aumento del limite di servizio prima che la limitazione di Amazon SWF sia abilitata. A seconda del [burstiness](#) delle chiamate, puoi configurare vari allarmi per segnalare quando i limiti del servizio stanno per essere superati:

- Se il traffico presenta picchi importanti, imposta un allarme al 60% dei limiti `ProvisionedBucketSize`.
- Se la frequenza delle chiamate è regolare, imposta un allarme al 60% del limite `ProvisionedRefillRate` per gli eventi API e di decisione correlati.

Metriche di Amazon SWF

Le seguenti metriche sono disponibili per Amazon SWF:

Parametro	Descrizione
<code>DecisionTaskScheduledToStartTime</code>	L'intervallo di tempo, in millisecondi, tra il momento in cui il task di decisione è stato pianificato e quello in cui è stato acquisito da un lavoratore e avviato. CloudWatch Unità: Time

Parametro	Descrizione
	<p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Average, Minimum, Maximum</p>
DecisionTaskStartToCloseTime	<p>L'intervallo di tempo, in millisecondi, tra il momento in cui il task di decisione è stato avviato e quello in cui è stato chiuso.</p> <p>CloudWatch Unità: Time</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Average, Minimum, Maximum</p>
DecisionTasksCompleted	<p>Il numero di task di decisione completati.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>
PendingTasks	<p>Il conteggio delle attività in sospeso in un intervallo di 1 minuto per un determinato elenco di attività.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, TaskListName</p> <p>Statistiche valide: Sum</p>

Parametro	Descrizione
StartedDecisionTasksTimedOutOnClose	<p>Il numero di task di decisione iniziati ma scaduti alla chiusura.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>
WorkflowStartToCloseTime	<p>L'intervallo di tempo, in millisecondi, tra il momento in cui il flusso di lavoro è stato avviato e quello in cui è stato chiuso.</p> <p>CloudWatch Unità: Time</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Average, Minimum, Maximum</p>
WorkflowsCanceled	<p>Il numero di flussi di lavoro che sono stati annullati.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>
WorkflowsCompleted	<p>Il numero di flussi di lavoro che sono stati completati.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>

Parametro	Descrizione
<code>WorkflowsContinuedAsNew</code>	<p>Il numero di flussi di lavoro che sono continuati come nuovi.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>
<code>WorkflowsFailed</code>	<p>Il numero di flussi di lavoro non riusciti.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>
<code>WorkflowsTerminated</code>	<p>Il numero di flussi di lavoro che sono stati terminati.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Cause, Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>
<code>WorkflowsTimedOut</code>	<p>Il numero di esecuzioni che sono scadute (per qualsiasi motivo).</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiche valide: Sum</p>

Parametro	Descrizione
<code>ActivityTaskScheduleToCloseTime</code>	<p>L'intervallo di tempo, in millisecondi, tra il momento in cui l'attività è stata pianificata e quello in cui è stata chiusa.</p> <p>CloudWatch Unità: Time</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Average, Minimum, Maximum</p>
<code>ActivityTaskScheduleToStartTime</code>	<p>L'intervallo di tempo, in millisecondi, tra il momento in cui l'attività è stata pianificata e quello in cui è stata avviata.</p> <p>CloudWatch Unità: Time</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Average, Minimum, Maximum</p>
<code>ActivityTaskStartToCloseTime</code>	<p>L'intervallo di tempo, in millisecondi, tra il momento in cui il task di attività è stato avviato e quello in cui è stato chiuso.</p> <p>CloudWatch Unità: Time</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Average, Minimum, Maximum</p>
<code>ActivityTasksCancelled</code>	<p>Il numero di attività che sono state annullate.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Sum</p>

Parametro	Descrizione
ActivityTasksCompleted	<p>Il numero di attività che sono state completate.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Sum</p>
ActivityTasksFailed	<p>Il numero di attività che non sono andate a buon fine.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Sum</p>
ScheduledActivityTasksTimedOutOnClose	<p>Il numero di attività che sono state pianificate ma sono scadute alla chiusura.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Sum</p>
ScheduledActivityTasksTimedOutOnStart	<p>Il numero di attività che sono state pianificate ma sono scadute all'avvio.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Sum</p>

Parametro	Descrizione
StartedActivityTasksTimedOutOnClose	<p>Il numero di attività che sono iniziate ma sono scadute alla chiusura.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Sum</p>
StartedActivityTasksTimedOutOnHeartbeat	<p>Il numero di attività che sono iniziate ma sono scadute a causa di un timeout di heartbeat.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiche valide: Sum</p>
ThrottledEvents	<p>Il numero di richieste che sono state limitate.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: APIName, DecisionName, ThrottlingScope</p> <p>Statistiche valide: Sum</p>
ProvisionedBucketSize	<p>Il numero di richieste disponibili al secondo.</p> <p>Dimensioni: APIName, DecisionName</p> <p>Statistiche valide: Minimum</p>

Parametro	Descrizione
ConsumedCapacity	<p>Il numero di richieste al secondo.</p> <p>CloudWatch Unità: Count</p> <p>Dimensioni: APIName, DecisionName</p> <p>Statistiche valide: Sum</p>
ConsumedLimit	<p>L'importo del limite generale che è stato consumato.</p> <p>Dimensioni: GeneralLimitType</p>
ProvisionedRefillRate	<p>Il numero di richieste al secondo consentite nel bucket.</p> <p>Dimensioni: APIName, DecisionName</p> <p>Statistiche valide: Minimum</p>
ProvisionedLimit	<p>L'importo del limite generale assegnato all'account.</p> <p>Dimensioni: GeneralLimitType</p>

Dimensione	Descrizione
Domain	Filtra i dati nel dominio Amazon SWF in cui è in esecuzione il flusso di lavoro o l'attività.
ActivityTypeName	Filtra i dati in base al nome del tipo di attività.
ActivityTypeVersion	Filtra i dati in base alla versione del tipo di attività.
WorkflowTypeName	Filtra i dati in base al nome del tipo di flusso di lavoro per l'esecuzione di flusso di lavoro corrente.
WorkflowTypeVersion	Filtra i dati in base alla versione del tipo di flusso di lavoro per l'esecuzione di flusso di lavoro corrente.
APIName	Filtra i dati in base a un'API con il nome API specificato.

Dimensione	Descrizione
DecisionName	Filtra i dati in base al nome di decisione specificato.
TaskListName	Filtra i dati in base al nome dell'elenco di attività specificato.
TaskListClassification	Filtra i dati in base alla classificazione dell'elenco delle attività. Il valore è «D» per gli elenchi delle attività decisionali e «A» per gli elenchi delle attività.
ThrottlingScope	Filtra i dati in base all'ambito di limitazione specificato. Il valore è «Account» quando si supera la quota a livello di account o «Workflow» quando si supera la quota a livello di flusso di lavoro.

Nomi e dimensioni delle risorse non ASCII di Amazon SWF CloudWatch

Amazon SWF consente caratteri non ASCII nei nomi di risorse come e. TaskList DomainName Tuttavia, i valori delle dimensioni delle CloudWatch metriche possono contenere solo caratteri ASCII stampabili. Per garantire che Amazon SWF utilizzi valori di dimensione compatibili con i [CloudWatch requisiti](#), i nomi delle risorse Amazon SWF che non soddisfano questi requisiti vengono convertiti e avranno un checksum aggiunto come segue:

- Qualsiasi carattere non ASCII viene sostituito con. ?
- La stringa di input o la stringa convertita, se necessario, verrà troncata. Ciò garantisce che quando viene aggiunto il checksum, la nuova lunghezza della stringa non superi il valore massimo.
- Poiché tutti i caratteri non ASCII vengono convertiti in?, alcuni valori delle dimensioni CloudWatch metriche che erano diversi prima della conversione potrebbero sembrare gli stessi dopo la conversione. Per aiutarli a distinguerli, al nome della risorsa viene aggiunto un trattino basso (_) seguito dai primi 16 caratteri del SHA256 checksum del nome della risorsa originale.

Esempi di conversione:

- test àappleverrebbe convertito in test ?pple_82cc5b8e3a771d12
- àòàverrebbe convertito in???_2fec5edbb2c05c22.

- I TaskList nomi `apple` e `applè` verrebbero entrambi convertiti in `ppl` e sarebbero identici. L'aggiunta del checksum restituisce valori distinti e `ppl_f39a36df9d85a69d`. `ppl_da3efb4f11dd0f7f`

Tip

È possibile generare il proprio checksum SHA256 Ad esempio, per utilizzare lo strumento da riga di comando `shasum`:

```
echo -n "<the original resource name>" | shasum -a 256 | cut -c1-16
```

Visualizzazione delle metriche di Amazon SWF per l'utilizzo di CloudWatch Console di gestione AWS

Amazon CloudWatch fornisce una serie di parametri visualizzabili per i flussi di lavoro e le attività di Amazon SWF. Puoi visualizzare le metriche e impostare allarmi per le esecuzioni dei flussi di lavoro Amazon SWF utilizzando il. [Console di gestione AWS](#) Devi essere collegato alla console per proseguire.

Per una descrizione di ognuno dei parametri disponibili, consulta [Metriche di Amazon SWF per CloudWatch](#).

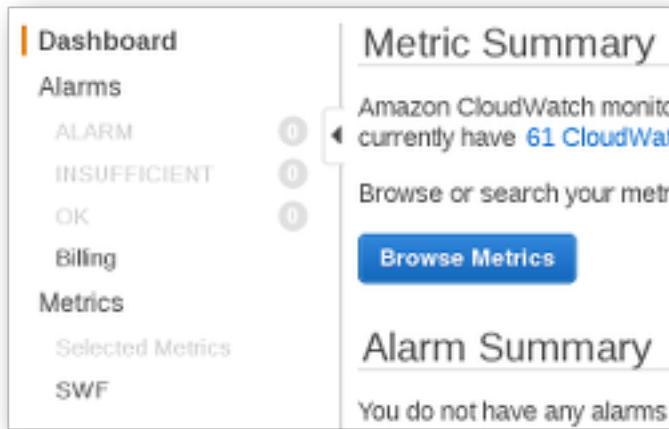
Argomenti

- [Visualizzazione dei parametri](#)
- [Impostazione degli allarmi](#)

Visualizzazione dei parametri

Per visualizzare i tuoi parametri per Amazon SWF

1. Accedi a Console di gestione AWS e apri la CloudWatch console all'indirizzo. <https://console.aws.amazon.com/cloudwatch/>
2. Nel riquadro di navigazione, in Metrics (Parametri), scegli SWF (SWF).



Se di recente hai eseguito esecuzioni del flusso di lavoro, vedrai due elenchi di parametri: Workflow Type Metrics (Parametri del tipo di flusso di lavoro) e Activity Type Metrics (Parametri del tipo di attività).

Browse Metrics				Search Metrics	X SWF Metrics
Showing all results (61) for SWF Metrics.					
Select All Clear					
SWF > Workflow Type Metrics					
Domain	WorkflowTypeName	WorkflowTypeVersion	Metric Name		
<input type="checkbox"/> HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowStartToCloseTime		
<input type="checkbox"/> HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowsCompleted		
SWF > Activity Type Metrics					
Domain	ActivityTypeName	ActivityTypeVersion	Metric Name		
<input type="checkbox"/> Booking	BookingActivity.reserve_airline	1.0	ActivityTaskScheduleToStartTime		
<input type="checkbox"/> Booking	BookingActivity.reserve_airline	1.0	ActivityTaskStartToCloseTime		

Note

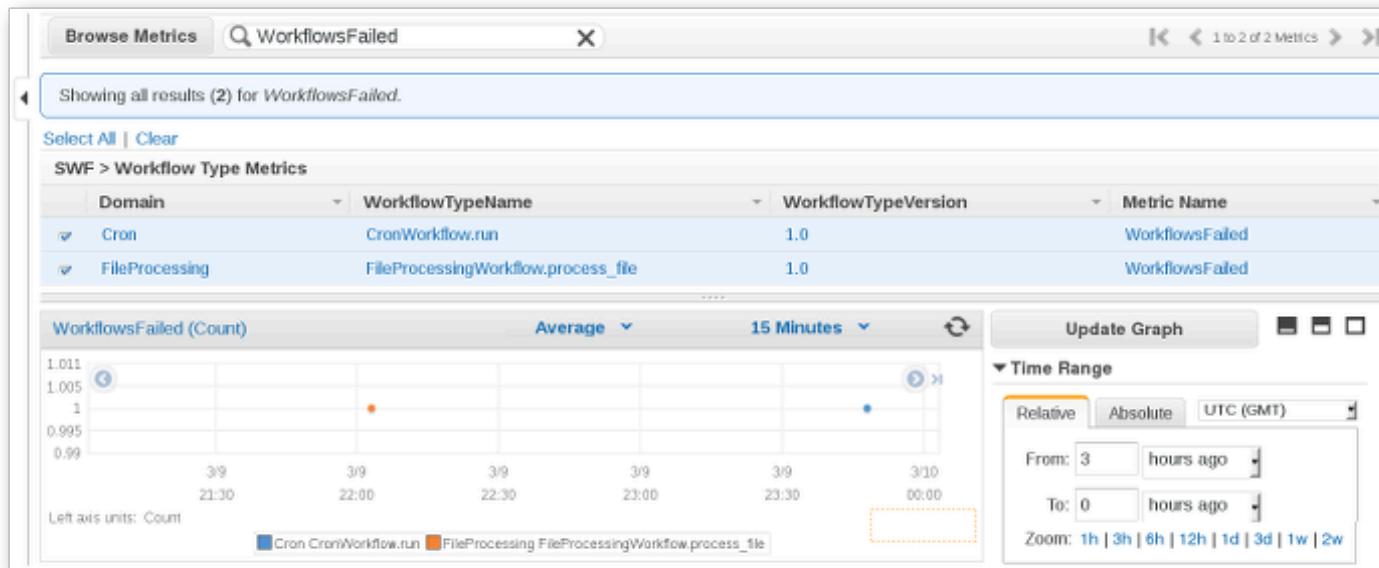
All'inizio puoi vedere solo i Workflow Type Metrics (Parametri del tipo di flusso di lavoro); i Activity Type Metrics (Parametri del tipo di flusso di lavoro) sono presentati nella stessa visualizzazione, ma per vederli devi scorrere la pagina verso il basso.

Ogni volta puoi visualizzare fino a 50 dei parametri più recenti, divisi tra parametri di flusso di lavoro e di attività.

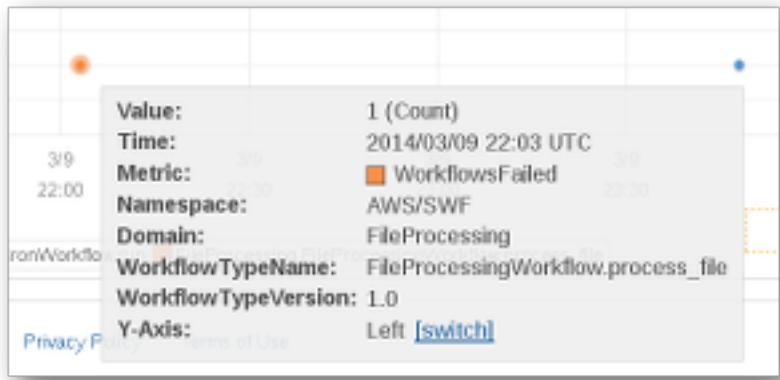
Puoi utilizzare le intestazioni interattive sopra ciascuna colonna dell'elenco per ordinare i parametri usando una delle dimensioni fornite. Per i flussi di lavoro, le dimensioni sono Domain, WorkflowType, WorkflowTypeVersion, e Metric Name. Per le attività, le dimensioni sono Domain, ActivityType, ActivityTypeVersion, e Metric Name.

I vari tipi di parametro sono descritti in [Metriche di Amazon SWF per CloudWatch](#).

Puoi visualizzare i grafici dei parametri selezionando le caselle accanto alla riga dei parametri nell'elenco; puoi modificare i parametri del grafico utilizzando i controlli Time Range (Intervallo di tempo) alla destra della visualizzazione del grafico.



Per i dettagli sui punti del grafico, posiziona il cursore su un punto. Visualizzerai un dettaglio delle dimensioni del punto.



Per ulteriori informazioni sull'utilizzo delle CloudWatch metriche, consulta [Viewing, Graphing and Publishing Metrics](#) nella Amazon CloudWatch User Guide.

Impostazione degli allarmi

Puoi utilizzare gli CloudWatch allarmi per eseguire azioni come avvisarti quando viene raggiunta una soglia di allarme. Ad esempio, puoi impostare un allarme per inviare una notifica a un argomento SNS o per inviare un'e-mail quando il parametro `WorkflowsFailed` supera una certa soglia.

Per impostare un allarme su uno dei parametri

1. Scegli un singolo parametro selezionando la relativa casella.
2. Alla destra del grafico, nei controlli Tools (Strumenti), scegli Create Alarm (Crea allarme).
3. Nella schermata Define Alarm (Definisci allarme), inserisci il valore di soglia dell'allarme, i parametri del periodo e le operazioni da eseguire.

1. Select Metric
2. Define Alarm

Back Next
Cancel

Please set the alarm threshold, actions and click **Create Alarm** below.

Create Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

Description:

Whenever: WorkflowsFailed

Is: >= 1

for: 2 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification	Delete
Whenever this alarm: State is ALARM	<input type="button" value="Delete"/>
Send notification to: SWF_Sample_Topic	<input type="button" value="New list"/>
Email list: me@example.com	<input type="button" value=""/>

Per ulteriori informazioni sull'impostazione e l'utilizzo degli CloudWatch allarmi, consulta [Creating Amazon CloudWatch Alarms](#) nella Amazon CloudWatch User Guide.

Registrazione delle chiamate API con AWS CloudTrail

Amazon Simple Workflow Service è integrato con [AWS CloudTrail](#), un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o un Servizio AWS. CloudTrail acquisisce tutte le chiamate API per Amazon SWF come eventi. Le chiamate acquisite includono chiamate dalla console Amazon SWF e chiamate in codice alle operazioni dell'API Amazon SWF. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata ad Amazon SWF, l'indirizzo IP da cui è stata effettuata la richiesta, quando è stata effettuata e ulteriori dettagli.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente root o utente.
- Se la richiesta è stata effettuata per conto di un utente del Centro identità IAM.

- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro Servizio AWS.

CloudTrail è attivo nel tuo account Account AWS quando crei l'account e hai automaticamente accesso alla cronologia degli CloudTrail eventi. La cronologia CloudTrail degli eventi fornisce un record visualizzabile, ricercabile, scaricabile e immutabile degli ultimi 90 giorni di eventi di gestione registrati in un. Regione AWS Per ulteriori informazioni, consulta [Lavorare con la cronologia degli CloudTrail eventi](#) nella Guida per l'utente AWS CloudTrail. Non sono CloudTrail previsti costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi degli Account AWS ultimi 90 giorni, crea un trail o un data store di eventi [CloudTrailLake](#).

CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando il Console di gestione AWS sono multiregionali. È possibile creare un trail per una singola Regione o per più Regioni tramite AWS CLI. La creazione di un percorso multiregionale è consigliata in quanto consente di registrare l'intera attività del proprio Regioni AWS account. Se si crea un trail per una singola Regione, è possibile visualizzare solo gli eventi registrati nella Regione AWS del trail. Per ulteriori informazioni sui trail, consulta [Creating a trail for your Account AWS](#) e [Creating a trail for an organization](#) nella Guida per l'utente di AWS CloudTrail .

Puoi inviare gratuitamente una copia dei tuoi eventi di gestione in corso al tuo bucket Amazon S3 CloudTrail creando un percorso, tuttavia ci sono costi di storage di Amazon S3. [Per ulteriori informazioni sui CloudTrail prezzi, consulta la pagina Prezzi AWS CloudTrail](#) Per informazioni sui prezzi di Amazon S3, consulta [Prezzi di Amazon S3](#).

CloudTrail Archivi di dati sugli eventi di Lake

CloudTrail Lake ti consente di eseguire query basate su SQL sui tuoi eventi. CloudTrail [Lake converte gli eventi esistenti in formato JSON basato su righe in formato Apache ORC](#). ORC è un formato di archiviazione a colonne ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i [selettori di eventi avanzati](#). I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta [Working with AWS CloudTrail Lake](#) nella Guida per l'utente AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'[opzione di prezzo](#) da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. [Per ulteriori informazioni sui CloudTrail prezzi, consulta Prezzi.AWS CloudTrail](#)

Eventi relativi ai dati in CloudTrail

Gli [eventi di dati](#) forniscono informazioni sulle operazioni delle risorse eseguite su o in una risorsa (ad esempio, lettura o scrittura su un oggetto Amazon S3). Queste operazioni sono definite anche operazioni del piano dei dati. Gli eventi di dati sono spesso attività che interessano volumi elevati di dati. Per impostazione predefinita, CloudTrail non registra gli eventi relativi ai dati. La cronologia CloudTrail degli eventi non registra gli eventi relativi ai dati.

Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione [AWS CloudTrail Prezzi](#).

Puoi registrare gli eventi relativi ai dati per i tipi di risorse Amazon SWF utilizzando la CloudTrail console o AWS CLI le operazioni CloudTrail API. Per ulteriori informazioni su come registrare gli eventi di dati, consulta [Registrazione di eventi di dati con Console di gestione AWS](#) e [Registrazione di eventi di dati con AWS Command Line Interface](#) nella Guida all'utente AWS CloudTrail .

La tabella seguente elenca i tipi di risorse Amazon SWF per i quali è possibile registrare gli eventi relativi ai dati. La colonna Data event type mostra il valore da scegliere dall'elenco Data event type sulla CloudTrail console. La colonna del valore resources.type mostra il resources.type valore da specificare durante la configurazione dei selettori di eventi avanzati utilizzando o. AWS CLI CloudTrail APIs La CloudTrail colonna Dati APIs registrati mostra le chiamate API registrate per il tipo di risorsa. CloudTrail

È possibile configurare selettori di eventi avanzati per filtrare i campi eventName, readOnly e resources.ARN per registrare solo gli eventi importanti per l'utente. Per ulteriori informazioni su questi campi, consulta [AdvancedFieldSelector](#) in Riferimento API AWS CloudTrail .

Tipo di evento relativo ai dati	valore resources.type	Dati APIs registrati su CloudTrail
Dominio SWF	AWS::SWF::Domain	Eventi del flusso di lavoro

Tipo di evento relativo ai dati	valore resources.type	Dati APIs registrati su CloudTrail
		<ul style="list-style-type: none"> • CountClosedWorkflowExecutions • CountOpenWorkflowExecutions • DescribeWorkflowExecution • ListClosedWorkflowExecutions • ListOpenWorkflowExecutions • GetWorkflowExecutionHistory • RequestCancelWorkflowExecution • SignalWorkflowExecution • StartWorkflowExecution • TerminateWorkflowExecution <p>Eventi delle attività</p> <ul style="list-style-type: none"> • CountPendingActivityTasks • PollForDecisionTask • PollForActivityTask • RecordActivityTaskHeartbeat • RespondActivityTaskCanceled • RespondActivityTaskCompleted • RespondActivityTaskFailed

Tipo di evento relativo ai dati	valore resources.type	Dati APIs registrati su CloudTrail
		<ul style="list-style-type: none"> • RespondDecisionTaskCompleted <p>Eventi decisionali</p> <ul style="list-style-type: none"> • CancelTimer • CancelWorkflowExecution • CompleteWorkflowExecution • ContinueAsNewWorkflowExecution • FailWorkflowExecution • RecordMarker • RequestCancelActivityTask • RequestCancelExternalWorkflowExecution • ScheduleActivityTask • ScheduleLambdaFunction • SignalExternalWorkflowExecution • StartChildWorkflowExecution • StartTimer

 CloudTrail eventi e RespondDecisionTaskCompleted

L'[RespondDecisionTaskCompleted](#)dazione prende un elenco di decisioni nel payload della richiesta. Una chiamata completata emetterà N+1 eventi di CloudTrail dati, uno per ogni decisione più uno per la chiamata API stessa. Gli eventi di dati e l'evento API avranno tutti lo stesso ID di richiesta.

Eventi di gestione in CloudTrail

Gli eventi di gestione forniscono informazioni sulle operazioni di gestione eseguite sulle risorse di Account AWS. Queste operazioni sono definite anche operazioni del piano di controllo (control-plane). Per impostazione predefinita, CloudTrail registra gli eventi di gestione.

Amazon Simple Workflow Service registra le seguenti operazioni del piano di controllo CloudTrail come eventi di gestione.

Eventi di dominio

- [RegisterDomain](#)
- [DescribeDomain](#)
- [ListDomains](#)
- [DeprecateDomain](#)
- [UndeprecateDomain](#)

Eventi di attività

- [RegisterActivityType](#)
- [DescribeActivityType](#)
- [ListActivityTypes](#)
- [DeprecateActivityType](#)
- [UndeprecateActivityType](#)
- [DeleteActivityType](#)

WorkflowType Eventi

- [RegisterWorkflowType](#)
- [DescribeWorkflowType](#)
- [ListWorkflowTypes](#)
- [DeprecateWorkflowType](#)
- [UndeprecateWorkflowType](#)
- [DeleteWorkflowType](#)

Tag Eventi

- TagResource
 - UntagResource
 - ListTagsForResource

Esempio di evento

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'operazione API richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle chiamate API pubbliche, quindi gli eventi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra un CloudTrail evento che dimostra l'CountClosedWorkflowExecutions operazione.

```
{  
  "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",  
  "count": 1  
}
```

```
"awsRegion": "us-east-1",
"sourceIPAddress": "198.51.100.42",
"userAgent": "aws-internal/3 aws-sdk-java/1.11.42",
"requestParameters": {
    "domain": "nsg-domain",
    "closeTimeFilter": {
        "oldestDate": "Nov 23, 2023 5:52:46 PM",
        "latestDate": "Nov 23, 2023 5:52:46 PM"
    }
},
"responseElements": null,
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb",
"readOnly": true,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::SWF::Domain",
        "ARN": "arn:aws:swf:us-east-1:111122223333:domain/nsg-domain"
    }
],
"eventType": "AwsApiCall",
"managementEvent": false,
"recipientAccountId": "111122223333",
"eventCategory": "Data",
"tlsDetails": {
    "clientProvidedHostHeader": "swf.example.amazondomains.com"
}
}
```

Per informazioni sul contenuto dei CloudTrail record, consultate il [contenuto dei CloudTrail record](#) nella Guida per l'AWS CloudTrail utente.

EventBridge per le modifiche allo stato di esecuzione di Amazon SWF

Utilizzzi Amazon EventBridge per rispondere a cambiamenti di stato o eventi in una AWS risorsa. Quando Amazon SWF emette un evento, questo passa sempre al bus EventBridge eventi predefinito per il tuo account. Puoi creare una regola per gli eventi, associarla al bus di eventi predefinito e specificare un'azione mirata da intraprendere quando EventBridge riceve un evento che corrisponde alla regola. In questo modo, puoi monitorare i tuoi flussi di lavoro senza dover effettuare sondaggi costanti utilizzando l'[GetWorkflowExecutionHistory](#) API. In base alle modifiche nelle esecuzioni del flusso di lavoro, puoi utilizzare un EventBridge target per richiamare AWS Lambda funzioni,

pubblicare messaggi su argomenti di Amazon Simple Notification Service (Amazon SNS) e altro ancora.

Puoi visualizzare il contenuto completo di un evento di modifica dello stato di esecuzione utilizzando.

[DescribeWorkflowExecution](#)

Per ulteriori informazioni, consulta la [Amazon EventBridge User Guide](#).

EventBridge eventi

I tipi di eventi della cronologia contengono le modifiche dello stato di esecuzione. La `detail` sezione di ogni evento contiene almeno i seguenti parametri:

- `eventId`: l'ID dell'evento mostrato da `GetWorkflowExecutionHistory`.
- `workflowExecutionDetail`: lo stato del flusso di lavoro al momento dell'emissione dell'evento.
- `eventType`: il tipo di evento della cronologia, uno dei seguenti:
 - `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

Esempi di eventi Amazon SWF

Di seguito sono riportati alcuni esempi di invio di eventi da parte di Amazon SWF a: EventBridge

Argomenti

- [Esecuzione iniziata](#)
- [Esecuzione completata](#)
- [Esecuzione non riuscita](#)
- [Esecuzione scaduta](#)
- [Esecuzione terminata](#)

In ogni caso, la sezione `detail` in caso di dati fornisce le stesse informazioni dell'API [DescribeWorkflowExecution](#). Il `executionStatus` campo indica lo stato dell'esecuzione al momento dell'invio dell'evento, OPEN oppure CLOSED.

Esecuzione iniziata

```
{  
  "version": "0",  
  "id": "444444444444",  
  "type": "WorkflowExecutionStarted",  
  "detail": {  
    "workflowId": "444444444444",  
    "runId": "444444444444",  
    "executionStartToCloseTimeout": 3600,  
    "executionStartToCloseTimestamp": 1516229180000,  
    "executionVersion": 1,  
    "taskList": "MyTaskList",  
    "tagList": []  
  }  
}
```

```
"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,
}
}
```

Esecuzione completata

```
{
  "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,
}
}
}
```

Esecuzione non riuscita

```
{
    "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::44445556666:role/BasicSWFLambdaExecution"
    },
    "openCounts": {
        "openActivityTasks": 0,
        "openDecisionTasks": 0,
        "openTimers": 0,
        "openChildWorkflowExecutions": 0,
        "openLambdaFunctions": 0
    },
    "latestActivityTaskTimestamp": null,
}
}
}
```

Esecuzione scaduta

```
{
    "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,
    }
}
}
```

Esecuzione terminata

```
{
    "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,
}
}
}
```

Utilizzo Notifiche all'utente AWS con Amazon Simple Workflow Service

Puoi utilizzarlo [Notifiche all'utente AWS](#) per configurare i canali di distribuzione per ricevere notifiche sugli eventi di Amazon Simple Workflow Service. L'utente riceverà una notifica quando un evento corrisponde a una regola specificata. Puoi ricevere notifiche relative agli eventi attraverso più canali, tra cui e-mail, [Amazon Q Developer in chat, applicazioni](#) di chat, notifiche di chat o notifiche [AWS Console Mobile Application](#) push. Puoi anche visualizzare le notifiche nel [Centro notifiche della console](#). Notifiche all'utente supporta l'aggregazione, che può ridurre il numero di notifiche ricevute durante eventi specifici.

Convalida della conformità per Amazon Simple Workflow Service

I revisori di terze parti valutano la sicurezza e la conformità di Amazon Simple Workflow Service nell'ambito di diversi programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta [AWS Services in Scope by Compliance Program AWS](#). Per informazioni generali, vedere Programmi di [AWS conformità Programmi](#).

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#)

La tua responsabilità di conformità quando usi Amazon SWF è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide rapide su sicurezza e conformità](#) [Guide introduttive](#) implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- Whitepaper [sull'architettura per la sicurezza e la conformità HIPAA: questo white paper](#) descrive in che modo le aziende possono utilizzare per creare applicazioni conformi all'HIPAA. AWS
- AWS Risorse per [la conformità Risorse per la conformità](#): questa raccolta di potrebbe riguardare il settore e la località in cui operate.
- [Valutazione delle risorse con le regole](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub CSPM](#)— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente, AWS che consente di verificare la conformità agli standard e alle best practice del settore della sicurezza.

Resilienza in Amazon Simple Workflow Service

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta [Global Infrastructure.AWS](#)

Oltre all'infrastruttura AWS globale, Amazon SWF offre diverse funzionalità per supportare le tue esigenze di resilienza e backup dei dati.

Sicurezza dell'infrastruttura in Amazon Simple Workflow Service

In quanto servizio gestito, è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Puoi chiamare queste operazioni API da qualsiasi posizione di rete, ma Amazon SWF supporta politiche di accesso basate sulle risorse, che possono includere restrizioni basate sull'indirizzo IP di origine. Puoi anche utilizzare le policy di Amazon SWF per controllare l'accesso da endpoint Amazon Virtual Private Cloud (Amazon VPC) specifici o specifici VPCs. In effetti, questo isola l'accesso alla rete a una determinata risorsa Amazon SWF solo dal VPC specifico all'interno della rete. AWS

Analisi della configurazione e delle vulnerabilità in Amazon Simple Workflow Service

La configurazione e i controlli IT sono una responsabilità condivisa tra te AWS e te, nostro cliente. Per ulteriori informazioni, consulta il [modello di responsabilità AWS condivisa](#).

Utilizzo del servizio AWS CLI con Amazon Simple Workflow Service

È possibile accedere a molte delle funzionalità di Amazon Simple Workflow Service da AWS CLI. AWS CLI Fornisce un'alternativa all'utilizzo di Amazon SWF con Console di gestione AWS o, in alcuni casi, alla programmazione con l'API Amazon SWF e il. AWS Flow Framework

Ad esempio, puoi utilizzare il per AWS CLI registrare un nuovo tipo di flusso di lavoro:

```
aws swf register-workflow-type --domain MyDomain --name "MySimpleWorkflow" --workflow-version "v1"
```

Puoi inoltre elencare i tipi di flusso di lavoro registrati:

```
aws swf list-workflow-types --domain MyDomain --registration-status REGISTERED
```

Quanto segue è un esempio dell'output di default in 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"  
      }  
    }  
  ]  
}
```

I comandi di Amazon SWF AWS CLI offrono la possibilità di avviare e gestire le esecuzioni dei flussi di lavoro, eseguire sondaggi sulle attività, registrare i battiti cardiaci delle attività e molto altro ancora! Per un elenco completo dei comandi Amazon SWF, con descrizioni degli argomenti disponibili ed esempi che ne illustrano l'utilizzo, consulta i comandi [Amazon SWF](#) nel AWS CLI Command Reference.

I AWS CLI comandi seguono da vicino l'API Amazon SWF, quindi puoi usarli AWS CLI per conoscere l'API Amazon SWF sottostante. Puoi anche utilizzare le tue conoscenze esistenti in materia di API per prototipare codice o eseguire azioni Amazon SWF sulla riga di comando.

Per ulteriori informazioni su AWS CLI, consulta la Guida per l'[AWS Command Line Interface utente](#).

Lavorare con Amazon SWF APIs

Oltre a utilizzare quanto descritto AWS SDKs in [Sviluppa con AWS SDKs](#), puoi utilizzare direttamente l'API HTTP.

Per usare l'API, devi inviare richieste HTTP all'[endpoint di SWF](#) che corrisponde alla regione che vuoi utilizzare per i domini, flussi di lavoro e attività. Per ulteriori informazioni su come effettuare richieste HTTP per Amazon SWF, consulta. [Invio di richieste HTTP ad Amazon SWF](#)

Questa sezione fornisce informazioni di base sull'uso dell'API HTTP per sviluppare flussi di lavoro con Amazon SWF. Nella sezione sono disponibili funzionalità più avanzate, come l'uso di timer, la registrazione CloudTrail e l'etichettatura dei flussi di lavoro. [Concetti di base sul flusso di lavoro in Amazon SWF](#)

Argomenti

- [Invio di richieste HTTP ad Amazon SWF](#)
- [Elenco delle azioni Amazon SWF per categoria](#)
- [Registrazione di un dominio con Amazon SWF](#)
- [Impostazione dei valori di timeout in Amazon SWF](#)
- [Registrazione di un tipo di flusso di lavoro con Amazon SWF](#)
- [Registrazione di un tipo di attività con Amazon SWF](#)
- [AWS Lambda attività in Amazon SWF](#)
- [Sviluppo di un Activity Worker in Amazon SWF](#)
- [Sviluppo di dispositivi decisionali in Amazon SWF](#)
- [Avvio di flussi di lavoro in Amazon SWF](#)
- [Impostazione della priorità delle attività in Amazon SWF](#)
- [Gestione degli errori in Amazon SWF](#)

Invio di richieste HTTP ad Amazon SWF

Se non utilizzi uno di questi AWS SDKs, puoi eseguire operazioni di Amazon Simple Workflow Service (Amazon SWF) su HTTP utilizzando il metodo di richiesta POST. Per utilizzare il metodo POST devi specificare l'operazione nell'intestazione della richiesta e fornire i dati per l'operazione nel formato JSON nel corpo della richiesta.

Contenuti nell'intestazione HTTP

Amazon SWF richiede le seguenti informazioni nell'intestazione di una richiesta HTTP:

- **host**L'endpoint Amazon SWF.
- **x-amz-date**È necessario fornire il timestamp nell'header Date o nel AWS x-amz-date header (alcune librerie client HTTP non consentono di impostare l'intestazione). Date Quando un'intestazione x-amz-date è presente, il sistema ignora qualsiasi intestazione Date all'autenticazione della richiesta.

La data deve essere specificata in uno dei seguenti tre formati, come indicato nel protocollo HTTP/1.1 RFC:

- Dom, 06 novembre 1994 08:49:37 GMT (RFC 822, aggiornato da RFC 1123)
- Domenica, 06-Nov-94 08:49:37 GMT (RFC 850, reso obsoleto da RFC 1036)
- Dom Nov 6 08:49:37 1994 (Formato asctime() ANSI C)
- **x-amzn-authorization** I parametri della richiesta firmata nel formato:

```
AWS3 AWSAccessKeyId=####,Algorithm=HmacSHA256, [ ,SignedHeaders=Header1;Header2;... ]  
Signature=S(StringToSign)
```

AWS3— Si tratta di un tag AWS specifico dell'implementazione che indica la versione di autenticazione utilizzata per firmare la richiesta (attualmente, per Amazon SWF questo valore è sempre). AWS3

AWSAccessKeyId— L'ID della tua chiave di accesso. AWS

Algorithm— L'algoritmo utilizzato per creare il valore HMAC-SHA di string-to-sign, ad esempio o. HmacSHA256 HmacSHA1

Signature— Base64 (Algorithm (,)). StringToSign SigningKey Per informazioni dettagliate, consulta [Calcolo della firma HMAC-SHA per Amazon SWF](#)

SignedHeaders— (Facoltativo) Se presente, deve contenere un elenco di tutte le intestazioni HTTP utilizzate nel calcolo canonicalizzato. HttpHeaders Un singolo punto e virgola (;) (carattere ASCII 59) deve essere utilizzato come delimitatore per i valori dell'elenco.

- **x-amz-target**— Il servizio di destinazione della richiesta e l'operazione per i dati, nel formato com.amazonaws.swf.service.model.SimpleWorkflowService. + **<action>**

Ad esempio,

`com.amazonaws.swf.service.model.SimpleWorkflowService.RegisterDomain`

- `content-type`— Il tipo deve specificare JSON e il set di caratteri, come `application/json; charset=UTF-8`

Il seguente è un esempio di intestazione per una richiesta HTTP per la creazione di un dominio.

```
POST http://swf.us-east-1.amazonaws.com/ HTTP/1.1
Host: swf.us-east-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.25) Gecko/20111212
Firefox/3.6.25 (.NET CLR 3.5.30729; .NET4.0E)
Accept: application/json, text/javascript, /*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Content-Type: application/json; charset=UTF-8
X-Requested-With: XMLHttpRequest
X-Amz-Date: Fri, 13 Jan 2012 18:42:12 GMT
X-Amz-Target: com.amazonaws.swf.service.model.SimpleWorkflowService.RegisterDomain
Content-Encoding: amz-1.0
X-Amzn-Authorization: AWS3
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Algorithm=HmacSHA256,SignedHeaders=Host;X-Amz-
Date;X-Amz-Target;Content-Encoding,Signature=tzjkF551xAxPhzp/BRGFYQRQRq6CqrM254dTDE/
EncI=
Referer: http://swf.us-east-1.amazonaws.com/explorer/index.html
Content-Length: 91
Pragma: no-cache
Cache-Control: no-cache

{"name": "867530902",
 "description": "music",
 "workflowExecutionRetentionPeriodInDays": "60"}
```

Di seguito è riportato un esempio della risposta HTTP corrispondente.

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

x-amzn-RequestId: 4ec4ac3f-3e16-11e1-9b11-7182192d0b57

Contenuto del corpo HTTP

Il corpo di una richiesta HTTP contiene i dati per l'operazione specificata nell'intestazione di una richiesta HTTP. Utilizza il formato dati JSON per trasmettere simultaneamente i valori dei dati e la struttura corrispondente. Gli elementi possono essere annidati all'interno di altri elementi utilizzando la notazione parentesi. Ad esempio, quanto segue mostra una richiesta per elencare tutte le esecuzioni del flusso di lavoro iniziate tra due momenti specifici, utilizzando la notazione Unix Time.

```
{  
  "domain": "867530901",  
  "startTimeFilter":  
  {  
    "oldestDate": 1325376070,  
    "latestDate": 1356998399  
  },  
  "tagFilter":  
  {  
    "tag": "music purchase"  
  }  
}
```

Esempio di richiesta e risposta JSON di Amazon SWF

L'esempio seguente mostra una richiesta ad Amazon SWF per una descrizione del dominio che abbiamo creato in precedenza. Quindi mostra la risposta di Amazon SWF.

Richiesta 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=IFJtq3M366CHqM1TpyqYqd9z0ChCoKDC5SCJBsLifu4=
Referer: http://swf.us-east-1.amazonaws.com/explorer/index.html
Content-Length: 21
Pragma: no-cache
Cache-Control: no-cache

{"name": "867530901"}
```

Risposta di 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"}
}
```

Il protocollo (HTTP/1.1) è seguito dal codice di stato (200). Un valore del codice di 200 indica un'operazione riuscita.

Amazon SWF non serializza valori nulli. Se il parser JSON è impostato per serializzare valori nulli per le richieste, Amazon SWF li ignora.

Calcolo della firma HMAC-SHA per Amazon SWF

Ogni richiesta ad Amazon SWF deve essere autenticata. Firma AWS SDKs automaticamente le tue richieste e gestisci l'autenticazione basata su token. Tuttavia, se vuoi scrivere una tua richiesta HTTP

POST, devi creare un valore `x-amzn-authorization` per i contenuti HTTP POST Header come parte dell'autenticazione della richiesta.

Per maggiori informazioni sulla formattazione delle intestazioni, consulta [Contenuti nell'intestazione HTTP](#). Per l' AWS SDK per Java implementazione della firma della AWS versione 3, consultate la [AWSSignerclasse.java](#).

Creazione di una firma di richiesta

Prima di creare una firma di richiesta HMAC-SHA, devi ricevere le credenziali AWS (l'ID chiave di accesso e la chiave segreta).

Important

Puoi usare entrambi SHA1 o SHA256 per firmare le tue richieste. Verifica comunque di utilizzare lo stesso metodo in tutto il processo di firma. Il metodo scelto deve corrispondere al valore del nome `Algorithm` nell'intestazione HTTP.

Per creare la firma di richiesta

1. Crea un modello canonico delle intestazioni di richiesta HTTP. Il formato canonico dell'intestazione HTTP comprende quanto segue:

- host
- Qualunque elemento dell'intestazione che inizi con `x-amz-`

Per ulteriori informazioni sulle intestazioni incluse, consulta [Contenuti nell'intestazione HTTP](#).

- Per ogni coppia nome-valore dell'intestazione, converti il nome dell'intestazione (ma non il valore) in caratteri minuscoli.
- Creare una mappa del nome dell'intestazione con i valori separati da virgole.

```
x-amz-example: value1
x-amz-example: value2 => x-amz-example:value1,value2
```

Per maggiori informazioni, consulta la [sezione 4.2 del protocollo RFC 2616](#).

- c. Per ogni coppia nome-valore dell'intestazione, converti la coppia nome-valore in una stringa nel formato `headerName : headerValue`. Elimina gli spazi all'inizio e alla fine del `headerName` e del `headerValue`, senza spazi ai lati dei due punti.

```
x-amz-example1:value1,value2  
x-amz-example2:value3
```

- d. Inserisci una nuova riga (U+000A) dopo ogni stringa convertita, compresa l'ultima.
- e. Disponi la raccolta delle stringhe convertite in ordine alfabetico in base al nome dell'intestazione.

2. Crea un string-to-sign valore che includa i seguenti elementi:

- Riga 1: il metodo HTTP (POST), seguito da una nuova riga.
- Riga 2: l'URI della richiesta (/), seguito da una nuova riga.
- Riga 3: una stringa vuota seguita da una nuova riga.

 Note

In genere, la stringa di query viene visualizzata qui, ma Amazon SWF non utilizza una stringa di query.

- Righe 4-n: la stringa che rappresenta le intestazioni di richiesta in forma canonica calcolate nella Fase 1, seguita da una nuova riga. Questa nuova riga crea una riga vuota tra le intestazioni e il corpo della richiesta HTTP. Per ulteriori informazioni, consulta il protocollo [RFC 2616](#).
 - Il corpo della richiesta, non seguito da una nuova riga.
3. Calcola SHA256 o SHA1 digest il valore. string-to-sign Utilizza lo stesso metodo SHA in tutto il processo.
 4. Calcola e codifica in Base64 l'HMAC-SHA utilizzando un digest SHA256 o un SHA1 digest (a seconda del metodo utilizzato) del valore risultante dal passaggio precedente e la chiave di accesso segreta temporanea del Security Token Service utilizzando l'azione API. AWS [GetSessionToken](#)

 Note

Amazon SWF prevede un segno di uguaglianza (=) alla fine del valore HMAC-SHA con codifica Base64. Se la tua routine di codifica in base 64 non comprende l'aggiunta di segni uguali, aggiungine uno alla fine del valore.

Per ulteriori informazioni sull'utilizzo di credenziali di sicurezza temporanee con Amazon SWF e AWS altri servizi, [AWS consulta Services That Work with IAM](#) nella IAM User Guide.

5. Inserisci il valore risultante come valore per il `Signature` nome nell'`x-amzn-authorization` intestazione della richiesta HTTP ad Amazon SWF.
6. Amazon SWF verifica la richiesta ed esegue l'operazione specificata.

Elenco delle azioni Amazon SWF per categoria

Questa sezione elenca gli argomenti di riferimento per le azioni di Amazon SWF nell'interfaccia di programmazione delle applicazioni (API) di Amazon SWF. Le operazioni sono elencate per categoria funzionale.

Per un elenco alfabetico delle azioni, consulta il [riferimento alle API di Amazon Simple Workflow Service](#).

Argomenti

- [Operazioni correlate alle attività](#)
- [Operazioni correlate ai decisori](#)
- [Operazioni correlate alle esecuzioni di flusso di lavoro](#)
- [Operazioni correlate all'amministrazione](#)
- [Operazioni di visibilità](#)

Operazioni correlate alle attività

I lavoratori che si occupano delle attività usano `PollForActivityTask` per ricevere nuovi task. Dopo che un lavoratore riceve un'attività da Amazon SWF, la esegue e risponde indicando `RespondActivityTaskCompleted` se l'operazione ha avuto esito positivo o `RespondActivityTaskFailed` negativo.

Di seguito sono elencate le operazioni eseguite dai lavoratori di attività.

- [PollForActivityTask](#)
- [RespondActivityTaskCompleted](#)
- [RespondActivityTaskFailed](#)
- [RespondActivityTaskCanceled](#)
- [RecordActivityTaskHeartbeat](#)

Operazioni correlate ai decisori

I decisori usano PollForDecisionTask per ottenere i task di decisione. Dopo che un decisore riceve un'attività decisionale da Amazon SWF, esamina la cronologia di esecuzione del flusso di lavoro e decide cosa fare dopo. Chiama RespondDecisionTaskCompleted per completare il task di decisione e propone zero o più decisioni successive.

Di seguito sono elencate le operazioni eseguite dai decisori.

- [PollForDecisionTask](#)
- [RespondDecisionTaskCompleted](#)

Operazioni correlate alle esecuzioni di flusso di lavoro

Le operazioni seguenti sono utilizzate in un'esecuzione di flusso di lavoro.

- [RequestCancelWorkflowExecution](#)
- [StartWorkflowExecution](#)
- [SignalWorkflowExecution](#)
- [TerminateWorkflowExecution](#)

Operazioni correlate all'amministrazione

Sebbene sia possibile eseguire attività amministrative dalla console Amazon SWF, è possibile utilizzare le azioni in questa sezione per automatizzare le funzioni o creare strumenti amministrativi personalizzati.

Gestione delle attività

- [RegisterActivityType](#)
- [DeprecateActivityType](#)
- [UndeprecateActivityType](#)
- [DeleteActivityType](#)

Gestione dei flussi di lavoro

- [RegisterWorkflowType](#)
- [DeprecateWorkflowType](#)
- [UndeprecateWorkflowType](#)
- [DeleteWorkflowType](#)

Gestione dei domini

Queste azioni consentono di registrare e rendere obsoleti i domini Amazon SWF.

- [RegisterDomain](#)
- [DeprecateDomain](#)
- [UndeprecateDomain](#)

Per ulteriori informazioni ed esempi relativi a queste operazioni di gestione di domini, consulta [Registrazione di un dominio con Amazon SWF](#).

Gestione dell'esecuzione dei flussi di lavoro

- [RequestCancelWorkflowExecution](#)
- [TerminateWorkflowExecution](#)

Operazioni di visibilità

Sebbene sia possibile eseguire azioni di visibilità dalla console Amazon SWF, è possibile utilizzare le azioni in questa sezione per creare la propria console o strumenti di amministrazione.

Visibilità delle attività

- [ListActivityTypes](#)
- [DescribeActivityType](#)

Visibilità dei flussi di lavoro

- [ListWorkflowTypes](#)
- [DescribeWorkflowType](#)

Visibilità dell'esecuzione dei flussi di lavoro

- [DescribeWorkflowExecution](#)
- [ListOpenWorkflowExecutions](#)
- [ListClosedWorkflowExecutions](#)
- [CountOpenWorkflowExecutions](#)
- [CountClosedWorkflowExecutions](#)
- [GetWorkflowExecutionHistory](#)

Visibilità dei domini

- [ListDomains](#)
- [DescribeDomain](#)

Visibilità degli elenchi di task

- [CountPendingActivityTasks](#)
- [CountPendingDecisionTasks](#)

Registrazione di un dominio con Amazon SWF

I tipi di flusso di lavoro e attività nonché l'esecuzione di flusso di lavoro rientrano nell'ambito di un dominio. I domini separano un set di tipi, esecuzioni ed elenchi di task da altri nello stesso account.

Puoi registrare un dominio utilizzando Console di gestione AWS o utilizzando l'RegisterDomainazione nell'API Amazon SWF. L'esempio seguente utilizza l'API.

```
https://swf.us-east-1.amazonaws.com
RegisterDomain
{
  "name" : "867530901",
  "description" : "music",
  "workflowExecutionRetentionPeriodInDays" : "60"
}
```

I parametri sono specificati in formato JavaScript Object Notation (JSON). In questo caso, il periodo di retention è di 60 giorni. Durante il periodo di conservazione, tutte le informazioni sull'esecuzione del flusso di lavoro sono disponibili tramite operazioni di visibilità utilizzando l'API Amazon SWF Console di gestione AWS o l'API Amazon SWF.

Dopo la registrazione del dominio, devi registrare il tipo di flusso di lavoro e i tipi di attività utilizzati dal flusso di lavoro. Registra dapprima il dominio in quanto un nome di dominio registrato è una delle informazioni necessarie per registrare i tipi di flusso di lavoro e attività.

Vedi anche

[RegisterDomain](#) nel riferimento all'API di riferimento di Amazon Simple Workflow Service

Impostazione dei valori di timeout in Amazon SWF

Argomenti

- [Quote sui valori di timeout](#)
- [Timeout dell'esecuzione del flusso di lavoro e del task di decisione](#)
- [Timeout del task di attività](#)
- [Vedi anche](#)

Quote sui valori di timeout

I valori di timeout sono sempre dichiarati in secondi e possono essere impostati su qualsiasi numero di secondi fino a un anno (31536000 secondi), il limite massimo di esecuzione per qualsiasi flusso di lavoro o attività. Il valore speciale NONE viene utilizzato per impostare un parametro di timeout su "no timeout" o su infinito, ma viene ancora applicato il limite massimo di un anno.

Timeout dell'esecuzione del flusso di lavoro e del task di decisione

Puoi impostare i valori di timeout per i task di flusso di lavoro e decisione durante la registrazione del tipo di flusso di lavoro. Per esempio:

```
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"
}
```

La registrazione di questo tipo di flusso di lavoro imposta il [defaultTaskStartToCloseTimeout](#) su 600 secondi (10 minuti) e [defaultExecutionStartToCloseTimeout](#) su 3.600 secondi (1 ora).

Per ulteriori informazioni sulla registrazione del tipo di flusso di lavoro [Registrazione di un tipo di flusso di lavoro con Amazon SWF](#), consulta e [RegisterWorkflowType](#) nel riferimento alle API di Amazon Simple Workflow Service.

Puoi sovrascrivere il valore impostato per `defaultExecutionStartToCloseTimeout` specificando [executionStartToCloseTimeout](#) .

Timeout del task di attività

Puoi impostare i valori di timeout per i task di attività durante la registrazione del tipo di attività. Per esempio:

```
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"
}
```

La registrazione del tipo di attività imposta `defaultTaskStartToCloseTimeout` su 600 secondi (10 minuti), `defaultTaskHeartbeatTimeout` su 120 secondi (2 minuti), `defaultTaskScheduleToStartTimeout` su 1.800 secondi (30 minuti) e `defaultTaskScheduleToCloseTimeout` su 5.400 secondi (1,5 ore).

Per ulteriori informazioni sulla registrazione del tipo di attività [Registrazione di un tipo di attività con Amazon SWF](#), consulta e [RegisterActivityType](#) nel riferimento alle API di Amazon Simple Workflow Service.

Puoi sovrascrivere il valore impostato per `defaultTaskStartToCloseTimeout` specificando `taskStartToCloseTimeout` durante la programmazione del task attività.

Vedi anche

[Tipi di timeout di Amazon SWF](#)

Registrazione di un tipo di flusso di lavoro con Amazon SWF

L'esempio discusso in questa sezione registra un tipo di flusso di lavoro utilizzando l'API Amazon SWF. Il nome e la versione specificati in fase di registrazione formano un identificativo univoco del tipo di flusso di lavoro. Il dominio specificato deve essere già stato registrato utilizzando l'operazione API [RegisterDomain](#).

I parametri di timeout nel seguente esempio sono valori di durata espressi in secondi. Per il parametro `defaultTaskStartToCloseTimeout`, puoi utilizzare lo specificatore di durata `NONE` per indicare l'assenza di timeout. Tuttavia, non puoi specificare un valore `NONE` per `defaultExecutionStartToCloseTimeout`; esiste un limite massimo di un anno per la durata dell'esecuzione di un flusso di lavoro. Il superamento di questo limite causa sempre la scadenza dell'esecuzione. Se specifichi un valore di `defaultExecutionStartToCloseTimeout` maggiore a un anno, la registrazione non andrà a buon fine.

```
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"  
}
```

Vedi anche

[RegisterWorkflowType](#) nel riferimento all'API di riferimento di Amazon Simple Workflow Service

Registrazione di un tipo di attività con Amazon SWF

L'esempio seguente registra un tipo di attività utilizzando l'API Amazon SWF. Il nome e la versione specificati durante la registrazione formano un identificativo univoco per il tipo di attività nel dominio. Il dominio specificato deve essere già stato registrato utilizzando l'operazione `RegisterDomain`.

I parametri di timeout in questo esempio sono valori di durata espressi in secondi. Puoi utilizzare l'identificatore di durata `NONE` per indicare l'assenza di timeout.

```
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"  
}
```

Vedi anche

[RegisterActivityType](#) nel riferimento all'API di riferimento di Amazon Simple Workflow Service

AWS Lambda attività in Amazon SWF

Argomenti

- [Informazioni su AWS Lambda](#)
- [Vantaggi e limiti dell'utilizzo delle attività Lambda](#)
- [Utilizzo delle attività Lambda nei flussi di lavoro](#)

Informazioni su AWS Lambda

AWS Lambda è un servizio di elaborazione completamente gestito che esegue il codice in risposta a eventi generati da codice personalizzato o da vari AWS servizi come Amazon S3, DynamoDB, Amazon Kinesis, Amazon SNS e Amazon Cognito. Per ulteriori informazioni su Lambda, consulta la [Guida per gli sviluppatori di AWS Lambda](#).

Amazon Simple Workflow Service fornisce un task Lambda che consente di eseguire funzioni Lambda al posto o insieme alle attività tradizionali di Amazon SWF.

Important

Sul tuo AWS account verranno addebitate le esecuzioni (richieste) Lambda eseguite da Amazon SWF per tuo conto. [Per informazioni dettagliate sui prezzi di Lambda, consulta <https://aws.amazon.com/lambda/pricing/>](#)

Vantaggi e limiti dell'utilizzo delle attività Lambda

L'utilizzo delle attività Lambda al posto di un'attività tradizionale di Amazon SWF offre numerosi vantaggi:

- Le attività Lambda non devono essere registrate o sottoposte a versioni come i tipi di attività di Amazon SWF.
- Puoi utilizzare qualsiasi funzione Lambda esistente che hai già definito nei tuoi flussi di lavoro.
- Le funzioni Lambda vengono richiamate direttamente da Amazon SWF; non è necessario implementare un programma di lavoro per eseguirle come è necessario fare con le attività tradizionali.
- Lambda fornisce metriche e log per tracciare e analizzare le esecuzioni delle funzioni.

L'utilizzo di task Lambda comporta anche alcuni limiti che è necessario conoscere:

- Le attività Lambda possono essere eseguite solo nelle AWS regioni che forniscono supporto per Lambda. Consulta [le regioni e gli endpoint Lambda](#) nel riferimento generale di Amazon Web Services per dettagli sulle regioni attualmente supportate per Lambda.
- Le attività Lambda sono attualmente supportate solo dall'API HTTP SWF di base e in Java. AWS Flow Framework Al momento non è disponibile alcun supporto per le attività Lambda in AWS Flow Framework for Ruby.

Utilizzo delle attività Lambda nei flussi di lavoro

Per utilizzare le attività Lambda nei flussi di lavoro Amazon SWF, dovrà:

1. Configura i ruoli IAM per fornire ad Amazon SWF l'autorizzazione a richiamare le funzioni Lambda.
2. Collegare i ruoli IAM ai tuoi flussi di lavoro.
3. Chiamare la funzione Lambda durante un'esecuzione di flusso di lavoro.

Configurazione di un ruolo IAM

Prima di poter richiamare le funzioni Lambda da Amazon SWF, devi fornire un ruolo IAM che fornisca l'accesso a Lambda da Amazon SWF. Puoi eseguire una delle seguenti operazioni:

- scegli un ruolo predefinito, Ruolo, AWSLambda per autorizzare i flussi di lavoro a richiamare qualsiasi funzione Lambda associata al tuo account.
- definisci la tua politica e il ruolo associato per autorizzare i flussi di lavoro a richiamare particolari funzioni Lambda, specificate dai rispettivi Amazon Resource Names (ARNs).

Limita le autorizzazioni su un ruolo IAM

Puoi limitare le autorizzazioni su un ruolo IAM che fornisci ad Amazon SWF utilizzando SourceARN le chiavi SourceAccount e context nella tua policy di attendibilità delle risorse. Queste chiavi limitano l'utilizzo di una policy IAM in modo che venga utilizzata solo dalle esecuzioni di Amazon Simple Workflow Service che appartengono all'ARN del dominio specificato. Se utilizzi entrambe le chiavi di contesto della condizione globale, il aws:SourceAccount valore e l'account a cui si fa riferimento nel aws:SourceArn valore devono utilizzare lo stesso ID account quando vengono utilizzati nella stessa dichiarazione politica.

Nel seguente esempio, la chiave di SourceArn contesto limita l'utilizzo del ruolo del servizio IAM solo nelle esecuzioni di Amazon Simple Workflow Service che appartengono someDomain all'account., 123456789012

- Dichiarazione 1

Preside: "Service": "swf.amazonaws.com"

Operazione: sts:AssumeRole

```
"Condition": {  
    "ArnLike": {  
        "aws:SourceArn": "arn:aws:swf:*:123456789012:/domain/someDomain"  
    }  
}
```

Nell'esempio seguente, la chiave di SourceAccount contesto limita l'utilizzo del ruolo del servizio IAM solo nelle esecuzioni di Amazon Simple Workflow Service nell'account., 123456789012

```
"Condition": {  
    "StringLike": {  
        "aws:SourceAccount": "123456789012"  
    }  
}
```

Fornire ad Amazon SWF l'accesso per richiamare qualsiasi ruolo Lambda

Puoi utilizzare il ruolo predefinito, Role, per dare ai flussi di lavoro Amazon SWF la possibilità di richiamare qualsiasi AWSLambda funzione Lambda associata al tuo account.

Utilizzare AWSLambda Role per consentire ad Amazon SWF l'accesso per richiamare le funzioni Lambda

1. Apri la [console Amazon IAM](#).
2. Scegli Roles (Ruoli), quindi Create New Role (Crea nuovo ruolo).
3. Assegna un nome al ruolo, come swf-lambda, quindi scegli Next Step (Fase successiva).
4. In AWS Service Roles, scegli Amazon SWF e scegli Next Step.
5. Nella schermata Attach Policy, scegli AWSLambdaRole dall'elenco.

6. Scegli Next Step (Fase successiva), quindi Create Role (Crea ruolo) dopo aver esaminato il ruolo.

Definizione di un ruolo IAM per fornire l'accesso per richiamare una funzione Lambda specifica

Se desideri fornire l'accesso per richiamare una funzione Lambda specifica dal tuo flusso di lavoro, dovrà definire la tua policy IAM.

Creare una policy IAM per fornire l'accesso a una particolare funzione Lambda

1. Apri la [console Amazon IAM](#).
2. Scegli Policies (Policy), quindi Create Policy (Crea policy).
3. Scegli Copia una policy AWS gestita e seleziona AWSLambdaRole dall'elenco. Viene generata una policy. Se necessario, modificala il nome e la descrizione.
4. Nel campo Resource del Policy Document, aggiungi l'ARN delle tue funzioni Lambda. Per esempio:
 - Risorsa: `arn:aws:lambda:us-east-1:111122223333:function:hello_lambda_function`

 Note

Per una descrizione completa di come specificare le risorse in un ruolo IAM, consulta [Panoramica delle politiche IAM](#) nell'uso di IAM.

5. Scegli Create Policy (Crea policy) per completare la creazione della policy.

Puoi quindi selezionare questa policy quando crei un nuovo ruolo IAM e utilizzarlo per concedere a invoke l'accesso ai tuoi flussi di lavoro Amazon SWF. Questa procedura è molto simile alla creazione di un ruolo con la politica AWSLambdaRole. Scegli invece la tua policy quando crei il ruolo.

Per creare un ruolo Amazon SWF utilizzando la tua policy Lambda

1. Apri la [console Amazon IAM](#).
2. Scegli Roles (Ruoli), quindi Create New Role (Crea nuovo ruolo).
3. Assegna un nome al ruolo, come `swf-lambda-function`, quindi scegli Next Step (Fase successiva).

4. In AWS Service Roles, scegli Amazon SWF e scegli Next Step.
5. Nella schermata Allega policy, scegli la policy specifica per la funzione Lambda dall'elenco.
6. Scegli Next Step (Fase successiva), quindi Create Role (Crea ruolo) dopo aver esaminato il ruolo.

Collegamento del ruolo IAM al flusso di lavoro

Una volta definito il ruolo IAM, dovrai collegarlo al flusso di lavoro che lo utilizzerà per richiamare le funzioni Lambda a cui hai fornito l'accesso ad Amazon SWF.

È possibile effettuare il collegamento del ruolo al flusso di lavoro:

- Durante la registrazione del tipo di flusso di lavoro. Questo ruolo può quindi essere utilizzato come ruolo Lambda di default per ogni esecuzione di quel tipo di flusso di lavoro.
- All'avvio di un'esecuzione di flusso di lavoro. Questo ruolo sarà utilizzato esclusivamente durante l'esecuzione di quel flusso di lavoro (e nel corso dell'intera esecuzione).

Per impostare un ruolo Lambda di default per un tipo di flusso di lavoro

- Durante la chiamata RegisterWorkflowType, imposta il defaultLambdaRole campo sull'ARN del ruolo che hai definito.

Per impostare un ruolo Lambda da utilizzare durante un'esecuzione di flusso di lavoro

- Durante la chiamata StartWorkflowExecution, imposta il campo LambdaRole sull'ARN del ruolo che hai definito.

Note

se l'account che chiama RegisterWorkflowType o StartWorkflowExecution non dispone dell'autorizzazione per utilizzare il ruolo specificato, la chiamata avrà esito negativo con un OperationNotPermittedFault

Richiama la tua funzione Lambda da un flusso di lavoro Amazon SWF

È possibile utilizzare il tipo di ScheduleLambdaFunctionDecisionAttributes dati per identificare la funzione Lambda da chiamare durante l'esecuzione di un workflow.

Durante una chiamata a RespondDecisionTaskCompleted, fornisci un ScheduleLambdaFunctionDecisionAttributes elenco delle tue decisioni. Per esempio:

```
{  
  "decisions": [  
    {"ScheduleLambdaFunctionDecisionAttributes": {  
      "id": "lambdaTaskId",  
      "name": "myLambdaFunctionName",  
      "input": "inputToLambdaFunction",  
      "startToCloseTimeout": "30"  
    },  
  ]  
}
```

Imposta i seguenti parametri:

- id con un identificatore per il task Lambda. Deve essere una stringa che contiene da 1 a 256 caratteri ma non i caratteri : (due punti), / (barra obliqua), | (barra verticale), i caratteri di controllo (\u0000 - \u001f e \u007f - \u009f) e la stringa letterale `arn`.
- name con il nome della funzione Lambda. Al tuo flusso di lavoro Amazon SWF deve essere fornito un ruolo IAM che gli consenta di accedere per chiamare la funzione Lambda. Il nome fornito deve rispettare i vincoli del `FunctionName` parametro come nell'azione Lambda `Invoke`.
- input con dati di input facoltativi per la funzione. Se impostato, deve seguire i vincoli per il `ClientContext` parametro come nell'azione Lambda `Invoke`.
- `startToCloseTimeout` con un periodo massimo opzionale, in secondi, che la funzione può impiegare per eseguire prima che l'attività abbia esito negativo con un'eccezione di timeout. Il valore `NONE` può essere utilizzato per specificare una durata illimitata.

[Per ulteriori informazioni, vedere Implementazione delle attività AWS Lambda](#)

Sviluppo di un Activity Worker in Amazon SWF

Un lavoratore di attività fornisce l'implementazione di uno o più tipi di attività. Un addetto alle attività comunica con Amazon SWF per ricevere attività ed eseguirle. È possibile avere molteplici lavoratori di attività che eseguono task di attività dello stesso tipo di attività.

Amazon SWF mette un'attività a disposizione degli addetti alle attività quando chi decide la pianifica. Quando un addetto alle decisioni pianifica un'attività, fornisce i dati (determinati dall'utente) di cui l'addetto all'attività ha bisogno per eseguire l'attività. Amazon SWF inserisce questi dati nell'attività prima di inviarli all'operatore dell'attività.

La gestione dei lavoratori di attività è di tua competenza. I lavoratori di attività possono essere scritti in qualsiasi linguaggio. Un worker può essere eseguito ovunque, purché sia in grado di comunicare con Amazon SWF tramite l'API. Poiché Amazon SWF fornisce tutte le informazioni necessarie per svolgere un'attività, tutti gli addetti all'attività possono essere apolidi. La condizione stateless garantisce flussi di lavoro a elevata scalabilità; per gestire i requisiti accresciuti di capacità, è sufficiente aggiungere più lavoratori di attività.

In questa sezione viene descritto come implementare un lavoratore di attività. I lavoratori di attività devono ripetere regolarmente le operazioni elencate di seguito.

1. Esegui un sondaggio su Amazon SWF per un'attività.
2. Iniziare l'esecuzione del task.
3. Segnala periodicamente un heartbeat ad Amazon SWF se l'attività è di lunga durata.
4. Segnala che l'attività è stata completata o non riuscita e restituisci i risultati ad Amazon SWF.

Argomenti

- [Esecuzione del polling dei task di attività](#)
- [Esecuzione di un task di attività](#)
- [Segnalazione di heartbeat di task di attività](#)
- [Task di attività completato o non riuscito](#)
- [Avvio di lavoratori di attività](#)

Esecuzione del polling dei task di attività

Per eseguire attività, ogni addetto alle attività deve interrogare Amazon SWF `PollForActivityTask` richiamando periodicamente l'azione.

Nell'esempio seguente, il lavoratore di attività `ChargeCreditCardWorker01` esegue il polling di un task nell'elenco di task, ovvero `ChargeCreditCard-v0.1`. Se non sono disponibili attività, dopo 60 secondi, Amazon SWF restituisce una risposta vuota. Una risposta vuota è una struttura Task in cui il valore del `taskToken` è una stringa vuota.

```
https://swf.us-east-1.amazonaws.com
PollForActivityTask
{
  "domain" : "867530901",
  "taskList" : { "name": "ChargeCreditCard-v0.1" },
  "identity" : "ChargeCreditCardWorker01"
}
```

Se un'attività diventa disponibile, Amazon SWF la restituisce all'operatore dell'attività. Il task contiene i dati che il decisore specifica quando pianifica l'attività.

Dopo che un lavoratore di attività riceve un task di attività, è pronto per eseguire il lavoro. La sezione successiva fornisce informazioni sull'esecuzione di un task di attività.

Esecuzione di un task di attività

Dopo la ricezione di un task di attività, il lavoratore di attività è pronto per eseguirlo.

Per eseguire un task di attività

1. Programma il lavoratore di attività per interpretare il contenuto del campo di input del task. Questo campo contiene i dati specificati dal decisore durante la pianificazione del task.
2. Programma il lavoratore di attività per iniziare a elaborare i dati e a eseguire la logica.

La sezione successiva descrive come programmare gli addetti alle attività in modo che forniscano aggiornamenti di stato ad Amazon SWF per attività di lunga durata.

Segnalazione di heartbeat di task di attività

Se un timeout di heartbeat è stato registrato con il tipo di attività, il lavoratore di attività deve registrare un heartbeat prima che il timeout venga superato. Se un'attività non fornisce un battito cardiaco entro il timeout, l'attività scade, Amazon SWF la chiude e pianifica una nuova attività decisionale per informare il decisore del timeout. Il decisore può quindi pianificare di nuovo il task di attività o intraprendere un'altra operazione.

Se, dopo il timeout, l'addetto all'attività tenta di contattare Amazon SWF, ad esempio `RespondActivityTaskCompleted` chiamando, Amazon SWF restituirà un guasto. `UnknownResource`

Questa sezione descrive come fornire un heartbeat di attività.

Per registrare un heartbeat di task di attività, programma il lavoratore di attività per chiamare l'operazione `RecordActivityTaskHeartbeat`. Questa operazione fornisce inoltre un campo stringa che puoi utilizzare per archiviare dati in formato libero allo scopo di quantificare l'avanzamento nel modo più appropriato per la tua applicazione.

In questo esempio, l'activity worker segnala il battito cardiaco ad Amazon SWF e utilizza il campo dei dettagli per segnalare che l'attività è stata completata al 40%. Per segnalare l'heartbeat, il lavoratore di attività deve specificare il token del task di attività.

```
https://swf.us-east-1.amazonaws.com
RecordActivityTaskHeartbeat
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "details" : "40"
}
```

Questa operazione non crea un evento nella cronologia dell'esecuzione di flusso di lavoro; tuttavia, in caso di timeout del task, la cronologia conterrà un evento `ActivityTaskTimedOut` che include le informazioni dell'ultimo heartbeat generato dal lavoratore di attività.

Task di attività completato o non riuscito

Dopo l'esecuzione di un task, il lavoratore di attività deve segnalare se il task è stato completato o se non è riuscito.

Completamento di un task di attività

Per completare un task di attività, programma il lavoratore di attività per chiamare l'operazione `RespondActivityTaskCompleted` dopo il completamento senza errori di un task di attività, specificando il token di task.

Nell'esempio seguente, il lavoratore di attività indica che il task è stato completato senza errori.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "results": "40"
}
```

Al termine dell'attività, Amazon SWF pianifica una nuova attività decisionale per l'esecuzione del flusso di lavoro a cui è associata l'attività.

Programma il lavoratore di attività per eseguire il polling di un altro task di attività dopo il completamento del task corrente. Questa operazione crea un ciclo in cui un lavoratore di attività cerca e completa i task continuamente.

Se l'attività non risponde entro il `StartToCloseTimeout` o se `ScheduleToCloseTimeout` è stata superata, Amazon SWF scade l'attività e pianifica un'attività decisionale. Ciò consente a un decisore di intraprendere un'operazione appropriata, come una nuova pianificazione del task.

Ad esempio, se un'EC2 istanza Amazon sta eseguendo un'attività e l'istanza fallisce prima del completamento dell'attività, il decisore riceve un evento di timeout nella cronologia di esecuzione del flusso di lavoro. Se l'attività utilizza un heartbeat, chi decide riceve l'evento quando l'attività non riesce a fornire il battito cardiaco successivo dopo il fallimento dell'istanza Amazon EC2. In caso contrario, il decisore riceve l'evento se il task di attività non viene completato entro i valori di timeout. Spetta quindi al decisore riassegnare il task o intraprendere un'altra operazione.

Generazione di un errore in un task di attività

Se un addetto all'attività non è in grado di eseguire un'attività per qualche motivo, ma riesce comunque a comunicare con Amazon SWF, puoi programmarlo in modo che non riesca.

A questo proposito, devi programmare il lavoratore di attività di modo che chiama l'operazione `RespondActivityTaskFailed` che specifica il token del task.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskFailed
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "reason" : "CC-Invalid",
  "details" : "Credit Card Number Checksum Failed"
}
```

In quanto sviluppatore, devi definire i valori nei campi relativi al motivo e ai dettagli. Si tratta di stringhe in formato libero; puoi utilizzare qualsiasi convenzione di codice di errore utilizzata dall'applicazione. Amazon SWF non elabora questi valori. Tuttavia, Amazon SWF può visualizzare questi valori nella console.

Quando un'attività fallisce, Amazon SWF pianifica un'attività decisionale per l'esecuzione del flusso di lavoro a cui è associata l'attività per informare chi decide dell'errore. Programma il decisore per gestire le attività non riuscite, ad esempio pianificando di nuovo l'attività o generando un errore nell'esecuzione di flusso di lavoro, a seconda della natura dell'errore.

Avvio di lavoratori di attività

Per avviare i lavoratori di attività, crea un pacchetto della logica come eseguibile che puoi utilizzare sulla piattaforma dei lavoratori di attività. Ad esempio, potresti creare un pacchetto del codice delle attività come eseguibile Java che puoi eseguire su server Linux e Windows.

Dopo l'avvio, i lavoratori iniziano a eseguire il polling dei task. Tuttavia, fino a che il decisore non pianifica task di attività, queste ricerche scadono senza task e i lavoratori continuano a eseguire il polling.

Poiché i sondaggi sono richieste in uscita, Activity Worker può essere eseguito su qualsiasi rete che abbia accesso all'endpoint Amazon SWF.

Puoi avviare un numero illimitato di lavoratori di attività. Man mano che chi decide pianifica le attività, Amazon SWF distribuisce automaticamente le attività agli addetti all'attività di sondaggio.

Sviluppo di dispositivi decisionali in Amazon SWF

Un decisore è un'implementazione della logica di coordinamento del tipo di flusso di lavoro operativa nell'esecuzione del flusso di lavoro. Puoi assegnare più decisori a un singolo tipo di flusso di lavoro.

Poiché lo stato di esecuzione di un workflow è memorizzato nella cronologia del workflow, i decisori possono essere privi di stato. Amazon SWF conserva la cronologia di esecuzione del flusso di lavoro e la fornisce a un decisore per ogni attività decisionale. Ciò ti permette di aggiungere e rimuovere dinamicamente i decisori in base alla necessità, il che rende l'elaborazione dei flussi di lavoro altamente scalabile. Se il carico sui sistemi aumenta, basta aggiungere altri decisori per gestire la maggiore capacità. Ricorda però che ci può essere solo un task di decisione aperto in qualsiasi momento per una determinata esecuzione del flusso di lavoro.

Ogni volta che si verifica un cambiamento di stato per l'esecuzione di un flusso di lavoro, Amazon SWF pianifica un'attività decisionale. Ogni volta che un decisore riceve un task di decisione, compie le seguenti operazioni:

- Interpreta la cronologia dell'esecuzione del flusso di lavoro fornita con il task di decisione
- Applica la logica di coordinamento in base alla cronologia e prende decisioni sui passaggi successivi. Ogni decisione è rappresentata da una Struttura di decisione
- Completa l'attività decisionale e fornisce un elenco di decisioni ad Amazon SWF.

Questa sezione descrive come sviluppare un decisore, il che comporta:

- Programmare il decisore perché esegua il polling dei task di decisione
- Programmare il decisore perché interpreti la cronologia e prenda decisioni
- Programmare il decisore perché risponda a un task di decisione.

Gli esempi in questa sezione illustrano come puoi programmare un decisore per un flusso di lavoro di e-commerce, a titolo di esempio.

Puoi implementare il decider in qualsiasi lingua che desideri ed eseguirlo ovunque, purché sia in grado di comunicare con Amazon SWF tramite la sua API di servizio.

Argomenti

- [Definire la logica di coordinamento](#)
- [Esecuzione del polling dei task di decisione](#)
- [Applicare la logica di coordinamento](#)
- [Rispondere con decisioni](#)
- [Chiudere un'esecuzione del flusso di lavoro](#)
- [Lancio dei decisori](#)

Definire la logica di coordinamento

La prima cosa da fare nello sviluppo di un decisore è definire la logica di coordinamento.

Nell'esempio dell'e-commerce, la logica di coordinamento che pianifica ogni attività dopo il completamento della precedente può avere il seguente aspetto:

```
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
```

Il decisore applica la logica di coordinamento alla cronologia dell'esecuzione del flusso di lavoro e crea un elenco di decisioni al completamento del task di decisione utilizzando l'operazione `RespondDecisionTaskCompleted`.

Esecuzione del polling dei task di decisione

Ogni decisore esegue il polling dei task di decisione. I task di decisione contengono le informazioni che il decisore utilizza per generare decisioni, ad esempio la pianificazione di task di attività. Per eseguire il polling dei task di decisione, il decisore utilizza l'operazione `PollForDecisionTask`.

In questo esempio, il decisore esegue il polling di un task di decisione specificando l'elenco di task `customerOrderWorkflow-v0.1`.

```
https://swf.us-east-1.amazonaws.com
PollForDecisionTask
{
    "domain": "867530901",
    "taskList": {"name": "customerOrderWorkflow-v0.1"},
```

```
"identity": "Decider01",
"maximumPageSize": 50,
"reverseOrder": true
}
```

Se un'attività decisionale è disponibile nell'elenco delle attività specificato, Amazon SWF la restituisce immediatamente. Se non è disponibile alcuna attività decisionale, Amazon SWF mantiene la connessione aperta per un massimo di 60 secondi e restituisce un'attività non appena diventa disponibile. Se nessuna attività diventa disponibile, Amazon SWF restituisce una risposta vuota. Una risposta vuota è una struttura Task in cui il valore di `taskToken` è una stringa vuota. Ricorda di programmare il decisore perché esegua il polling di un altro task se riceve una risposta vuota.

Se è disponibile un'attività decisionale, Amazon SWF restituisce una risposta che contiene l'attività decisionale e una visualizzazione in pagine della cronologia di esecuzione del flusso di lavoro.

In questo esempio, il tipo dell'evento più recente indica che l'esecuzione è partita e che l'elemento di input contiene le informazioni necessarie a eseguire il primo task.

```
{
  "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",
        "parentWorkflowId": "parentWorkflowId"
      }
    }
  ]
}
```

```
  "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"
  }
}
],
...
}
```

Dopo aver ricevuto la cronologia, il decisore la interpreta e prende decisioni in base alla logica di coordinamento.

Poiché il numero di eventi della cronologia per una sola esecuzione può essere elevato, il risultato restituito può essere suddiviso in una serie di pagine. Per recuperare le pagine successive, effettua chiamate aggiuntive `PollForDecisionTask` utilizzando la chiamata `nextPageToken` restituita dalla chiamata iniziale. Nota che non chiavi `GetWorkflowExecutionHistory` con questo `nextPageToken` comando. Invece, chiama `PollForDecisionTask` nuovamente.

Applicare la logica di coordinamento

Dopo che il decisore riceve un task di decisione, programmallo perché interpreti la cronologia dell'esecuzione del flusso di lavoro per stabilire cosa è successo finora. In base a questo, il decisore genera un elenco di decisioni.

Nell'esempio dell'e-commerce, ci interessa solo l'ultimo evento della cronologia, quindi definiamo la seguente logica.

```
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` è `CompleteVerifyOrderActivity`, aggiungerai l'attività `ScheduleChargeCreditCardActivity` all'elenco di decisioni.

Dopo aver determinato le decisioni da prendere, il decisore può rispondere ad Amazon SWF con le decisioni appropriate.

Rispondere con decisioni

Dopo aver interpretato la cronologia del flusso di lavoro e generato un elenco di decisioni, il decisore è pronto a rispondere ad Amazon SWF con tali decisioni.

Programma il decisore perché estragga i dati che servono dalla cronologia e crei decisioni che specificano le prossime operazioni appropriate del flusso di lavoro. Chi decide ritrasmette queste decisioni ad Amazon SWF utilizzando l'azione `RespondDecisionTaskCompleted`. Consulta il riferimento all'API di Amazon Simple Workflow Service per un elenco dei [tipi di decisione](#) disponibili.

Nell'esempio delle-commerce, quando il decisore risponde con il set di decisioni generato, include anche l'input delle carte di credito dalla cronologia dell'esecuzione del flusso di lavoro. Il lavoratore di attività ha quindi le informazioni necessarie all'esecuzione del task di attività.

Quando tutte le attività nell'esecuzione sono completate, il decisore chiude l'esecuzione del flusso di lavoro.

```
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"
}
}
]
}
```

Chiudere un'esecuzione del flusso di lavoro

Quando il decisore stabilisce che il processo commerciale è completo, cioè che non ci sono più attività da eseguire, il decisore genera una decisione per chiudere l'esecuzione del flusso di lavoro.

Per chiudere un'esecuzione, programma il decisore perché interpreti gli eventi nella cronologia per stabilire cosa è successo finora e vedere se l'esecuzione deve essere chiusa.

Se il flusso di lavoro è stato completato con successo, chiudi l'esecuzione chiamando `RespondDecisionTaskCompleted` con la decisione `CompleteWorkflowExecution`. In alternativa, un'esecuzione errata può non andare a buon fine usando la decisione `FailWorkflowExecution`.

Nell'esempio dell'e-commerce, il decisore esamina la cronologia e in base alla logica di coordinamento aggiunge una decisione per chiudere l'esecuzione all'elenco di decisioni, e avvia un'operazione `RespondDecisionTaskCompleted` con una decisione di chiusura del flusso di lavoro.

Note

Ci sono alcuni casi in cui la chiusura di un'esecuzione del flusso di lavoro non va a buon fine. Ad esempio, se viene ricevuto un segnale mentre il decisore sta chiudendo l'esecuzione del flusso di lavoro, la decisione di chiusura non andrà a buon fine. Per gestire questa possibilità, assicurati che il decisore continui a eseguire il polling dei task di decisione. Inoltre, assicurati

che il decisore che riceve l'attività decisionale successiva risponda all'evento, in questo caso un segnale, che ha impedito la chiusura dell'esecuzione.

Puoi anche supportare l'annullamento delle esecuzione del flusso di lavoro. Questo può essere particolarmente utile per i flussi di lavoro di lunga durata. Per supportare l'annullamento, il decisore deve gestire l'evento `WorkflowExecutionCancelRequested` nella cronologia. Questo evento indica che è stato richiesto l'annullamento dell'esecuzione. Il decisore deve eseguire le corrette operazioni di pulizia, come l'annullamento dei task di attività in corso e la chiusura del flusso di lavoro chiamando l'operazione `RespondDecisionTaskCompleted` con la decisione `CancelWorkflowExecution`.

L'esempio seguente chiama `RespondDecisionTaskCompleted` per specificare che l'esecuzione corrente è stata annullata.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions" : [
    {
      "decisionType": "CancelWorkflowExecution",
      "CancelWorkflowExecutionAttributes": {
        "Details": "Customer canceled order"
      }
    }
  ]
}
```

Amazon SWF verifica che la decisione di chiudere o annullare l'esecuzione del flusso di lavoro sia l'ultima decisione inviata dal decisore. Ciò significa che non vale avere un set di decisioni in cui siano presenti decisioni successive a quella che chiude il flusso di lavoro.

Lancio dei decisori

Dopo averne completato lo sviluppo, sei pronto a lanciare uno o più decisori.

Per lanciare i decisori, crea un pacchetto della logica di coordinamento in un eseguibile da utilizzare sulla piattaforma dei decisori. Ad esempio, puoi creare un pacchetto del codice decisore come un eseguibile Java che funzioni su computer Linux e Windows.

Una volta avviato, i decisori dovrebbero iniziare a sondare Amazon SWF per le attività. Fino a quando non avvierai le esecuzioni dei flussi di lavoro e Amazon SWF non pianificherà le attività decisionali, questi sondaggi scadranno e riceveranno risposte vuote. Una risposta vuota è una struttura Task in cui il valore di `taskToken` è una stringa vuota. I decisori devono semplicemente proseguire il polling.

Amazon SWF garantisce che una sola attività decisionale possa essere attiva per l'esecuzione di un flusso di lavoro in qualsiasi momento. Questo impedisce problemi, come le decisioni in conflitto. Inoltre, Amazon SWF garantisce che una singola attività decisionale venga assegnata a un unico decisore, indipendentemente dal numero di decisori in esecuzione.

Se accade qualcosa che genera un'attività decisionale mentre un decisore sta elaborando un'altra attività decisionale, Amazon SWF mette in coda la nuova attività fino al completamento dell'attività corrente. Una volta completata l'attività corrente, Amazon SWF rende disponibile la nuova attività decisionale. Inoltre, le attività decisionali vengono raggruppate, nel senso che, se più attività vengono completate mentre un decisore elabora un'attività decisionale, Amazon SWF creerà una sola nuova attività decisionale per tenere conto dei diversi completamenti delle attività. Tuttavia, ciascun completamento riceve un singolo evento nella cronologia dell'esecuzione del flusso di lavoro.

Poiché i sondaggi sono richieste in uscita, i decisori possono essere eseguiti su qualsiasi rete che abbia accesso all'endpoint Amazon SWF.

Per far proseguire le esecuzioni, devono operare uno o più decisori. Puoi avviare tutti i decisori che desideri. Amazon SWF supporta più decisori che effettuano il polling sullo stesso elenco di attività.

Avvio di flussi di lavoro in Amazon SWF

Puoi avviare un'esecuzione di un tipo di flusso di lavoro registrato da qualunque applicazione che utilizza l'azione `StartWorkflowExecution`. Quando avvii l'esecuzione, a tale esecuzione associa un identificatore, chiamato `workflowId`. `workflowId` può essere qualunque stringa appropriata all'applicazione, come il numero di ordine in un'applicazione di elaborazione degli ordini. Non puoi utilizzare lo stesso `workflowId` per diverse esecuzioni del flusso di lavoro aperte all'interno dello stesso dominio. Ad esempio, se avvii due esecuzioni del flusso di lavoro con `workflowId Customer Order 01`, la seconda esecuzione non si avvierà e la richiesta restituirà un errore. Tuttavia, puoi riutilizzare il contenuto `workflowId` di un'esecuzione chiusa. Amazon SWF associa inoltre un identificatore univoco generato dal sistema, chiamato, a ogni esecuzione del flusso di `runId` lavoro.

Dopo la registrazione del tipo di flusso di lavoro e di attività, avvia il flusso di lavoro chiamando l'azione `StartWorkflowExecution`. Il valore del parametro `input` può

essere una stringa specificata dall'applicazione che sta avviando il flusso di lavoro. `executionStartToCloseTimeout` è il tempo espresso in secondi che l'esecuzione del flusso di lavoro può utilizzare dall'avvio alla chiusura. Il superamento di questo limite causa il timeout dell'esecuzione. A differenza di altri parametri di timeout in Amazon SWF, non è possibile specificare un valore `NONE` per questo timeout; esiste un limite massimo di un anno per l'esecuzione di un flusso di lavoro. Analogamente, `taskStartToCloseTimeout` è il periodo di tempo in secondi che un'attività decisionale associata a questo flusso di lavoro può impiegare prima del timeout.

```
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 l'`StartWorkflowExecution`azione ha esito positivo, Amazon SWF restituisce il file `runId` per l'esecuzione del flusso di lavoro. `runId` di un'esecuzione del flusso di lavoro è univoco all'interno di una regione specifica. Salva il `runId` nel caso in cui in seguito sia necessario specificare l'esecuzione di questo flusso di lavoro in una chiamata ad Amazon SWF. Ad esempio, dovrai usare `runId` nel caso in cui in un secondo momento dovessi inviare un segnale all'esecuzione del flusso di lavoro.

```
{"runId": "9ba33198-4b18-4792-9c15-7181fb3a8852"}
```

Impostazione della priorità delle attività in Amazon SWF

Per impostazione predefinita, i task in un elenco di task sono consegnati in base alla relativa ora di arrivo. Per quanto possibile, i task pianificati per primi vengono eseguiti per primi. Impostando una priorità opzionale, puoi dare priorità a determinate attività: Amazon SWF cercherà di fornire attività con priorità più alta in un elenco di attività prima di quelle con priorità inferiore.

Note

I task pianificati per primi sono in genere i primi a essere eseguiti, ma questo ordine non è garantito.

Puoi impostare priorità di task per flussi di lavoro e attività. La priorità di task di un flusso di lavoro non ha alcuna incidenza sulla priorità di task di attività che pianifica e nemmeno sui flussi di lavoro figlio che avvia. La priorità predefinita per un'attività o un flusso di lavoro viene impostata (da te o da Amazon SWF) durante la registrazione e la priorità dell'attività registrata viene sempre utilizzata a meno che non venga sostituita durante la pianificazione dell'attività o l'avvio di un'esecuzione del flusso di lavoro.

I valori della priorità di task possono andare da "-2147483648" a "2147483647", con i numeri più alti indicanti la priorità più elevata. Se non imposta la priorità di task per un'attività o un flusso di lavoro, verrà assegnata la priorità zero ("0").

Argomenti

- [Impostazione della priorità di task per flussi di lavoro](#)
- [Impostazione della priorità di task per attività](#)
- [Operazioni che restituiscono informazioni relative alla priorità di task](#)

Impostazione della priorità di task per flussi di lavoro

Puoi impostare la priorità di task per un flusso di lavoro durante la registrazione o l'avvio dello stesso. La priorità di task impostata alla registrazione del flusso di lavoro è utilizzata come impostazione di default per qualsiasi esecuzione di flusso di lavoro di quel tipo, a meno che non venga sovrascritta all'avvio dell'esecuzione di flusso di lavoro.

Per registrare un tipo di flusso di lavoro con una priorità di attività predefinita, imposta l'opzione `l'defaultTaskPriority` quando usi l'azione: [RegisterWorkflowType](#)

```
{  
  "domain": "867530901",  
  "name": "expeditedOrderWorkflow",  
  "version": "1.0",  
  "description": "Expedited customer orders workflow",  
  "defaultTaskStartToCloseTimeout": "600",  
  "taskList": "expeditedOrderWorkflowTaskList",  
  "state": "REGISTERED",  
  "tags": []  
}
```

```
"defaultExecutionStartToCloseTimeout": "3600",
"defaultTaskList": {"name": "mainTaskList"},
"defaultTaskPriority": "10",
"defaultChildPolicy": "TERMINATE"
}
```

È possibile sovrascrivere la priorità delle attività registrate di un tipo di flusso di lavoro quando si avvia l'esecuzione di un flusso di lavoro con [StartWorkflowExecution](#):

```
{
  "childPolicy": "TERMINATE",
  "domain": "867530901",
  "executionStartToCloseTimeout": "1800",
  "input": "arbitrary-string-that-is-meaningful-to-the-workflow",
  "tagList": ["music purchase", "digital", "ricoh-the-dog"],
  "taskList": {"name": "specialTaskList"},
  "taskPriority": "-20",
  "taskStartToCloseTimeout": "600",
  "workflowId": "20110927-T-1",
  "workflowType": {"name": "customerOrderWorkflow", "version": "1.0"}
}
```

È inoltre possibile ignorare la priorità delle attività registrate quando si avvia un flusso di lavoro secondario o quando si continua un flusso di lavoro come nuovo, ad esempio quando si risponde a una decisione con [RespondDecisionTaskCompleted](#)

Per impostare la priorità di task di un flusso di lavoro figlio, indica il valore in [startChildWorkflowExecutionDecisionAttributes](#):

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAA...",
  "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"
        }
    }
}
]
```

Quando continui un flusso di lavoro come se fosse nuovo, imposta la priorità di task in `continueAsNewWorkflowExecutionDecisionAttributes`:

```
{
    "taskToken": "AAAAAKgAAAAEAAAAAAA...",
    "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"
            }
        }
    ]
}
```

Impostazione della priorità di task per attività

Puoi impostare la priorità di task per un attività durante la registrazione o la pianificazione della stessa. La priorità di task impostata quando si registra un tipo di attività è utilizzata come priorità di default all'esecuzione dell'attività, a meno che non venga sovrascritta quando si pianifica l'attività.

Per impostare la priorità delle attività durante la registrazione di un tipo di attività, imposta l'`defaultTaskPriority` opzione quando utilizzi l'azione: [RegisterActivityType](#)

```
{
```

```
"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"
}
```

Per pianificare un'attività con una priorità, utilizza l'opzione `TaskPriority` quando pianifichi l'attività con l'azione: [RespondDecisionTaskCompleted](#)

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAA...",
  "decisions": [
    {
      "decisionType": "ScheduleActivityTask",
      "scheduleActivityTaskDecisionAttributes": {
        "activityId": "verify-account",
        "activityType": {
          "name": "activityVerify",
          "version": "1.0"
        },
        "control": "digital music",
        "input": "abab-101",
        "taskList": {"name": "mainTaskList"},
        "taskPriority": "15"
      }
    }
  ]
}
```

Operazioni che restituiscono informazioni relative alla priorità di task

Puoi ottenere informazioni sulla priorità delle attività impostata (o sulla priorità predefinita delle attività) dalle seguenti azioni di Amazon SWF:

- [DescribeActivityType](#) restituisce il `defaultTaskPriority` tipo di attività nella `configuration` sezione della risposta.

- [DescribeWorkflowExecution](#) restituisce la TaskPriority dell'esecuzione del flusso di lavoro nella sezione executionConfiguration della risposta.
- [DescribeWorkflowType](#) restituisce il tipo defaultTaskPriority di flusso di lavoro nella configuration sezione della risposta.
- [GetWorkflowExecutionHistory](#) e [PollForDecisionTask](#) forniscono informazioni sulla priorità delle attività nelle workflowExecutionStartedEventAttributes sezioni activityTaskScheduledEventAttributes decisionTaskScheduledEventAttributes workflowExecutionContinuedAsNewEventAttributes della risposta.

Gestione degli errori in Amazon SWF

Durante un'esecuzione di flusso di lavoro è possibile che si verifichino vari tipi di errore.

Argomenti

- [Errori di convalida](#)
- [Errori nell'applicazione di operazioni o decisioni](#)
- [Timeout](#)
- [Errori generati da codice utente](#)
- [Errori correlati alla chiusura di un'esecuzione di flusso di lavoro](#)

Errori di convalida

Gli errori di convalida si verificano quando una richiesta ad Amazon SWF fallisce perché non è formata correttamente o contiene dati non validi. In questo contesto, una richiesta potrebbe essere un'operazione come `DescribeDomain` oppure una decisione come `StartTimer`. Se la richiesta è un'azione, Amazon SWF restituisce un codice di errore nella risposta. Verifica questo codice di errore in quanto può fornire informazioni sull'aspetto della richiesta che ha causato l'errore. Ad esempio, uno o più degli argomenti passati con la richiesta potrebbero non essere validi. Per un elenco dei codici di errore più comuni, consulta l'argomento relativo all'azione nell'Amazon Simple Workflow Service API Reference.

Se la richiesta non riuscita è una decisione, un evento appropriato sarà elencato nella cronologia dell'esecuzione di flusso di lavoro. Ad esempio, se la decisione `StartTimer` presenta degli errori, nella cronologia viene visualizzato l'evento `StartTimerFailed`. Il decisore deve

verificare questi eventi quando riceve la cronologia in risposta a `PollForDecisionTask` o `GetWorkflowExecutionHistory`. Di seguito è riportato un elenco di possibili eventi di errore che possono verificarsi quando il formato della decisione o i dati in essa contenuti non sono validi.

Errori nell'applicazione di operazioni o decisioni

Anche se la richiesta è stata formulata correttamente, possono verificarsi degli errori quando Amazon SWF tenta di eseguire la richiesta. In tal caso, uno degli eventi seguenti nella cronologia indicherà che si è verificato un errore. Osserva il campo `reason` dell'evento per determinare la causa dell'errore.

- [CancelTimerFailed](#)
- [RequestCancelActivityTaskFailed](#)
- [RequestCancelExternalWorkflowExecutionFailed](#)
- [ScheduleActivityTaskFailed](#)
- [SignalExternalWorkflowExecutionFailed](#)
- [StartChildWorkflowExecutionFailed](#)
- [StartTimerFailed](#)

Timeout

[Decisori](#), [lavoratori di attività](#) ed [esecuzioni di flusso di lavoro](#) sono tutti soggetti a periodi di timeout.

Gli errori di questo tipo sono dovuti al timeout di un task o di un flusso di lavoro figlio. Un evento che descrive il timeout sarà visualizzato nella cronologia. Il decisore deve gestire questo evento, ad esempio, pianificando di nuovo il task o riavviando il flusso di lavoro figlio. Per ulteriori informazioni sui timeout, consulta [Tipi di timeout di Amazon SWF](#).

- [ActivityTaskTimedOut](#)
- [ChildWorkflowExecutionTimedOut](#)
- [DecisionTaskTimedOut](#)
- [WorkflowExecutionTimedOut](#)

Errori generati da codice utente

Esempi di questo tipo di condizione di errore sono gli errori nei task di attività e nei flussi di lavoro figlio. Come per gli errori di timeout, Amazon SWF aggiunge un evento appropriato alla cronologia di esecuzione del flusso di lavoro. Il decisore deve gestire questo evento, possibilmente pianificando di nuovo il task o riavviando il flusso di lavoro figlio.

- [ActivityTaskFailed](#)
- [ChildWorkflowExecutionFailed](#)

Errori correlati alla chiusura di un'esecuzione di flusso di lavoro

I decisori possono anche vedere gli eventi seguenti se tentano di chiudere un flusso di lavoro che ha un task di decisione in sospeso.

- [FailWorkflowExecutionFailed](#)
- [CompleteWorkflowExecutionFailed](#)
- [ContinueAsNewWorkflowExecutionFailed](#)
- [CancelWorkflowExecutionFailed](#)

Per ulteriori informazioni su uno qualsiasi degli eventi sopra elencati, consulta [History Event](#) nell'Amazon SWF API Reference.

Quote Amazon SWF

Amazon SWF impone quote sulle dimensioni di determinati parametri del flusso di lavoro, ad esempio sul numero di domini per account e sulla dimensione della cronologia di esecuzione del flusso di lavoro. Queste quote sono progettate per evitare che flussi di lavoro errati consumino tutte le risorse del sistema, ma non rappresentano limiti rigidi. [Se scopri che la tua applicazione supera spesso queste quote, puoi richiedere un aumento della quota di servizio.](#)

Indice

- [Quote generali degli account per Amazon SWF](#)
- [Quote sulle esecuzioni dei flussi di lavoro](#)
- [Quote sulle esecuzioni delle attività](#)
- [Quote di limitazione di Amazon SWF](#)
 - [Limitazione delle quote per tutte le regioni](#)
 - [Quote decisionali per tutte le regioni](#)
 - [Quote a livello di flusso di lavoro](#)
 - [Richiesta di aumento delle quote](#)

Quote generali degli account per Amazon SWF

- Numero massimo di domini registrati: 100

Questa quota include sia i domini registrati che quelli obsoleti.

- Numero massimo di flussi di lavoro e tipi di attività: 10.000 ciascuno per dominio

Questa quota include sia i tipi registrati che quelli obsoleti.

- Quota di chiamate API: oltre ai picchi occasionali, le applicazioni potrebbero essere limitate se effettuano un numero elevato di chiamate API in un periodo di tempo molto breve.
- Dimensione massima della richiesta: 1 MB per richiesta

Questa è la dimensione totale dei dati per richiesta API Amazon SWF, inclusa l'intestazione della richiesta e tutti gli altri dati di richiesta associati.

- Risposte troncate per Count APIs: indica che è stata raggiunta una quota interna e che la risposta non corrisponde al conteggio completo.

Alcune domande raggiungeranno internamente la quota di 1 MB menzionata sopra prima di restituire una risposta completa. Le seguenti query possono restituire una risposta troncata anziché il conteggio completo.

- [CountClosedWorkflowExecutions](#)
- [CountOpenWorkflowExecutions](#)
- [CountPendingActivityTasks](#)
- [CountPendingDecisionTasks](#)

Per ciascuna di queste query, se la risposta truncated è impostata su true, il conteggio risulta inferiore all'importo completo. Questa quota interna non può essere aumentata.

- Numero massimo di tag: 50 tag per risorsa.

Tentativo di aggiungere i tag oltre 50 comporta un errore 400, `TooManyTagsFault`.

Quote sulle esecuzioni dei flussi di lavoro

- Esecuzioni massime di flussi di lavoro aperti: 100.000 per dominio

Questo conteggio include le esecuzioni del flusso di lavoro figlio.

- Tempo massimo di esecuzione del flusso di lavoro: 1 anno. Si tratta di una quota fissa che non può essere modificata.
- Dimensione massima della cronologia di esecuzione del flusso di lavoro: 25.000 eventi. Si tratta di una quota fissa che non può essere modificata.

La best practice da adottare è strutturare ciascun flusso di lavoro in modo che la cronologia non superi i 10.000 eventi. Poiché il decisore deve recuperare la cronologia del flusso di lavoro, una cronologia più piccola consente al decisore di completare più rapidamente. Se si utilizza [Flow Framework](#), è possibile utilizzarlo `ContinueAsNew` per continuare un flusso di lavoro con una nuova cronologia.

- Numero massimo di esecuzioni di workflow secondarie aperte: 1.000 per esecuzione del flusso di lavoro

Se il tuo caso d'uso richiede di superare queste quote, puoi utilizzare le funzionalità fornite da Amazon SWF per continuare le esecuzioni e strutturare le tue applicazioni [utilizzando](#) esecuzioni

di flussi di lavoro secondari. Se ritieni che sia ancora necessario aumentare la quota, consulta.

[Richiesta di aumento delle quote](#)

Quote sulle esecuzioni delle attività

- Numero massimo di sondaggi per elenco di attività: 1.000 per elenco di attività

Puoi avere un massimo di 1.000 poller che analizzano simultaneamente un particolare elenco di task. Se superi i 1.000 poller, ricevi `LimitExceeded` exception.

Note

Sebbene il massimo sia 1.000, è possibile che si verifichino `LimitExceeded` errori ben prima di questa quota. Questo errore non significa che le tue attività siano in ritardo. Significa invece che hai il numero massimo di poller inattivi in un elenco di attività. Amazon SWF imposta questo limite per risparmiare risorse sia sul lato client che sul lato server. L'impostazione del limite impedisce che un numero eccessivo di sondaggi attenda inutilmente. È possibile ridurre gli `LimitExceeded` errori utilizzando più elenchi di attività per distribuire i sondaggi.

- Numero massimo di attività pianificate al secondo: 2.000 per elenco di attività

È possibile pianificare un massimo di 2.000 attività al secondo in un determinato elenco di attività. Se superi i 2.000, `ScheduleActivityTask` le tue decisioni falliranno con `ACTIVITY_CREATION_RATE_EXCEEDED` errori.

Note

Sebbene il massimo sia 2.000, è possibile che si verifichino `ACTIVITY_CREATION_RATE_EXCEEDED` errori ben prima di questa quota. Per ridurre questi errori, utilizzate più elenchi di attività per distribuire il carico.

- Tempo massimo di esecuzione delle attività: 1 anno (vincolato dal tempo massimo di esecuzione del flusso di lavoro)

Puoi configurare i [timeout di attività](#) per far sì che un evento del timeout avvenga se una particolare fase dell'esecuzione del [task di attività](#) dura troppo.

- Tempo massimo di permanenza in coda di un'operazione da parte di SWF: 1 anno (vincolato dalla quota di tempo di esecuzione del flusso di lavoro)

Puoi configurare i [timeout di attività](#) predefiniti durante la registrazione dell'attività per far sì che un evento del timeout avvenga se una particolare fase dell'esecuzione del [task di attività](#) dura troppo.

Puoi anche sovrascrivere i timeout di attività predefiniti quando pianifichi un task di attività nel codice del decisore.

- Numero massimo di attività aperte: 1.000 per esecuzione del flusso di lavoro.

Questa quota include sia le attività che sono state pianificate sia quelle elaborate dai lavoratori.

- Timer di apertura massimi: 1.000 per esecuzione del flusso di lavoro
- Dimensione massima input/result dei dati: 32.768 caratteri

Questa quota influisce sui dati dei risultati dell'attività o dell'esecuzione del flusso di lavoro, sui dati di input durante la pianificazione delle attività o delle esecuzioni del flusso di lavoro e sull'input inviato con un segnale di esecuzione del flusso di lavoro.

- Numero massimo di decisioni in una risposta all'attività decisionale: varia

A causa della quota di 1 MB sulla [dimensione massima delle richieste API](#), il numero di decisioni restituite in una singola chiamata [RespondDecisionTaskCompleted](#) sarà limitato in base alla dimensione dei dati utilizzati da ciascuna decisione, inclusa la dimensione dei dati di input forniti per attività pianificate o per l'esecuzione dei flussi di lavoro.

Quote di limitazione di Amazon SWF

Oltre alle quote di servizio descritte in precedenza, alcune chiamate API Amazon SWF ed eventi decisionali vengono limitati per mantenere la larghezza di banda del servizio, utilizzando uno schema token bucket. Se la frequenza delle richieste supera costantemente le tariffe elencate qui, puoi richiedere un aumento della quota limite.

Le quote di limitazione e decisione sono le stesse in tutte le aree geografiche.

Limitazione delle quote per tutte le regioni

Le seguenti quote sono applicabili a livello di account individuale. Puoi anche richiedere un aumento delle seguenti quote. Per informazioni su questa operazione, vedere [Richiesta di aumento delle quote](#).

Nome API	Dimensione bucket	Tasso di riempimento al secondo
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 API	Dimensione bucket	Tasso di riempimento al secondo
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

Quote decisionali per tutte le regioni

Le seguenti quote sono applicabili a livello di account individuale. Puoi anche richiedere un aumento delle seguenti quote. Per informazioni su questa operazione, vedere [Richiesta di aumento delle quote](#).

Nome API	Dimensione bucket	Tasso di riempimento al secondo
RequestCancelExternalWorkflowExecution	1200	120
ScheduleActivityTask	1000	200
SignalExternalWorkflowExecution	1200	120
StartChildWorkflowExecution	500	12
StartTimer	2000	200

Quote a livello di flusso di lavoro

Le seguenti quote sono applicabili a livello di flusso di lavoro e non possono essere aumentate.

Nome API	Dimensione bucket	Tasso di riempimento al secondo
GetWorkflowExecutionHistory	400	200
SignalWorkflowExecution	1000	1000
RecordActivityTaskHeartbeat	1000	1000
RequestCancelWorkflowExecution	200	200

Richiesta di aumento delle quote

Per ulteriori informazioni, consulta [Service Quotas di AWS](#) nella Riferimenti generali di AWS.

Risorse aggiuntive e informazioni di riferimento per Amazon SWF

Questo capitolo fornisce risorse aggiuntive e informazioni di riferimento utili per lo sviluppo di flussi di lavoro con Amazon SWF.

Argomenti

- [Tipi di timeout di Amazon SWF](#)
- [Endpoint del servizio Amazon Simple Workflow](#)
- [Documentazione aggiuntiva per Amazon Simple Workflow Service](#)
- [Risorse Web per Amazon Simple Workflow Service](#)
- [Opzioni di migrazione per Ruby Flow](#)

Tipi di timeout di Amazon SWF

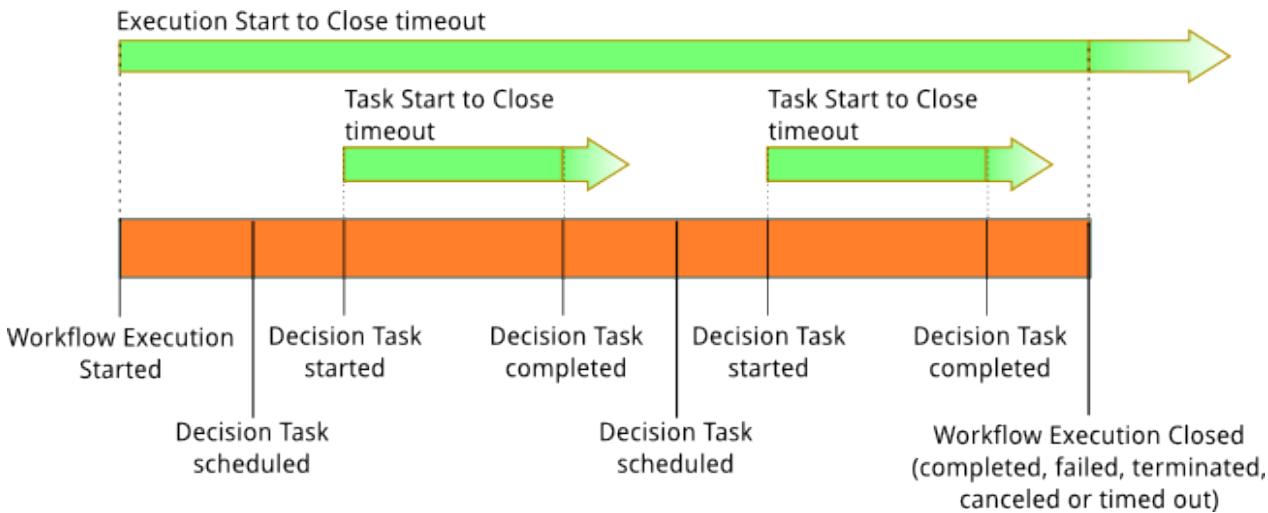
Per garantire che le esecuzioni dei flussi di lavoro vengano eseguite correttamente, puoi impostare diversi tipi di timeout con Amazon SWF. Alcuni timeout specificano la durata totale del flusso di lavoro. Altri timeout specificano quanto impiegano le attività prima di essere assegnate a un lavoratore e quanto ci vuole a completarle dal momento in cui sono state pianificate. Tutti i timeout nell'API Amazon SWF sono specificati in secondi. Amazon SWF supporta anche la stringa NONE come valore di timeout, che indica l'assenza di timeout.

Per i timeout relativi alle attività decisionali e alle attività, Amazon SWF aggiunge un evento alla cronologia di esecuzione del flusso di lavoro. Gli attributi dell'evento forniscono informazioni sul tipo di timeout verificatosi e su quale attività decisionale o attività è stata influenzata. Amazon SWF pianifica anche un'attività decisionale. Quando il decisore riceve il nuovo compito decisionale, vedrà l'evento di timeout nella cronologia e intraprenderà l'azione appropriata richiamando l'azione. [RespondDecisionTaskCompleted](#)

Un task si considera aperto dal momento in cui è pianificato fino alla sua chiusura. Perciò un task è segnalato come aperto quando un lavoratore lo sta elaborando. Un task è chiuso quando un lavoratore lo segnala come [completato](#), [annullato](#) o [non riuscito](#). Un'attività può anche essere chiusa da Amazon SWF a seguito di un timeout.

I timeout nel flusso di lavoro e i task di decisione

Il diagramma seguente mostra la correlazione tra i timeout del flusso di lavoro e di decisione e il ciclo di vita di un flusso di lavoro:



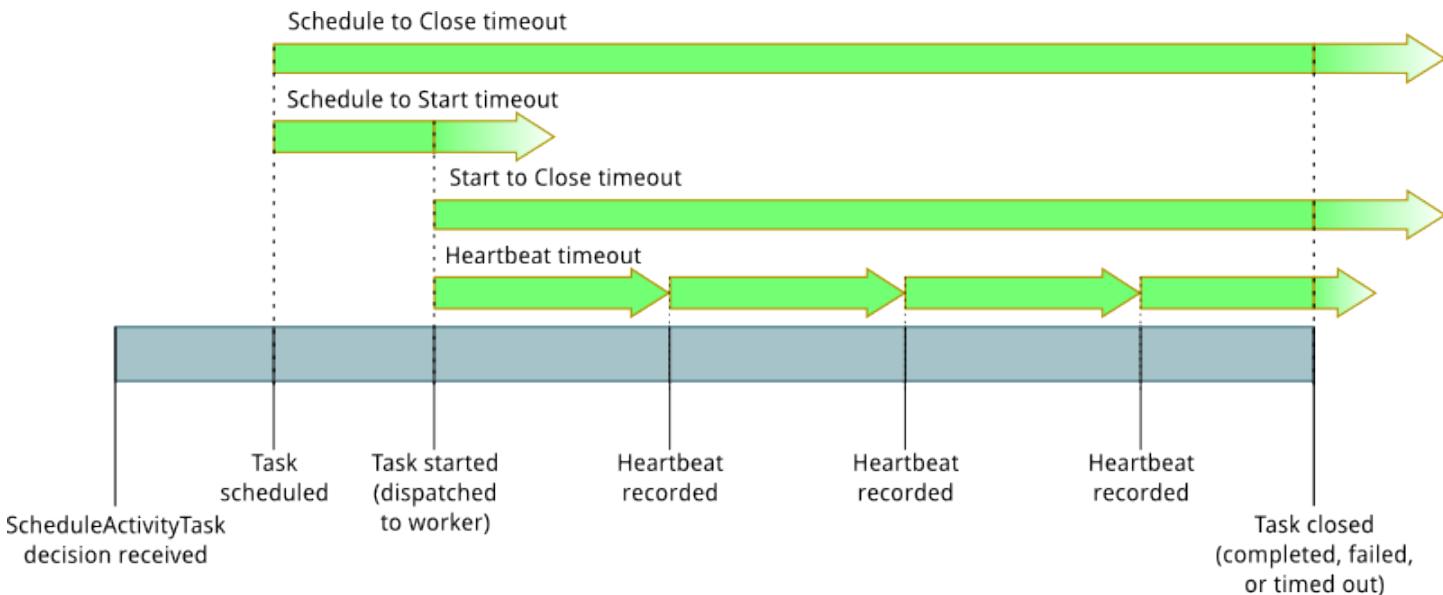
Esistono due tipi di timeout che interessano i task del flusso di lavoro e di decisione:

- Workflow Start to Close (**timeoutType: START_TO_CLOSE**): questo timeout specifica il tempo massimo necessario per completare l'esecuzione di un flusso di lavoro. È impostato come predefinito durante la registrazione del flusso di lavoro, ma può essere sovrascritto con un valore diverso quando il flusso di lavoro inizia. Se questo timeout viene superato, Amazon SWF chiude l'esecuzione del flusso di lavoro e aggiunge un evento di [WorkflowExecutionTimedOut](#) tipo alla cronologia di esecuzione del flusso di lavoro. Oltre al **timeoutType**, gli attributi dell'evento specificano la **childPolicy** valida per l'esecuzione del flusso di lavoro. La policy figlio specifica in che modo vengono gestite le esecuzioni del flusso di lavoro figlio se quella padre scade o termina in altro modo. Ad esempio, se la **childPolicy** è impostata su TERMINATA, allora le esecuzioni del flusso di lavoro figlio verranno terminate. Una volta scaduta un'esecuzione del flusso di lavoro, non potrai più intervenire se non con chiamate di visibilità.
- Inizio e chiusura dell'attività decisionale (**timeoutType: START_TO_CLOSE**): questo timeout specifica il tempo massimo che il decisore corrispondente può impiegare per completare un'attività decisionale. Viene impostato durante la registrazione del tipo di flusso di lavoro. Se questo timeout viene superato, l'attività viene contrassegnata come scaduta nella cronologia di esecuzione del flusso di lavoro e Amazon SWF aggiunge un evento di tipo [DecisionTaskTimedOut](#) alla cronologia del flusso di lavoro. Gli attributi dell'evento includeranno gli eventi che corrispondono a quando questo task decisionale è stato pianificato (`scheduledEventId`) e quando è stato avviato (`startedEventId`). Oltre ad aggiungere l'evento, Amazon SWF pianifica anche una

nuova attività decisionale per avvisare il decisore che tale attività decisionale è scaduta. Dopo che si verifica questo timeout, il tentativo di completare il task di decisione scaduto utilizzando `RespondDecisionTaskCompleted` non andrà a buon fine.

Timeout nei task di attività

Il diagramma seguente mostra la correlazione tra i timeout e il ciclo di vita di un task di attività:



Esistono quattro tipi di timeout che interessano i task di attività:

- Inizio attività da inizio a chiusura (**timeoutType: START_TO_CLOSE**): questo timeout specifica il tempo massimo che un addetto all'attività può impiegare per elaborare un'attività dopo che il lavoratore ha ricevuto l'attività. Tenta di chiudere un'attività scaduta utilizzando [RespondActivityTaskCanceled](#), [RespondActivityTaskCompleted](#), e [RespondActivityTaskFailed](#) avrà esito negativo.
- Activity Task Heartbeat (**timeoutType: HEARTBEAT**): questo timeout specifica il tempo massimo di esecuzione di un'attività prima che il relativo avanzamento nel corso dell'azione possa avvenire. [RecordActivityTaskHeartbeat](#)
- Activity Task Schedule to Start (**timeoutType: SCHEDULE_TO_START**): questo timeout specifica per quanto tempo Amazon SWF attende prima di scadere il timeout dell'attività se non sono disponibili lavoratori per eseguire l'attività. Una volta scaduto, il task non verrà assegnato ad altri lavoratori.
- Activity Task Schedule to Close (**timeoutType: SCHEDULE_TO_CLOSE**): questo timeout specifica quanto tempo può impiegare l'attività dal momento in cui è pianificata al momento in cui

viene completata. Come procedura ottimale, questo valore non deve essere maggiore della somma del timeout dell'attività e del schedule-to-start timeout dell'attività. start-to-close

Note

Ciascun tipo di timeout ha un valore predefinito, generalmente impostato su NONE (infinito). In ogni caso, il tempo massimo per l'esecuzione delle attività è un anno.

In fase di registrazione del tipo di attività si impostano valori predefiniti, ma puoi sovrascriverli con nuovi valori quando [pianifichi](#) il task di attività. Quando si verifica uno di questi timeout, Amazon SWF aggiungerà [un](#) evento di [ActivityTaskTimedOut](#) tipo alla cronologia del flusso di lavoro. L'attributo del valore `timeoutType` di questo evento specifica quale di questi timeout si è verificato. Per ciascuno dei timeout, il valore del `timeoutType` è indicato tra parentesi. Gli attributi dell'evento includeranno anche gli eventi che corrispondono a quando l'attività è stata pianificata (`scheduledEventId`) e quando è stata avviata (`startedEventId`). Oltre ad aggiungere l'evento, Amazon SWF pianifica anche una nuova attività decisionale per avvisare chi decide che si è verificato il timeout.

Endpoint del servizio Amazon Simple Workflow

Un elenco delle [regioni e degli endpoint attuali di Amazon SWF](#) è fornito in Riferimenti generali di Amazon Web Services, insieme agli endpoint per altri servizi.

I domini Amazon SWF e tutti i flussi di lavoro e le attività correlati devono esistere all'interno della stessa regione per comunicare tra loro. Inoltre, i domini, i flussi di lavoro e le attività registrati in una regione non esistono in altre regioni. Ad esempio, se crei un dominio denominato "MySampleDomain" sia in us-east-1 che in us-west-2, questi esistono come domini separati: nessuno dei flussi di lavoro, degli elenchi di attività, delle attività o dei dati associati ai tuoi domini viene condiviso tra le regioni.

Se utilizzi altre AWS risorse nei tuoi flussi di lavoro, come EC2 le istanze Amazon, anche queste devono esistere nella stessa regione delle risorse Amazon SWF. Le uniche eccezioni sono i servizi che si estendono su più regioni, come Amazon S3 e IAM. Puoi accedere a questi servizi dai flussi di lavoro presenti nelle regioni che li supportano.

Documentazione aggiuntiva per Amazon Simple Workflow Service

Oltre alla Guida per gli sviluppatori, puoi trovare utile la seguente documentazione.

Riferimento all'API Amazon Simple Workflow Service

Il [riferimento all'API Amazon Simple Workflow Service](#) fornisce informazioni dettagliate sull'API HTTP di Amazon SWF, tra cui azioni, strutture di richiesta e risposta e codici di errore.

AWS Flow Framework Documentazione

[AWS Flow Framework](#) È un framework di programmazione che semplifica il processo di implementazione di applicazioni asincrone distribuite che utilizzano Amazon SWF per gestire i flussi di lavoro e le attività, in modo che tu possa concentrarti sull'implementazione della logica del flusso di lavoro.

Ciascuno AWS Flow Framework è progettato per funzionare in modo idiomatico nel linguaggio per cui è stato progettato, quindi puoi lavorare in modo naturale con il tuo linguaggio preferito per implementare flussi di lavoro con tutti i vantaggi di Amazon SWF.

Esiste un AWS Flow Framework per Java. La [AWS Flow Framework for Java Developer Guide](#) fornisce informazioni su come ottenere, configurare e utilizzare la versione AWS Flow Framework per Java.

AWS Documentazione SDK

I AWS Software Development Kit (SDKs) forniscono l'accesso ad Amazon SWF in molti linguaggi di programmazione diversi. SDKs Seguono da vicino l'API HTTP, ma forniscono anche interfacce di programmazione specifiche del linguaggio per alcune funzionalità di Amazon SWF. Puoi trovare altre informazioni su ciascun kit SDK visitando i seguenti link.

Note

Di seguito sono elencate solo le aziende SDKs che supportano Amazon SWF al momento della stesura del documento. Per un elenco completo di quelli disponibili AWS SDKs, visita la pagina [Tools for Amazon Web Services](#).

Java

AWS SDK per Java Fornisce un'API Java per i servizi di AWS infrastruttura.

Per vedere la documentazione disponibile, consulta la pagina [Documentazione di AWS SDK per Java](#). Puoi anche accedere direttamente alle sezioni di Amazon SWF nel riferimento dell'SDK seguendo questi link:

- [Class: AmazonSimpleWorkflowClient](#)
- [Class: AmazonSimpleWorkflowAsyncClient](#)
- [Interface: AmazonSimpleWorkflow](#)
- [Interface: AmazonSimpleWorkflowAsync](#)

JavaScript

AWS SDK per JavaScript Consente agli sviluppatori di creare librerie o applicazioni che utilizzano AWS servizi utilizzando una semplice easy-to-use API disponibile sia nel browser che all'interno delle applicazioni Node.js sul server.

Per vedere la documentazione disponibile, consulta la pagina [Documentazione di AWS SDK per JavaScript](#). Puoi anche andare direttamente alla sezione Amazon SWF nel riferimento all'SDK seguendo questo link:

- [Class: AWS.SimpleWorkflow](#)

.NET

AWS SDK per .NET Si tratta di un unico pacchetto scaricabile che include i modelli di progetto di Visual Studio, la AWS libreria.NET, esempi di codice C# e documentazione. AWS SDK per .NET Ciò semplifica la creazione di applicazioni.NET per Amazon SWF e altri servizi per gli sviluppatori Windows.

Per vedere la documentazione disponibile, consulta la pagina [Documentazione di AWS SDK per .NET](#). Puoi anche accedere direttamente alle sezioni di Amazon SWF nel riferimento dell'SDK seguendo questi link:

- [Namespace: Amazon.SimpleWorkflow](#)
- [Namespace: Amazon.SimpleWorkflow.Model](#)

PHP

AWS SDK for PHP Fornisce un'interfaccia di programmazione PHP per Amazon SWF.

Per vedere la documentazione disponibile, consulta la pagina [Documentazione di AWS SDK for PHP](#). Puoi anche andare direttamente alla sezione Amazon SWF nel riferimento all'SDK seguendo questo link:

- [Class: SwfClient](#)

Python

AWS SDK per Python (Boto) Fornisce un'interfaccia di programmazione Python per Amazon SWF.

Per visualizzare la documentazione disponibile, consulta la pagina [boto: Un'interfaccia Python per Amazon Web Services](#). Puoi anche accedere direttamente alle sezioni Amazon SWF della documentazione seguendo questi link:

- [Tutorial su Amazon SWF](#)
- [Riferimento Amazon SWF](#)

Ruby

AWS SDK per Ruby Fornisce un'interfaccia di programmazione Ruby per Amazon SWF.

Per vedere la documentazione disponibile, consulta la pagina [Documentazione di AWS SDK per Ruby](#). Puoi anche andare direttamente alla sezione Amazon SWF nel riferimento all'SDK seguendo questo link:

- [Classe: Workflow AWS::Simple](#)

AWS CLI Documentazione

Il AWS Command Line Interface (AWS CLI) è uno strumento unificato per gestire i tuoi AWS servizi. Con un solo strumento da scaricare e configurare, puoi controllare più AWS servizi dalla riga di comando e automatizzarli tramite script.

Per ulteriori informazioni su AWS CLI, consulta la [AWS Command Line Interface](#) pagina.

Per una panoramica dei comandi disponibili per Amazon SWF, consulta [swf](#) nel Command Reference.AWS CLI

Risorse Web per Amazon Simple Workflow Service

Esistono numerose risorse Web che puoi utilizzare per saperne di più su Amazon SWF o per ricevere assistenza sull'uso del servizio e sullo sviluppo di flussi di lavoro.

Forum Amazon SWF

Il forum Amazon SWF ti consente di comunicare con altri sviluppatori di Amazon SWF e membri del team di sviluppo Amazon SWF di Amazon per porre domande e ottenere risposte.

Puoi visitare il forum all'indirizzo: [Forum: Amazon Simple Workflow Service](#).

Domande frequenti su Amazon SWF

Le domande frequenti su Amazon SWF forniscono risposte alle domande frequenti su Amazon SWF, inclusa una panoramica dei casi d'uso comuni, delle differenze tra Amazon SWF e altri servizi e altro ancora.

Puoi accedere alle domande frequenti qui: Domande frequenti [su Amazon SWF](#).

Video di Amazon SWF

Il canale [Amazon Web Services](#) YouTube offre corsi di formazione video per tutti i servizi Web di Amazon, incluso Amazon SWF. Per un elenco completo dei video relativi ad Amazon SWF, utilizza la seguente query: [Simple Workflow in Amazon Web Services](#)

Opzioni di migrazione per Ruby Flow

AWS Flow Framework for Ruby non è più in fase di sviluppo attivo. Il codice esistente continuerà a funzionare indefinitamente, ma non verranno rilasciate nuove funzionalità o versioni. Questo argomento tratterà le opzioni di utilizzo e migrazione per continuare a lavorare con Amazon SWF e informazioni su come migrare a Step Functions.

Opzione	Descrizione
Continua a utilizzare Ruby Flow Framework	Per ora, Ruby Flow Framework continuerà a funzionare. Se non fai niente, il codice continuerà a funzionare così com'è. Pianifica la migrazione da AWS Flow Framework for Ruby nelle prossime future.
Esegui la migrazione a Java Flow Framework	Java Flow Framework rimane in fase di sviluppo attivo e continuerà a ricevere nuove funzionalità e aggiornamenti.
Migrare a Step Functions	Step Functions consente di coordinare i componenti di applicazioni distribuite utilizzando flussi di lavoro visivi controllati da una macchina a stati.
Utilizzate direttamente l'API SWF, senza Flow Framework	Potete continuare a lavorare in Ruby e utilizzare direttamente l'API SWF anziché Ruby Flow Framework.

Il vantaggio offerto da Flow Framework, sia per Ruby che per Java, è che consente di concentrarsi sulla logica del flusso di lavoro. Il framework gestisce la maggior parte delle informazioni di comunicazione e coordinamento e una parte della complessità è astratta. Puoi continuare ad avere lo stesso livello di astrazione migrando a Java Flow Framework oppure puoi interagire direttamente con Amazon SWF SDK.

Continua a utilizzare Ruby Flow Framework

Il AWS Flow Framework for Ruby continuerà a funzionare come adesso a breve termine. Se hai dei flussi di lavoro scritti in AWS Flow Framework for Ruby, questi continueranno a funzionare. Senza aggiornamenti, supporto o correzioni di sicurezza, è meglio avere un piano preciso per migrare da Ruby in un prossimo futuro. AWS Flow Framework

Esegui la migrazione a Java Flow Framework

The AWS Flow Framework for Java rimarrà in fase di sviluppo attivo. Concettualmente, AWS Flow Framework per Java è simile a AWS Flow Framework Ruby: puoi comunque concentrarti sulla logica del flusso di lavoro e il framework ti aiuterà a gestire la logica di decisione e semplificherà la gestione di altri aspetti di Amazon SWF.

- [AWS Flow Framework per Java](#)
- [AWS Flow Framework per Java API Reference](#)

Migrare a Step Functions

AWS Step Functions fornisce un servizio simile ad Amazon SWF, ma in cui la logica del flusso di lavoro è controllata da una macchina a stati. Step Functions consente di coordinare i componenti delle applicazioni e dei microservizi distribuiti utilizzando flussi di lavoro visivi. La creazione di applicazioni a partire da componenti individuali che eseguono ciascuno una funzione discreta, detta anche task, consente di dimensionare e modificare rapidamente le applicazioni. Step Functions offre un modo affidabile per coordinare i componenti e gestire le funzioni dell'applicazione. Una console grafica visualizza i componenti della tua applicazione in una serie di fasi. Attiva e tiene traccia automaticamente di ogni passaggio e riprova in caso di errori, in modo che l'applicazione venga eseguita nell'ordine e come previsto, ogni volta. Step Functions registra lo stato di ogni passaggio, così quando qualcosa va storto, puoi diagnosticare ed eseguire rapidamente il debug dei problemi.

In Step Functions, gestisci il coordinamento delle tue attività utilizzando una macchina a stati, scritta in JSON dichiarativo, definita utilizzando [Amazon States Language](#). Utilizzando una macchina

a stati, non è necessario scrivere e gestire un programma decisionale per controllare la logica dell'applicazione. Step Functions offre un approccio intuitivo, produttivo e agile per coordinare i componenti delle applicazioni utilizzando flussi di lavoro visivi. Dovresti prendere in considerazione l'utilizzo AWS Step Functions per tutte le tue nuove applicazioni e Step Functions offre un'eccellente piattaforma su cui migrare per i flussi di lavoro che hai attualmente implementato in AWS Flow Framework for Ruby.

Per aiutarti a migrare le tue attività su Step Functions, continuando a sfruttare le tue competenze linguistiche in Ruby, Step Functions fornisce un esempio di Ruby activity worker. Questo esempio utilizza le migliori pratiche per l'implementazione di un activity worker e può essere utilizzato come modello per migrare la logica delle attività a Step Functions. Per ulteriori informazioni, consultate l'argomento [Example Activity Worker in Ruby](#) nella Guida per gli AWS Step Functions sviluppatori.

Note

Per molti clienti, la migrazione a Step Functions da AWS Flow Framework for Ruby è l'opzione migliore. Tuttavia, se desideri che i segnali intervengano nei tuoi processi o se devi avviare processi secondari che restituiscono un risultato a un genitore, prendi in considerazione l'utilizzo diretto dell'API Amazon SWF o la migrazione a AWS Flow Framework for Java.

Per ulteriori informazioni su AWS Step Functions, consulta:

- [AWS Step Functions Guida per gli sviluppatori](#)
- [AWS Step Functions Documentazione di riferimento delle API](#)
- [AWS Step Functions Riferimento alla riga di comando](#)

Usa direttamente l'API Amazon SWF

Sebbene AWS Flow Framework for Ruby gestisca parte della complessità di Amazon SWF, puoi anche utilizzare direttamente l'API Amazon SWF. Ciò ti consente di creare flussi di lavoro in cui hai un controllo totale sull'implementazione e sul coordinamento dei task, senza doverti preoccupare delle complessità sottostanti come il monitoraggio dell'avanzamento e la gestione dello stato.

- [Guida per gli sviluppatori di Amazon Simple Workflow Service](#)
- [Riferimento all'API Amazon Simple Workflow Service](#)

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione della Amazon Simple Workflow Service Developer Guide.

Modifica	Descrizione	Data della modifica
Aggiornamento solo della documentazione	Amazon SWF ora include una sezione sulle notifiche AWS utente, Servizio AWS che funge da posizione centrale per le AWS notifiche nel. Console di gestione AWS Per ulteriori informazioni, consulta Utilizzo Notifiche all'utente AWS con Amazon Simple Workflow Service .	4 maggio 2023
Aggiornamento	Amazon SWF offre ora una nuova esperienza di console per gestire i flussi di lavoro SWF e le relative azioni relative all'esecuzione. Per ulteriori informazioni, consulta i tutorial sulla console Amazon SWF .	12 settembre 2022
Aggiornamento	È stata aggiornata la Quote sulle esecuzioni delle attività sezione per includere Maximum tasks scheduled per second e la Metriche di Amazon SWF per CloudWatch pagina per includere informazioni sull' utilizzo di nomi di risorse non ASCII con CloudWatch	12 maggio 2021
Nuova caratteristica	Amazon Simple Workflow Service ora supporta Amazon EventBridge. Per ulteriori informazioni, consultare: <ul style="list-style-type: none">• EventBridge per Amazon SWF• EventBridge Guida per l'utente	18 dicembre 2020
Nuova caratteristica	Amazon Simple Workflow Service supporta le autorizzazioni IAM tramite tag. Per ulteriori informazioni, consulta gli argomenti seguenti. <ul style="list-style-type: none">• Tag in Amazon SWF• Gestione dei tag	20 giugno 2019

Modifica	Descrizione	Data della modifica
	<ul style="list-style-type: none"> • Esecuzioni di flussi di lavoro di tag • Controlla l'accesso ai domini con i tag • TagResource • UntagResource • ListTagsForResource • RegisterDomain 	
Nuova caratteristica	Amazon Simple Workflow Service è ora disponibile nella regione Europa (Stoccolma).	12 dicembre 2018
Aggiornamento	È stato migliorato l'argomento di Amazon Simple Workflow Service sull' CloudTrail integrazione. Consultare Registrazione delle chiamate API con AWS CloudTrail .	7 agosto 2018
Aggiornamento	Sono state aggiunte informazioni sulla nuova PendingTasks metrica per CloudWatch. Per ulteriori informazioni, consulta Metriche di Amazon SWF .	18 giugno 2018
Aggiornamento	Miglioramento dell'evidenziazione della sintassi nei codici di esempio.	29 marzo 2018
Aggiornamento	È stato aggiunto un argomento che descrive le opzioni per gli utenti di Ruby Flow per migrare da quella piattaforma. Per ulteriori informazioni, consulta Opzioni di migrazione per Ruby Flow .	9 marzo 2018
Aggiornamento	Miglioramento della navigazione nell'argomento sui concetti avanzati. Consultare Concetti avanzati di flusso di lavoro in Amazon SWF .	19 febbraio 2018
Aggiornamento	Documentazione CloudWatch delle metriche migliorata aggiungendo informazioni statistiche valide. Consultare Metriche di Amazon SWF per CloudWatch .	4 dicembre 2017

Modifica	Descrizione	Data della modifica
Aggiornamento	Modifica del sommario per migliorare la struttura del documento. Aggiunta di nuove informazioni sui Parametri di eventi API e di decisione .	9 novembre 2017
Aggiornamento	Aggiornamento della sezione Quote Amazon SWF per includere i limiti di throttling per tutte le regioni.	18 ottobre 2017
Aggiornamento	Modifica di <code>task_list</code> in <code>workflowId</code> nella sezione Guida introduttiva ad Amazon SWF per evitare confusione con <code>activity_list</code> .	25 luglio 2017
Aggiornamento	Pulizia degli esempi di codice all'interno di questa guida.	5 giugno 2017
Aggiornamento	Semplificazione e miglioramento dell'organizzazione e dei contenuti della guida.	19 maggio 2017
Aggiornamento	Aggiornamenti e correzione di collegamenti.	16 maggio 2017
Aggiornamento	Aggiornamenti e correzione di collegamenti.	1 Ottobre 2016
Supporto per attività Lambda	Puoi specificare attività Lambda oltre alle tradizionali attività nei tuoi flussi di lavoro. Per ulteriori informazioni, consulta AWS Lambda attività in Amazon SWF .	21 luglio 2015
Supporto per l'impostazione della priorità di task	Amazon SWF ora include il supporto per l'impostazione della priorità delle attività in un elenco di attività e cercherà di fornire quelle con priorità più alta prima delle attività con priorità inferiore. Informazioni sull'impostazione della priorità di task per flusso di lavoro e attività sono fornite in Impostazione della priorità delle attività in Amazon SWF .	17 dicembre 2014

Modifica	Descrizione	Data della modifica
Aggiornamento	È stato aggiunto un nuovo argomento che descrive come registrare le chiamate API Amazon SWF utilizzando CloudTrail: Registrazione delle chiamate API con AWS CloudTrail	8 maggio 2014
Aggiornamento	Sono stati aggiunti due nuovi argomenti relativi ai CloudWatch parametri per Amazon SWF Metriche di Amazon SWF per CloudWatch : che fornisce un elenco e descrizioni dei parametri supportati Visualizzazione delle metriche di Amazon SWF per l'utilizzo di CloudWatch Console di gestione AWS e, che fornisce informazioni su come visualizzare i parametri e impostare allarmi con CloudWatch Console di gestione AWS	28 aprile 2014
Aggiornamento	Aggiunta di una nuova sezione, Risorse aggiuntive e informazioni di riferimento per Amazon SWF , Questa sezione fornisce alcune informazioni di riferimento sul servizio e fornisce informazioni su documentazione aggiuntiva, esempi, codice e altre risorse Web per gli sviluppatori di Amazon SWF.	19 marzo 2014
Aggiornamento	Aggiunta di un nuovo tutorial relativo ai flussi di lavoro. Consultare Guida introduttiva ad Amazon SWF .	25 ottobre 2013
Aggiornamento	Aggiunta di informazioni ed esempi di AWS CLI .	26 agosto 2013
Aggiornamento	Aggiornamenti e correzioni.	1 agosto 2013
Aggiornamento	È stato aggiornato il documento per descrivere come utilizzare IAM per il controllo degli accessi.	22 febbraio 2013
Versione iniziale	Questa è la prima versione della Amazon Simple Workflow Service Developer Guide.	16 Ottobre 2012

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.